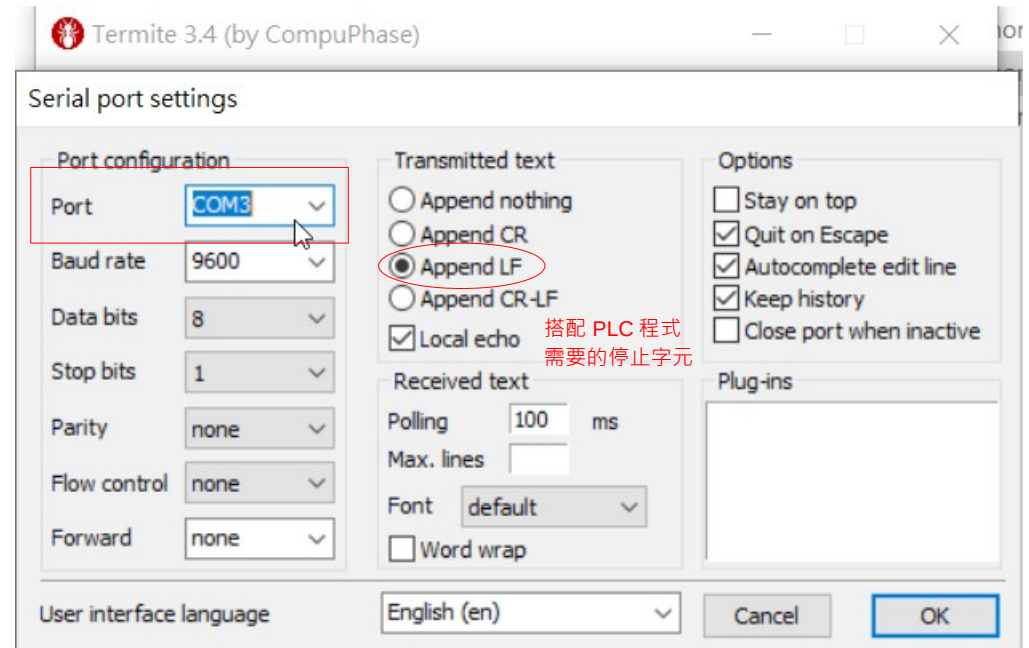


# PC 端和 PLC 序列埠溝通的軟體

- 發展 PLC 程式 (韌體) 初期，由於沒把握程式是否可正常工作，所以得用現成的工具來驗證
  - 超級終端機 (Hyper Terminal)
    - 原本內建於 windows 作業系統，自 windows 7 起被拿掉，當然還是可以透過其他方式間接重新安裝，不過我用起來不太適應，故不建議用它 (超級終端機 .zip)
  - Termit : termit-3.4.zip 解開後即可使用，操作簡單
- 一旦確定 PLC 程式可以正常工作，萬能的 Python 就可以上場了

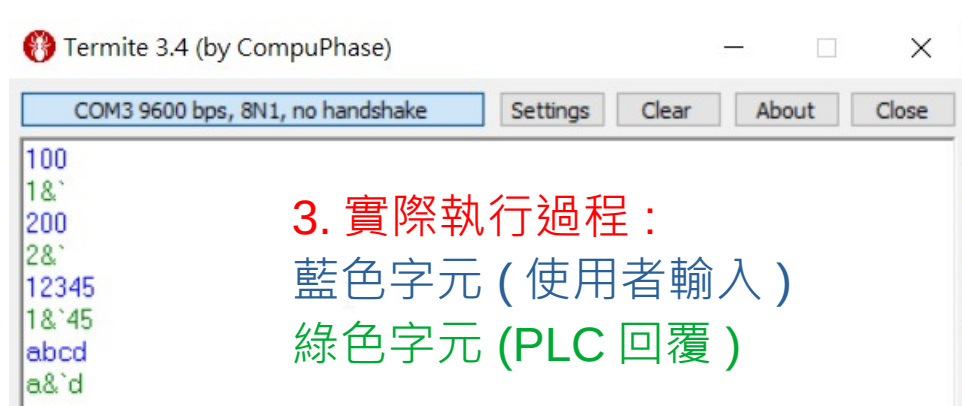
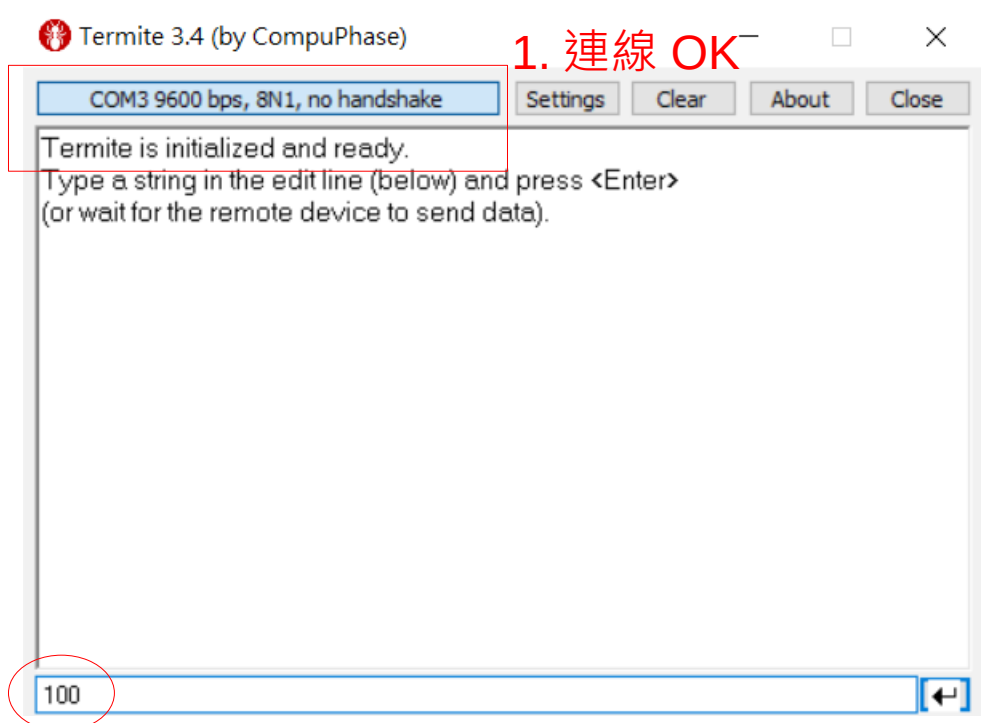
# Termite 的設定



先確定 windows 10 的 com port 設定

搭配 PLC 程式  
需要的停止字元

# Termite 執行過程



2. 準備輸入字元 (例如，100 這三個字元)，稍後透過序列埠傳給 PLC

# 用 Python 解讀資料

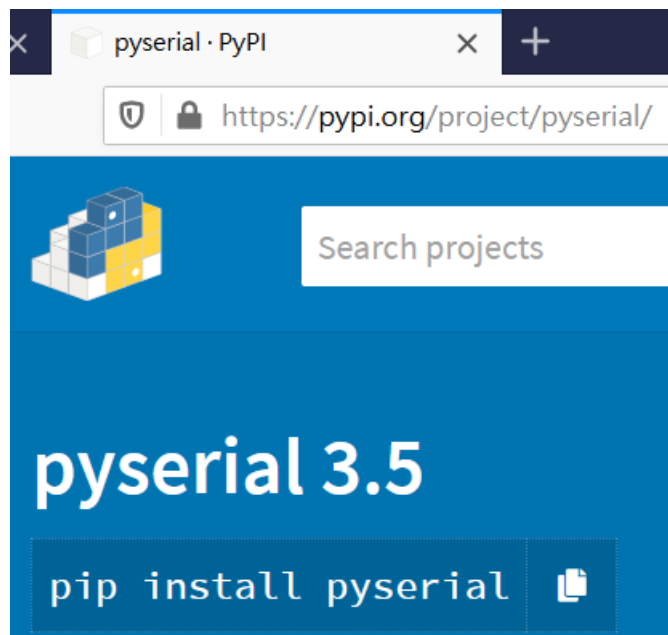
```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Python_projects\rs232\plc_rs232_UserInput.py =====
COM3
-----
input:100
got b'l\x00\n'
A10 = 3.1 V
-----
input:200
got b'2&\n'
A10 = 3.1 V
-----
input:12345
got b'l&45\n'
A10 = 3.1 V
-----
input:abcd
got b'a&d\n'
A10 = 3.1 V
-----
input:
Traceback (most recent call last):
  File "D:\Python_projects\rs232\plc_rs232_UserInput.py", line 27, in <module>
    control = input('input:')
KeyboardInterrupt
>>>
```

&' 這兩個字元被解讀成 3.1 這個數值

按「Ctrl」+「C」離開互動式迴路

# Python 程式的原始碼

記得先安裝 **pyserial** 第三方模組，之後 **python** 才能使用序列埠和 **PLC** 溝通



```
plc_rs232_UserInput.py - D:\Python_projects\rs232\plc_rs232_UserInput.py
File Edit Format Run Options Window Help
import serial
import time
ser = serial.Serial('COM3') # open first serial port
print(ser.portstr) # check which port was really used
ser.timeout = 0.5 # read timeout?

def write_plc(characters_sent):
    ser.write(characters_sent + b'\n')
    time.sleep(1)
    get = ser.read(len(characters_sent)+1)
    print('got', str(get))
    adc0 = int.from_bytes(get[1:3], 'big')
    voltage = adc0 / 3200 # suppose max 10 volt @ 32000
    print('AIO = {:.1f} V'.format(voltage))

while True:
    print('-'*5)
    control = input('input:')
    write_plc(bytes(control, encoding='ascii'))

ser.close()
```

使用時，名稱是 **serial**，而不是 **pyserial**