Appendix

E

Python 軟硬整合：使用 Arduino

Arduino IDE 內建了各種應用的 Firmata 韌體，透過 Firmata 韌體，Python 程式就可以使用 USB 串列埠與 Arduino 作傳輸，達到由 Python 控制 Arduino 的目標。

PyFirmata 模組可以讓 Python 程式和已上傳 Firmata 韌體的 Arduino 板子，透過 USB 串列埠作資料傳輸，也就是說可以利用 Python 程式控制 Arduino。

要在 Python 中撰寫程式控制 Arduino，必須在電腦中安裝 Python 的 pySerial 模組，同時也要撰寫並上傳 Arduino 程式，透過 pySerial 模組和 Arduino 進行通訊。

E.1 使用 Python 控制 Arduino

Python 程式可以使用 USB 序列埠與 Arduino 作傳輸，常用的方式有下列幾種：

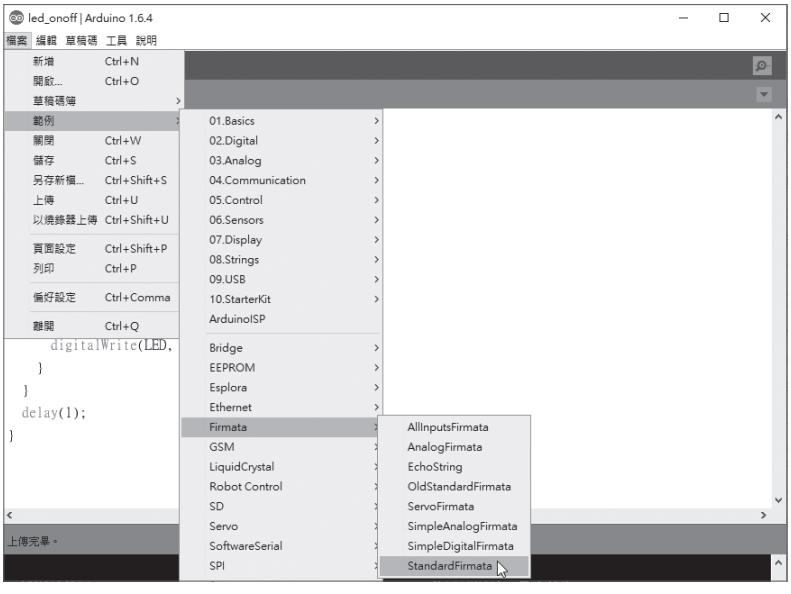
- 利用 PyFirmata 模組與 Arduino 裡的 StandardFirmata 程式碼互動。
- 利用 pySerial 模組與 Arduino 程式碼互動。

在通訊之前，可以利用 firmata_test.exe 通訊軟體，透過 USB 序列埠與 Arduino 裡的 StandardFirmata 程式碼作通訊測試。

E.1.1 使用 Firmata 通訊

Arduino IDE 內建了各種應用的 Firmata 韌體，透過 Firmata 韌體，Python 程式就可以使用 USB 序列埠與 Arduino 作傳輸，達到由 Python 控制 Arduino 的目標。

開啟 Arduino IDE 功能表的 **檔案 / 範例 / Firmata / StandardFirmata**，按 **上傳** 鈕將程式燒錄到 Arduino 控制板中。



E.1.2 下載 Firmata 測試程式

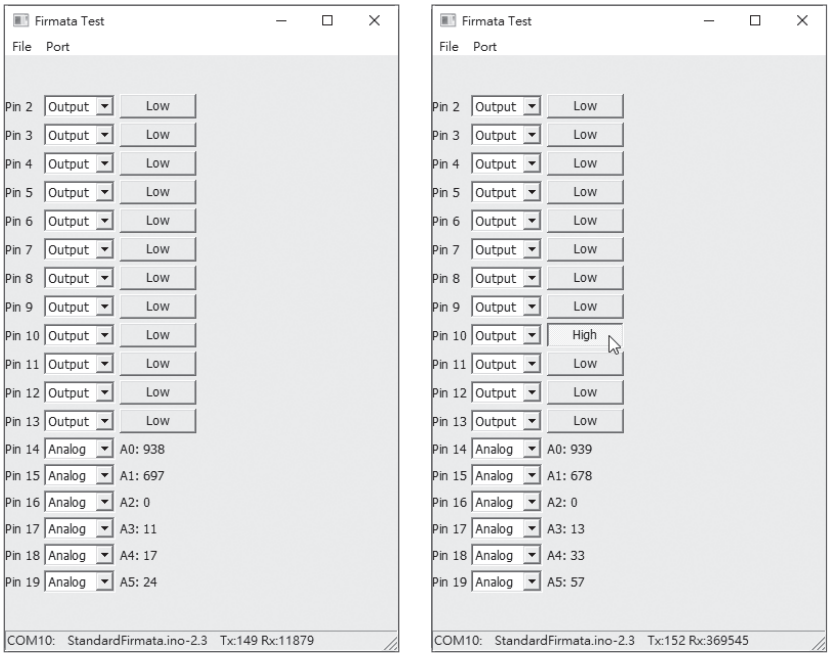
請連結「http://www.firmata.org/wiki/Main_Page」官方網址，下載 Windows 版本的 firmata_test.exe。

E.1.3 測試

將 Arduino 板和電腦連接好後，執行 firmata_test.exe，點選上方的 Port 選單，選擇 Arduino 所連接的 USB COM 埠。

連接成功之後就會出現 Arduino 每一個 Port 的目前狀態，其中 6 個 Analog 因為接腳都是浮接，其值會不斷閃爍，D2~D13 都是 Low。

按下 D10 的按鈕切換至 High 狀態時，Arduino 板子 (Sensor Board) 上內建的 D10 LED 燈會亮起來，切回 Low 就 LED 燈就會熄滅。



E.2 PyFirmata 模組

firmata_test.exe 只適合作測試，如果要在 Python 撰寫程式和 Arduino 互動，最簡便的方式是使用 PyFirmata 模組。

PyFirmata 模組可以讓 Python 程式和已上傳 Firmata 韌體的 Arduino 板子，透過 USB 序列埠作資料傳輸，也就是說可以利用 Python 程式控制 Arduino。

安裝 PyFirmata 模組

首先必須安裝 PyFirmata，語法：

```
pip install pyfirmata
```

建立 Arduino 控制板物件

以 from pyfirmata import Arduino 匯入模組之後，即可以 Arduino 方法建立 Arduino 板物件，語法：

```
Arduino板物件 = Arduino(port)
```

參數 port 代表通訊埠 (com port)。

例如：以 com10 建立 Arduino 控制板物件 board。

```
from pyfirmata import Arduino
port = 'com10'          # 指定通訊埠名稱
board = Arduino(port) # 建立 Arduino板物件
```

等待 pyFirmata 和 Arduino 同步

建立 Arduino 板物件後，最好加上適當的等待，讓 pyFirmata 和 Arduino 同步。例如：等待 1 秒鐘。

```
from time import sleep
sleep(1)
```

開啟 Iterator 避免序列溢位

如果程式中有讀取通訊埠的動作，必須開啟 Iterator 避免序列溢位。

```
from pyfirmata import Arduino,util
it = util.Iterator(board)
it.start()
```

腳位設定

在 Python 程式中，可以透過以下的方式設定 Arduino 板腳位並建立指定的物件，例如：

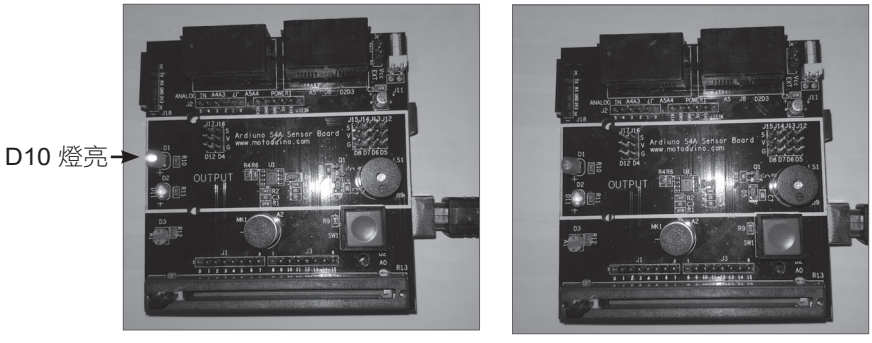
```
A2 = board.get_pin('a:2:i')     # a 代表類比，i 代表 input
D3 = board.get_pin('d:3:o')     # d 代表數位，o 代表 output
D10 = board.get_pin('d:10:p')   # p 代表 PWM
D11 = board.get_pin('d:11:s')   # s 代表 SERVO
```

關於 PyFirmata API 使用說明，可參考下列網址。

```
http://pyfirmata.readthedocs.io/en/latest/
```

範例：Led 閃爍

將 Arduino 板上的 LED 10 不斷閃爍，間隔時間為 1 秒。(註：我們使用的板子是 Sendor Board)。



Arduino 版必須上傳 (燒錄) StandardFirmata 韌體，才能不斷地接收序列埠傳送過來的資料。

Python 程式碼

程式碼：chAA\led_blink.py

```
1  #pip install pyfirmata
2  from pyfirmata import Arduino
3  from time import sleep
4  PORT="com10"
5  board=Arduino(PORT)
6  # 等待 pyFirmata 和 Arduino 同步
```

```
7   sleep(1)
8   while True:
9       # D10 未設定，預設為 OUTPUT
10      board.digital[10].write(1)
11      sleep(1)
12      board.digital[10].write(0)
13      sleep(1)
```

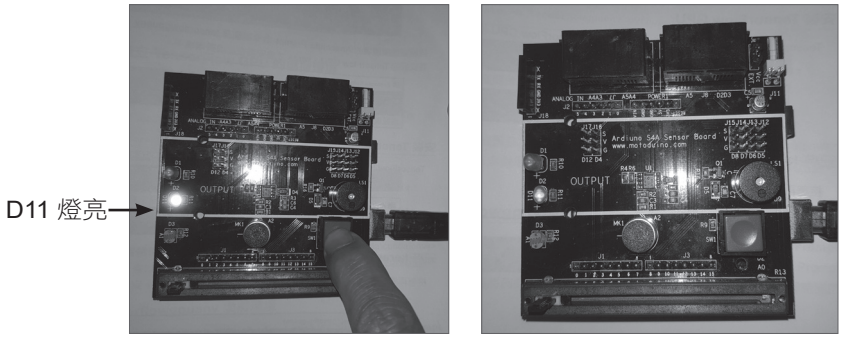
程式說明

- 1　記得必須安裝 pyfirmata。
- 5　以 com10 建立 Arduino 板物件。
- 7　等待一秒，讓 pyFirmata 和 Arduino 同步。
- 8-13　不斷寫入序列埠。
- 10-11　Pin 10 未設定為輸入或輸出時，因未設定時預設為輸出埠，因此 D10 為輸出。D10 送出 1，將 LED 燈點亮，並暫停一秒。
- 12-13　D10 送出 0，將 LED 燈熄滅，並暫停一秒。

上一個範例只用到數位腳位的輸入，我們再來看看數位腳位輸入的用法。

範例：按鈕控制 Led 燈

按下 Arduino 板的 D2 按鈕，LED 11 燈點亮，放開按鈕 LED 11 燈熄滅。



(註：我們使用的板子是 Sendor Board，Arduino 版必須上傳 StandardFirmata 韌體。)

Python 程式碼

程式碼：chAA\ReadButton.py

```
1   from pyfirmata import Arduino,util
2   from time import sleep
3   port = 'com10'
4   board = Arduino(port)
5   # 等待 pyFirmata 和 Arduino 同步
6   sleep(1)
7
8   # 開啟 Iterator 避免序列溢位
9   it = util.Iterator(board)
10  it.start()
11
12  D2 = board.get_pin('d:2:i')  # D2 為 intput
13  try:
14      while True:
15          p = D2.read() # 讀取 D2
16          print(p)
17          if p==True:
18              board.digital[11].write(1)
19          else:
20              board.digital[11].write(0)
21  except KeyboardInterrupt:
22      board.exit()
```

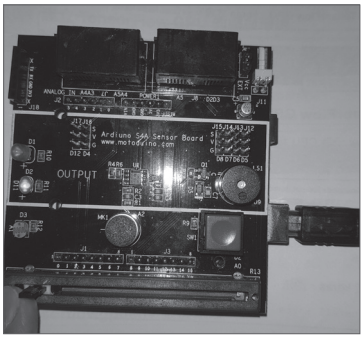
程式說明

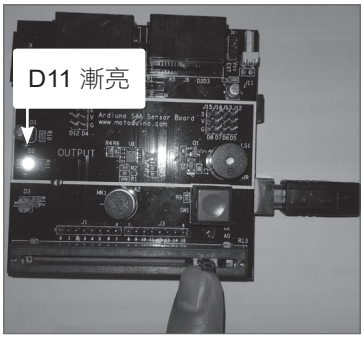
- 1-6　以 com10 建立 Arduino 板物件並等待同步。
- 9-10　本例會讀取通訊埠，必須開啟 Iterator 避免序列埠產生溢位。
- 12　設定 D2 為數位輸入腳位，在 Sensor Board 這個腳位是接到按鈕上。
- 13,21　以 try…except…補捉錯誤，所以遇到 except 的時候，就會執行 board.exit()，停用這個 board 的 COM port 資源。
- 14-20　不斷讀取 D2 腳位。
- 17-18　按下 D2 按鈕，D11 送出 1，將 LED 燈點亮。
- 19-20　放開 D2 按鈕，D11 送出 0，將 LED 燈熄滅。

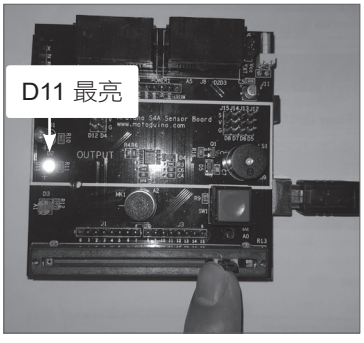
最後再來看看類比腳位和 PWM 腳位的用法。

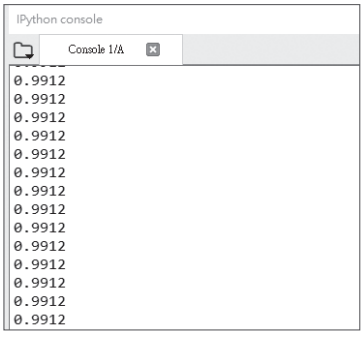
範例：滑桿控制 Led 燈亮度

調整 Arduino 板上的 A0 滑桿，LED 11 亮度隨著滑桿輸入值大小變化。









(註：我們使用的板子是 Sendor Board，Arduino 版必須上傳 StandardFirmata 韌體。)

Python 程式碼

程式碼：chAA\PWM.py

```
1   from pyfirmata import Arduino, util
2   from time import sleep
3   port = 'com10'
4   board = Arduino(port)
5   # 等待 pyFirmata 和 Arduino 同步
6   sleep(3)
7
8   # 開啟 Iterator 避免序列溢位
9   it = util.Iterator(board)
10  it.start()
11
12  A0 = board.get_pin('a:0:i')    # A0 為 Analog 輸入埠
13  D11 = board.get_pin('d:11:p') # D11 為 PWM
14
15  try:
16      while True:
17          p = A0.read()  # 讀取 A0 埠
18          print(p)
19          D11.write(p)
20  except KeyboardInterrupt:
21      board.exit()
```

程式說明

- 12-13　建立 A0 物件設定以 A0 為輸入埠，建立 D11 物件設定以 D11 為 PWM 輸出腳位。
- 16-17　不斷讀取 A0 腳位。
- 18　顯示 A0 腳位讀取的數值，請注意：數值大小是介於 0.0 ~ 1.0 之間。
- 19　將讀取的值送到 D11 腳位，因為 Sensor Board 第 11 腳位接到 Led 燈，因此調整滑桿就可以控制 Led 燈的亮度變化。

E.3 pySerial 模組

使用 PyFirmata 模組將 Python 程式和已上傳 Firmata 韌體的 Arduino 板子，透過 USB 序列埠作資料傳輸，雖然使用方便，但我們並無法靈活自行撰寫 Arduino 程式來控制，最好的方式就是不論 Python 或 Arduino 程式，都是由自己來撰寫，當然這需要對 Arduino 程式有一定的基礎。

要在 Python 中撰寫程式控制 Arduino，必須在電腦中安裝 Python 的 pySerial 模組，同時也要撰寫和上傳 Arduino 程式，透過 pySerial 模組和 Arduino 進行通訊。

E.3.1 安裝 pySerial 模組

安裝 pySerial 模組語法：

```
pip install pyserial
```

以 import serial 匯入後即可以 Serial 方法初始化序列通訊埠，語法：

```
序列通訊物件 = serial.Serial(COM_PORT, BAUD_RATES)
```

- 參數 COM_PORT 代表通訊埠，BAUD_RATES 代表傳輸速率。

例如：以 com10、9600 傳輸速率建立序列通訊埠物件 s。

```
import serial          # 匯入 pySerial 模組
COM_PORT = 'com10'     # 指定通訊埠名稱
BAUD_RATES = 9600      # 設定傳輸速率
s = serial.Serial(COM_PORT, BAUD_RATES)    # 初始化序列通訊埠
```

E.3.2 pySerial 的方法

PySerial 提供許多的方法：

write 方法傳送資料

write 方法的語法為：

```
write(data)
```

write 方法會傳送指定的資料到序列通訊埠，並返回傳送位元組數目。data 為傳送的資料，資料型別為位元組 (bytes)。

注意：因為 Python 3 字串是以 Unicode 編碼，因此在輸出到序列通訊埠時必須呼叫 encode() 將傳送資料轉成 byte 編碼。例如：

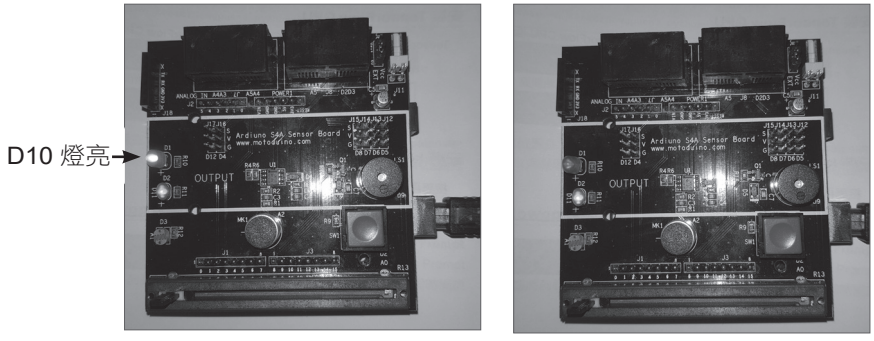
```
s.write('Hello'.encode())
```

或直接以 b 指定為 byte 型別。

```
s.write(b'Hello')
```

範例：Led 閃爍

將 Arduino 板上的 LED 10 不斷閃爍，間隔時間為 1 秒。(註：我們使用的板子是 Sendor Board)



Arduino 程式碼

Arduino 版必須上傳 (燒錄) 程式，不斷地接收序列埠傳送過來的資料。

程式碼：chAA\led_onoff.ino

```
1   int LED = 10; // Pin 10
2   void setup() {
3      Serial.begin(9600);
4   }
5
6   void loop() {
7     if (Serial.available() > 0) {
8       if (Serial.read() == 'H') {
9         digitalWrite(LED, HIGH); // 燈亮
10      }else{
11        digitalWrite(LED, LOW);  // 燈熄
12      }
13    }
```

```
14    delay(1);
15  }
```

程式說明

- 1　設定 LED 燈為 Pin 10。
- 3　設定傳輸速率為 9600。
- 7-13　如果有接收到字元。
- 8-9　接收到「H」字元，將 LED 燈點亮。
- 10-12　接收到「L」字元，將 LED 燈熄滅。
- 14　暫停一秒。

Python 程式碼

Python 程式不斷地對序列埠送出 'H' 和 'L' 字元，每次間隔時間為一秒。

程式碼：chAA\led_onoff.py

```
1   #pip install pyserial
2   import serial
3   s=serial.Serial("com10",9600)
4   from time import sleep
5   while True:
6       s.write('H'.encode()) # 傳送 H 字元
7       sleep(1)
8       s.write('L'.encode()) # 傳送 L 字元
9       sleep(1)
```

程式說明

- 1　記得要安裝 pyserial 模組。
- 2　匯入 serial 模組。
- 3　初始化序列通訊埠，並建立通訊物件 s。
- 4　匯入 time 模組。
- 5-9　以 write 方法持續對序列埠送出 'H' 和 'L' 字元，每次間隔時間為一秒鐘。

其實在實務應用中，判斷的接收字元會是字串而不是一個字元，下面的範例就改用字串方式來控制 Led 閃爍。

範例：使用字串控制 Led 閃爍

使用接收字串，控制 Arduino 板上的 LED 10 不斷閃爍，間隔時間為 1 秒。

執行結果同前面範例。

Arduino 程式碼

Arduino 版必須上傳 (燒錄) 程式，不斷地接收序列埠傳送過來的資料。

程式碼：chAA\readString\readString.ino

```
1   int LED = 10;
2   String ReadString;
3   char ch;
4   void setup() {
5      Serial.begin(9600);
6   }
7
8   void loop() {
9      while( Serial.available() > 0) {
10        ch=Serial.read(); // 讀取序列埠
11        ReadString +=ch; // 將字串組合
12        delay(0.01);
13     }
14     if (ReadString.length()>0){ // 顯示接收字串
15         Serial.println("ReadString=" + ReadString);
16     }
17     if (ReadString == "ON") { // Led 點亮
18        digitalWrite(LED, HIGH);
19        ReadString="";
20     }
21     if (ReadString == "OFF") { // Led 熄滅
22        digitalWrite(LED, LOW);
23        ReadString="";
24     }
25  }
```

程式說明

- 1　設定 LED 燈為 Pin 10。
- 5　設定傳輸速率為 9600。
- 9-13　如果有接收到字元，將字元組合成字串。

- 14-16　如果有接收到字元，顯示接收字串。
- 17-20　接收到「ON」字串，將 LED 燈點亮，同時清除 ReadString 字串。
- 21-24　接收到「OFF」字元，將 LED 燈熄滅，同時清除 ReadString 字串。

Python 程式碼

Python 程式不斷地對序列埠送出 'H' 和 'L' 字元，每次間隔時間為一秒。

程式碼：chAA\led_onoff_str.py

```
1   import serial
2   s=serial.Serial("com10",9600)
3   from time import sleep
4   while True:
5       s.write('ON'.encode()) # 傳送 ON 字串
6       sleep(1)
7       s.write('OFF'.encode()) # 傳送 OFF 字串
8       sleep(1))
```

程式說明

- 2　初始化序列通訊埠，並建立通訊物件 s。
- 4-8　不斷地以 write 方法持續對序列埠送出 'ON' 和 'OFFL' 字串，每次間隔時間為一秒鐘。

read 方法讀取資料

read 方法的語法為：

```
read(size=1)
```

read 方法會以位元組 (bytes) 讀取通訊埠並傳回讀取的字元，size 可以指定多少個字元，最多會讀取 0~size 個字元 。

readline 方法讀取一列資料

readline 方法的語法為：

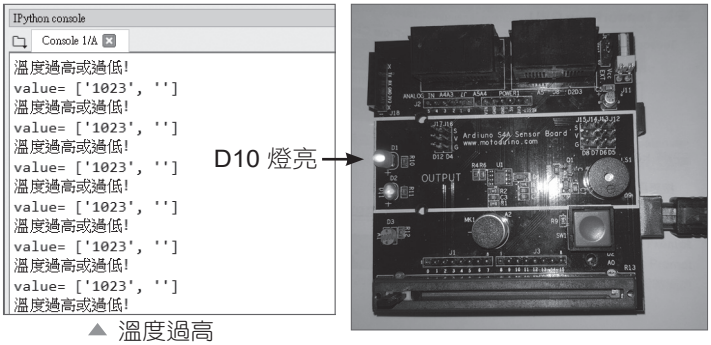
```
readline()
```

readline 方法會以位元組 (bytes) 讀取通訊埠並傳回一列讀取的字元 。

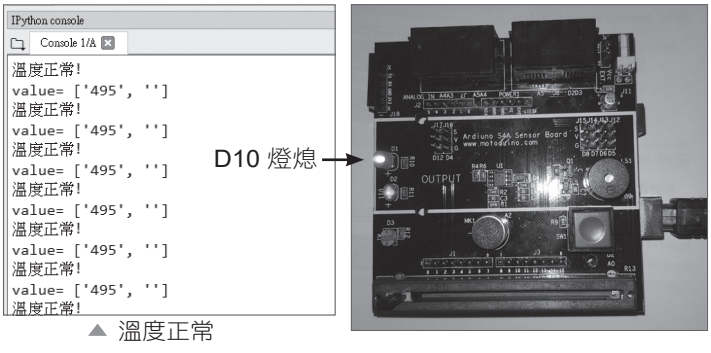
PySerial 模組 API 用法，可參考下列網址「http://pyserial.readthedocs.io/en/latest/pyserial_api.html」。

範例：偵測 A0 埠當狀況異常時以 Led 顯示

偵測 A0 (搖桿)，當 A0 值過高 (>900) 或過低 (<200) ，在 Python 程式中顯示「溫度過高或過低 !」訊息，並將 Arduino 板上的 LED 10 點亮，表示目前的偵測狀況出現異常，否則就顯示「溫度正常 !」訊息，並將 Arduino 板上的 LED 10 熄滅，表示目前的偵測狀況已回復正常。



▲ 溫度過高



▲ 溫度正常

Arduino 程式碼

Arduino 版必須上傳 (燒錄) 程式，程式會不斷地傳送 A0 的值，並接收序列埠傳送過來的資料，如果接收到「ON」字串將 Led 點亮，接收到「OFF」字串將 Led 熄滅。

程式碼：chAA\readA0\readA0.ino

```
1   int LED = 10;
2   String ReadString;
3   char ch;
4   void setup() {
5       Serial.begin(9600);
```

```
6        // 設定 A0 輸入
7        pinMode(A0, INPUT);
8    }
9
10   void loop()
11   {
12      int val = analogRead(A0); // 讀取 A0 埠
13      // 傳送資料至電腦
14      Serial.println(val);
15      while( Serial.available() > 0) {
16         ch=Serial.read(); // 讀取序列埠
17         ReadString +=ch;  // 將字串組合
18         delay(0.01);
19      }
20      if (ReadString == "ON") { // Led 點亮
21         digitalWrite(LED, HIGH);
22         ReadString="";
23      }
24      if (ReadString == "OFF") { // Led 熄滅
25         digitalWrite(LED, LOW);
26         ReadString="";
27      }
28      if (ReadString.length()>0){ // 顯示接收字串
29          ReadString="";
30      }
31   }
```

程式說明

- 7　設定 A0 為輸入埠。
- 12　不斷讀取 A0。
- 14　將 A0 資料送到序列埠。
- 15-19　如果有接收到字元，將接收到字元組合成字串。
- 20-22　接收到「ON」字串，將 LED 燈點亮，同時清除 ReadString 字串。
- 24-27　接收到「OFF」字元，將 LED 燈熄滅，同時清除 ReadString 字串。
- 28-30　清除剛才接收的 ReadString 字串。

Python 程式碼

Python 程式必須不斷地讀取 A0 埠，當 A0 > 900 或 A0 < 200 時，發出警告訊息，並對序列埠送出 'ON' 字串；否則就發出正常訊息並對序列埠送出 'OFF' 字串。

程式碼：chAA\readLine.py

```
1   import serial
2   s=serial.Serial("com10",9600)
3   while True:
4       # 讀取序列埠並將字元轉換為字串
5       value=s.readline().decode("utf-8")
6       # 去除跳列字元
7       value=value.split('\r\n')
8       print("value=",value) # 顯示讀取值
9       value=int(value[0])
10      if (value> 900 or value<200):
11          print("溫度過高或過低！")
12          s.write(b'ON') # 傳送 ON 字串
13      else:
14          print("溫度正常！")
15          s.write(b'OFF') # 傳送 OFF 字串
```

程式說明

- 5　讀取序列通訊埠，實際會讀到 Arduino 傳送過來 A0 埠的值，將讀取序列埠的字元以 decode("utf-8") 方法轉換為字串。
- 7　去除跳列字元。
- 9　value=int(value[0])，value[0] 為實際讀取的資料，將它轉換為整數型別。
- 10-12　溫度過高或過低的處理。
- 12　傳送 ON 字串給 Arduino。
- 13-15　溫度正常的處理。
- 15　傳送 OFF 字串給 Arduino。

Memo