

# **ARDUINO + ESP8266 應用**

**通過 Wifi 並建置網頁伺服器，讓手機控制 Arduino**

課程製作：盧祉蘋

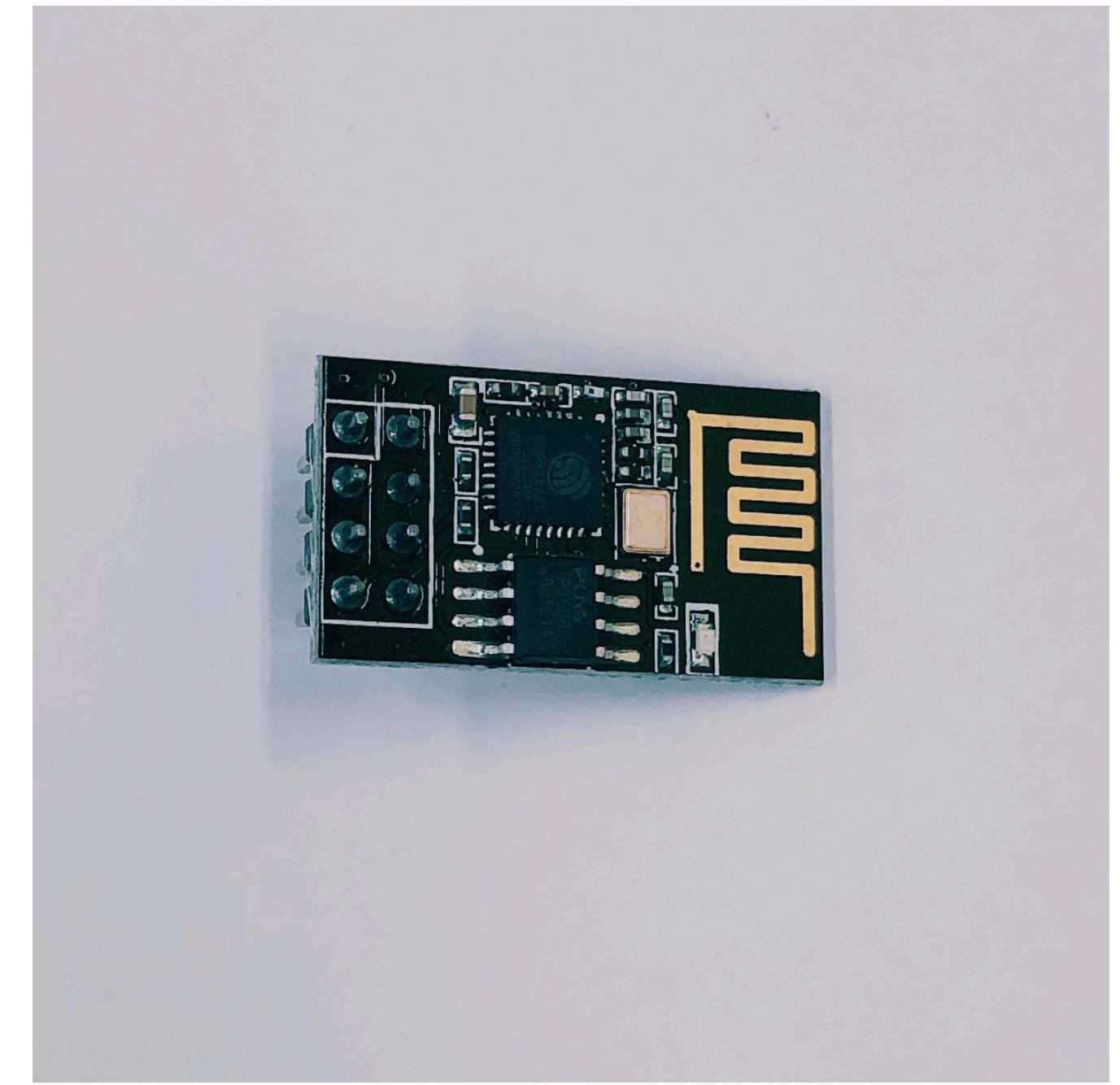
# 主要硬體介紹



Arduino UNO



USB轉TTL轉換器  
(FT232BL)



ESP8266-01  
Wifi模組

# 材料說明

杜邦線(母-公) x 7	木製盒 x 1	焊接工具
杜邦線(公-公) x 2	麵包板 x 1	膠水(熱融膠)
Arduino UNO x 1	Arduino USB數據線 x 1	絕緣膠帶 x 1
ESP8266 x 1	玻璃罐子(加蓋) x 1	LED串 x 1
USB轉TTL轉換器 x1	砂紙 x 1	

# 安裝須知

## 必讀

- 安裝指標(紅框)裡的數字與當頁安裝步驟相同，請按順序操作。
- 安裝電路中，端口說明格式為[物件] [端口]，如ESP TX指ESP8266的TX端口。
- 文中所指的ESP為ESP8266 Wifi模組、轉換器意指USB轉TTL轉換器。
- 軟體安裝文檔建議放置在桌面方便定位。
- 請遵循安裝順序，不可跳頁或無視任一安裝步驟！

# 總體結構



HTTP

A double-headed arrow indicating the communication protocol between the client and the ESP8266 module.

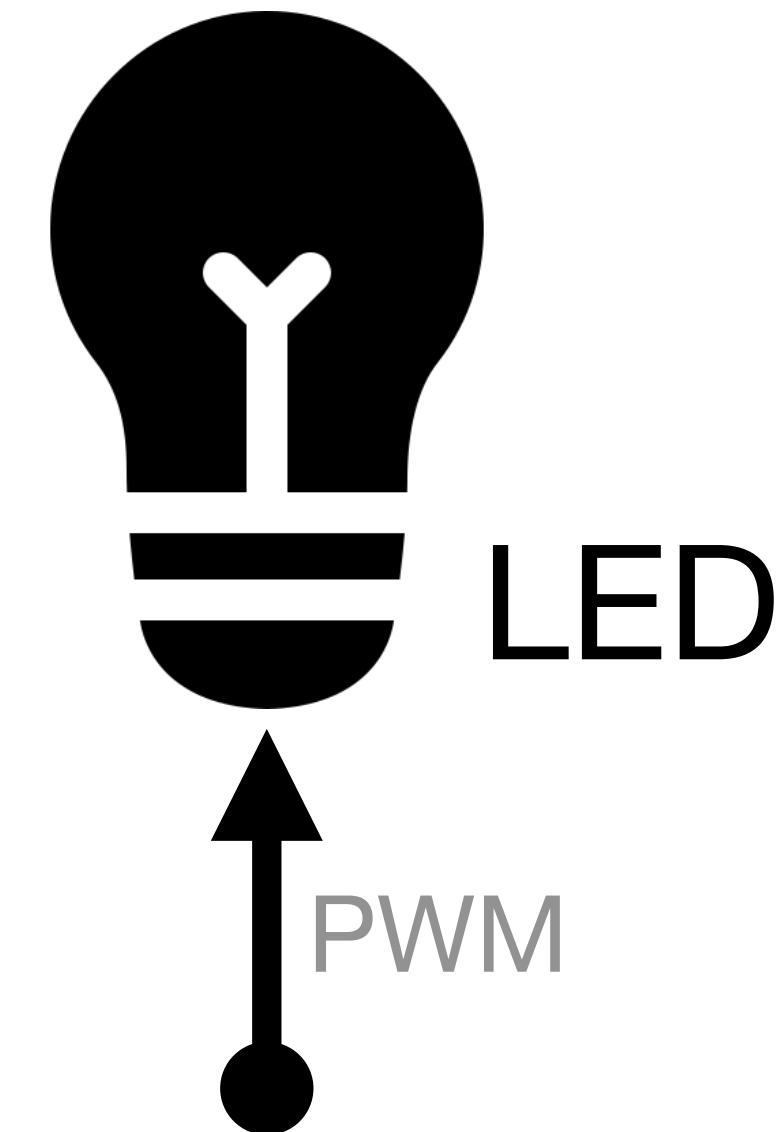
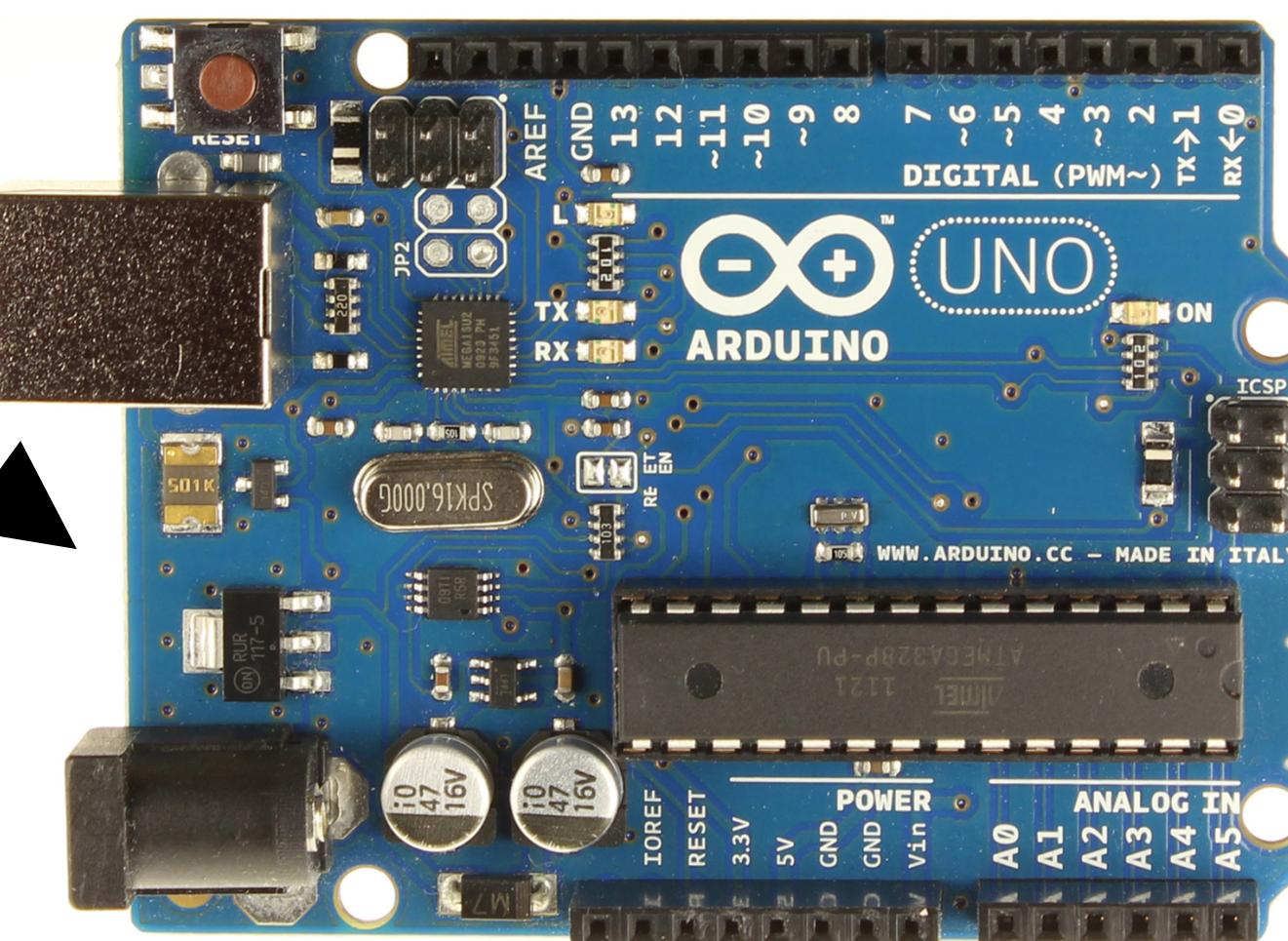
呼吸燈範圍

A blue header bar containing the text "呼吸燈範圍" (Breath-Light Range).



序列埠

An arrow pointing from the "序列埠" text to the TX and RX pins of the Arduino Uno board.



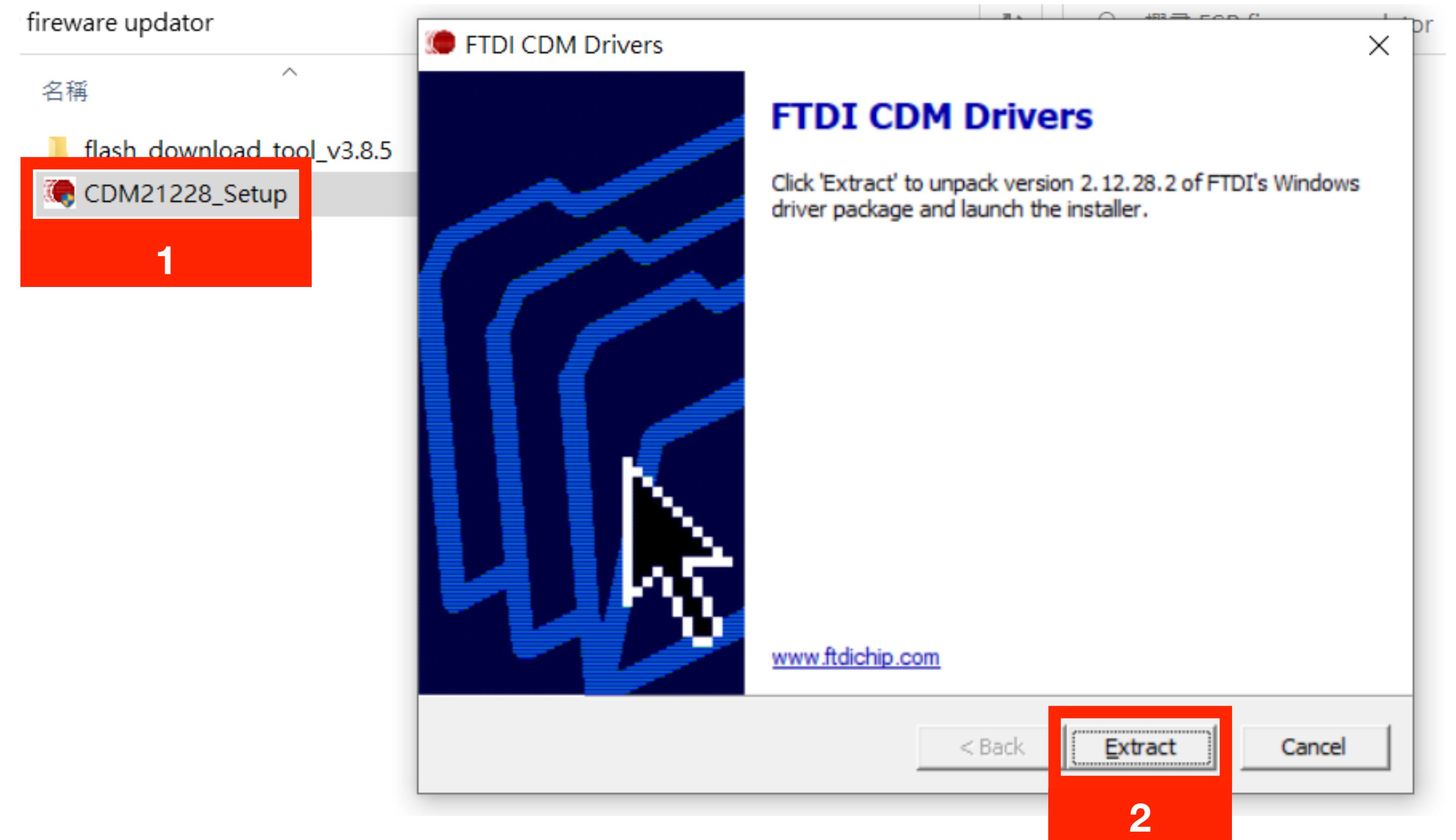
PWM

An arrow pointing from the "PWM" text to the digital pin connected to the Arduino's PWM output and the LED.

**更新ESP8266韌體  
及更改鮑率**

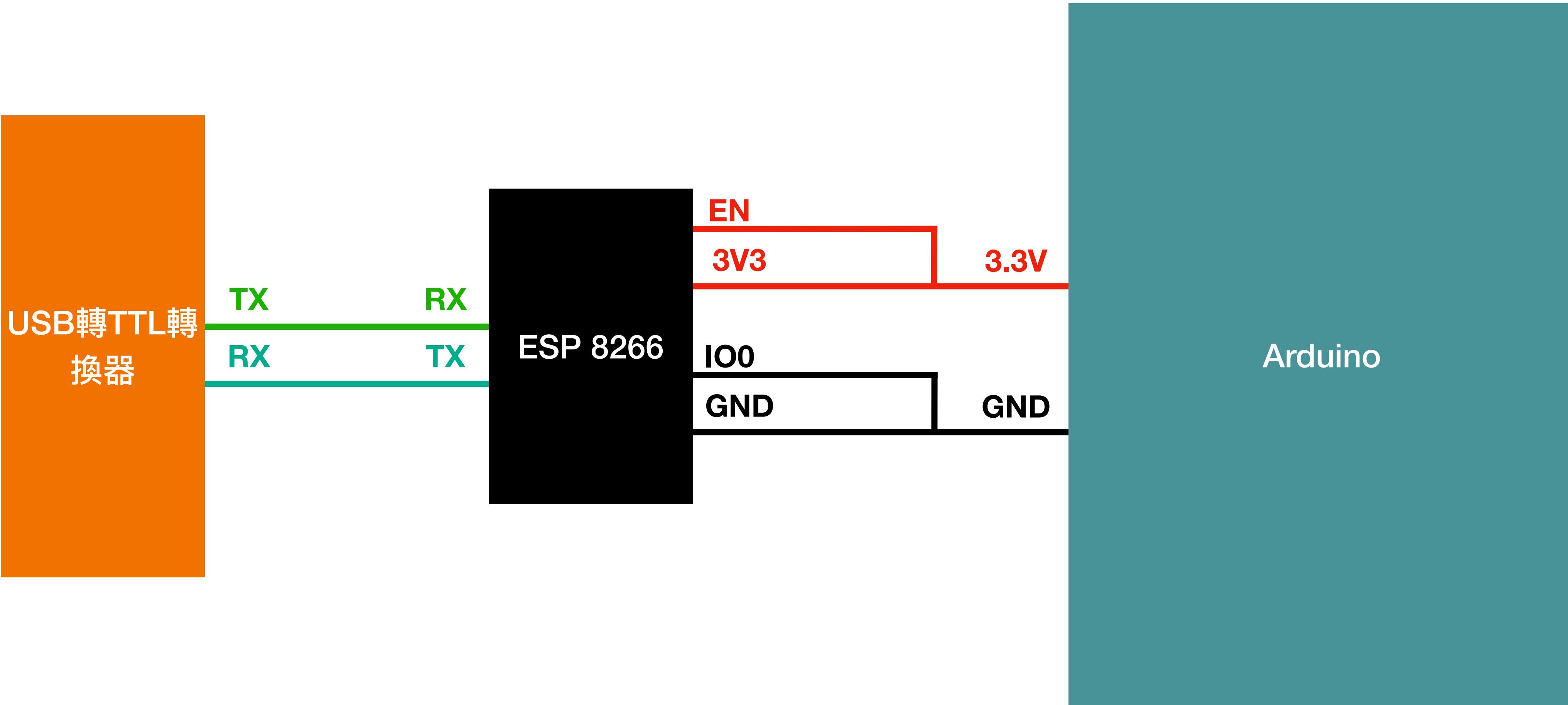
# 安裝FTDI驅動 使電腦可讀取轉接器

1. 雙擊  
[CDM21228\_Setup]
2. 按Extract或Next等步驟  
直到安裝完成
3. 重啟電腦



# 安裝ESP8266模組

## ESP8266電路連接

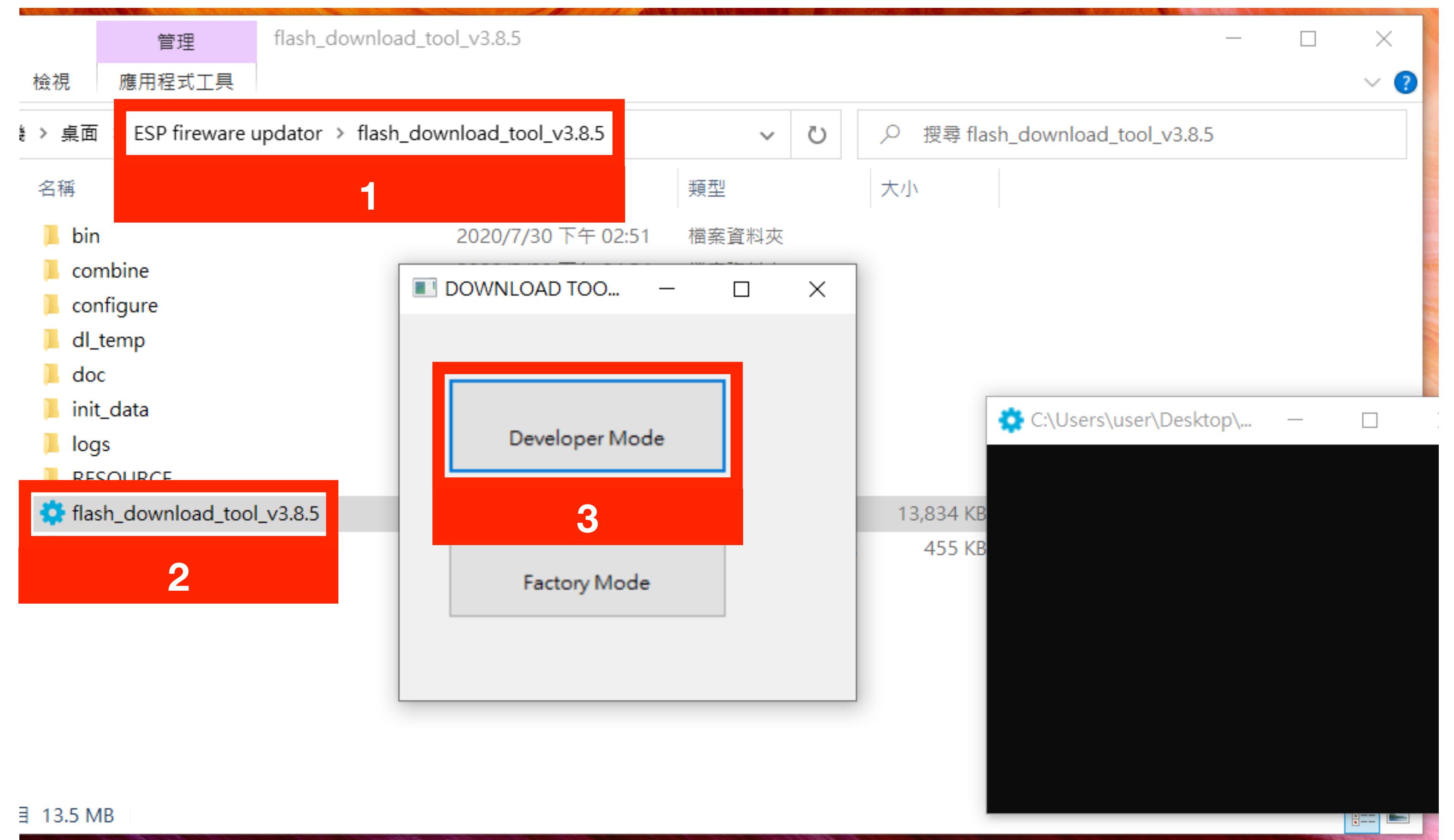


# 使用閃存更新軟體 開啟介面

1. 前往  
[flash\_download\_to  
ol\_v3.8.5]文件夾

2. 點擊  
[flash\_download\_to  
ol\_v3.8.5]

3. 點擊 [Developer  
Mode]



# 使用閃存更新軟體

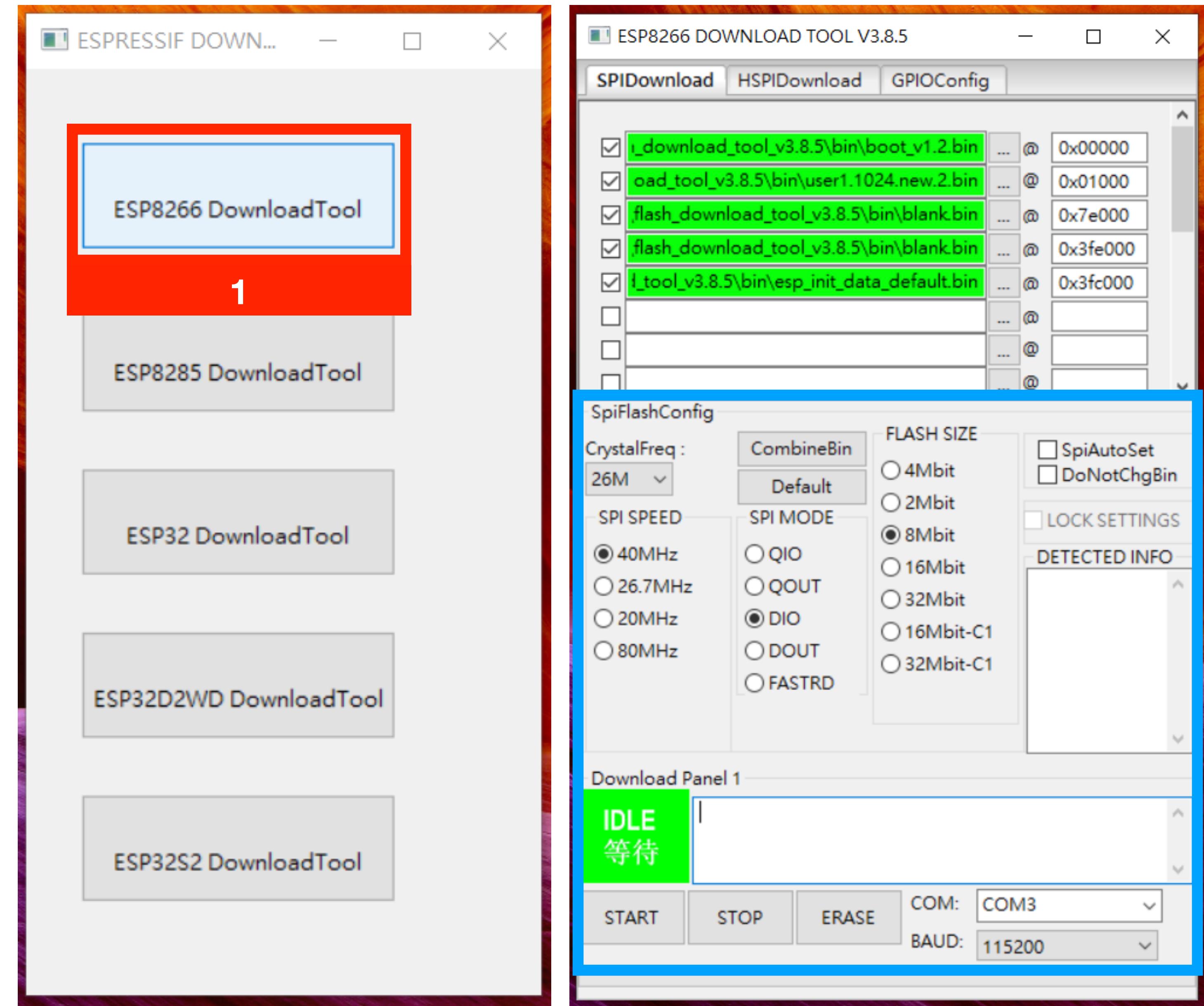
## 開啟介面

1. 點擊[ESP8266  
Download Tool]

2. 將轉換器插入電腦

3. 更換下載設定（藍框內須  
完全一樣，除了COM）

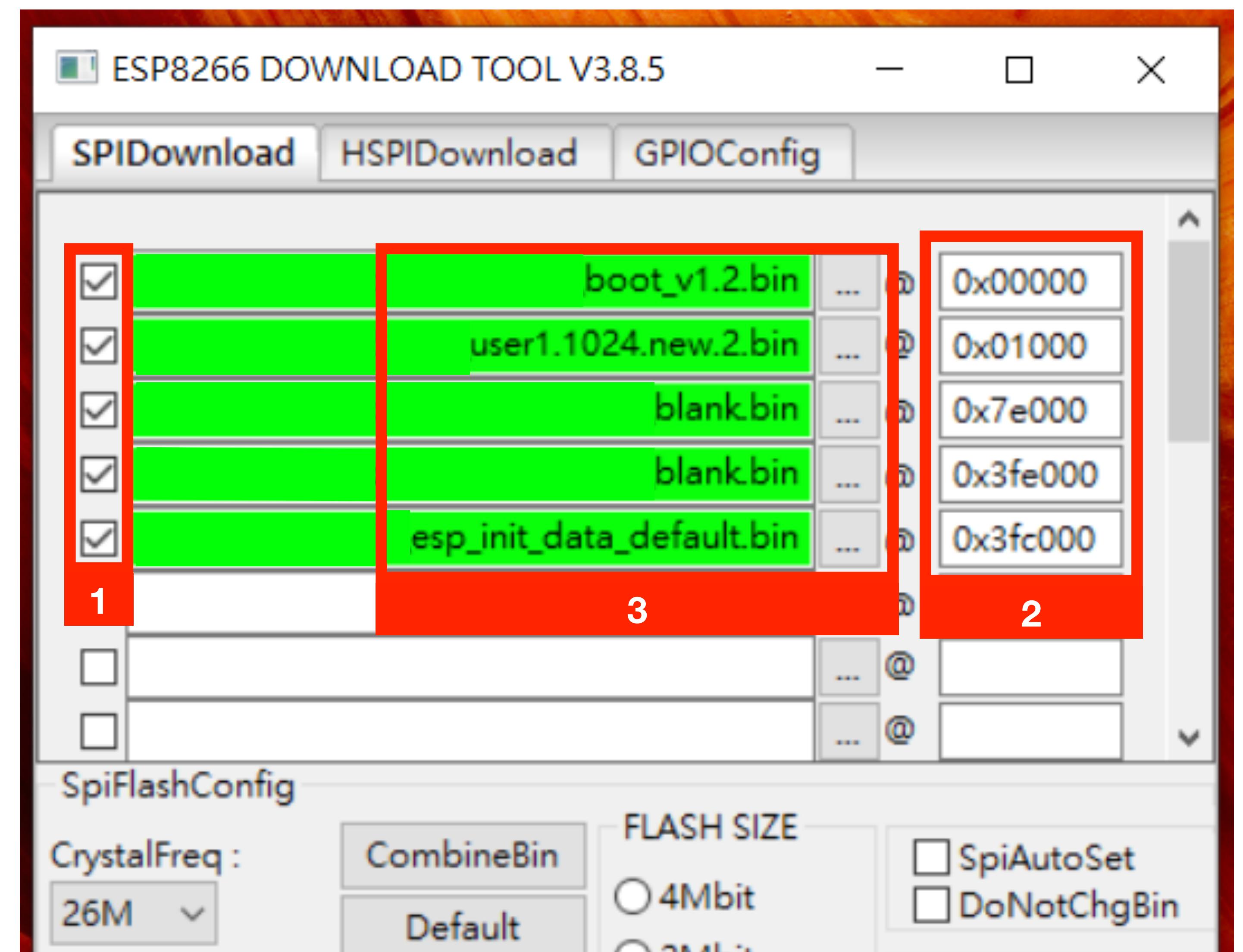
4. COM口未必是3喔！



# 使用閃存更新軟體

## 選擇韌體文檔

1. 點擊前五行的checkbox
2. 輸入閃存地址
3. 點擊文件選擇鈕[...]，並按順序選擇文件
4. 韌體文件位置於  
\flash\_download\_tool\_  
v3.8.5\bin



# 使用閃存更新軟體 格式化ESP8266

1. 確認ESP8266已完成接線

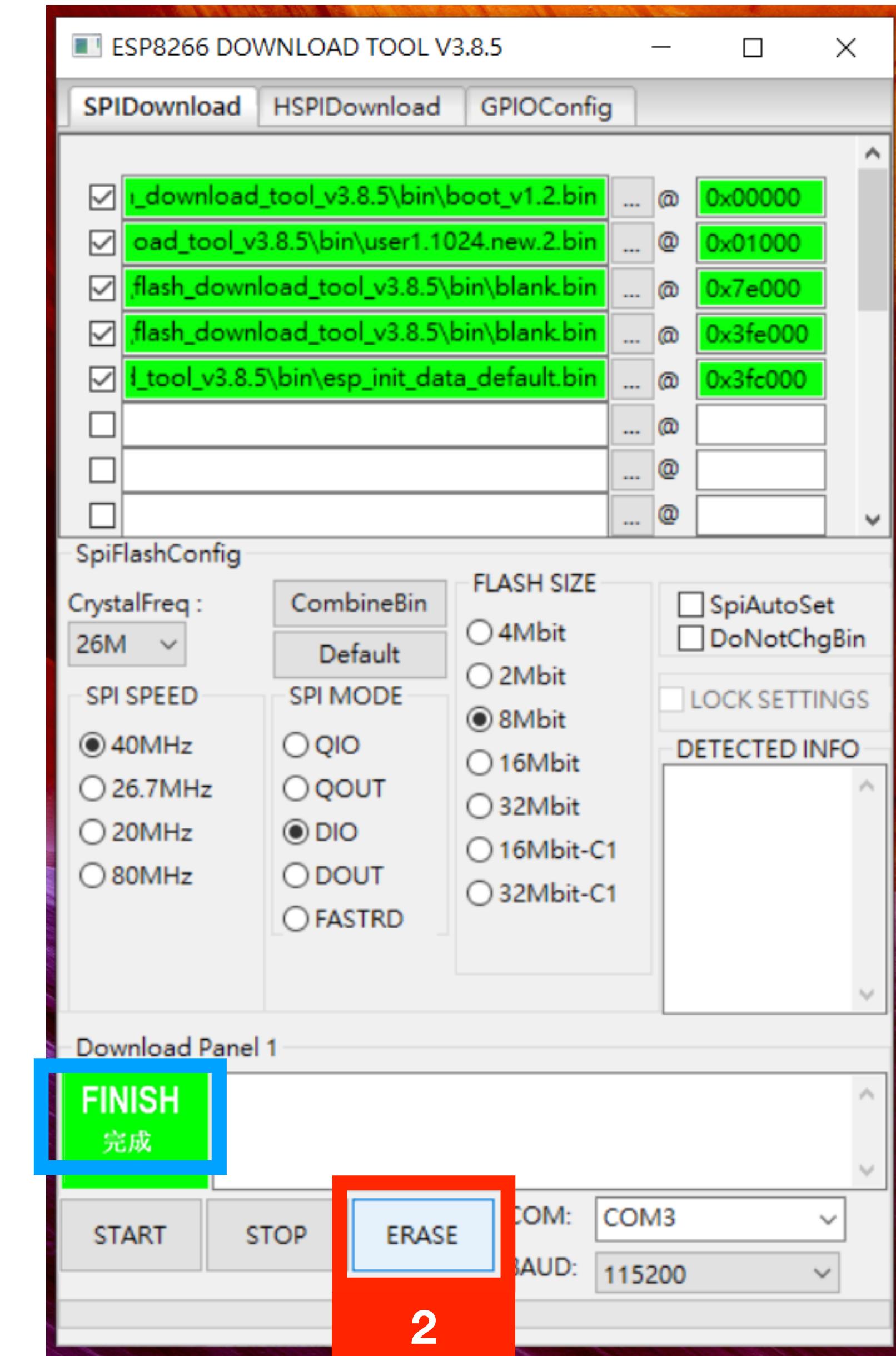
(Arduino不接電)

2. 點擊[ERASE]

3. 顯示[等待上電同步]後將

Arduino上電

4. 完成後應顯示藍框內的FINISH



# 使用閃存更新軟體

上傳韌體文件

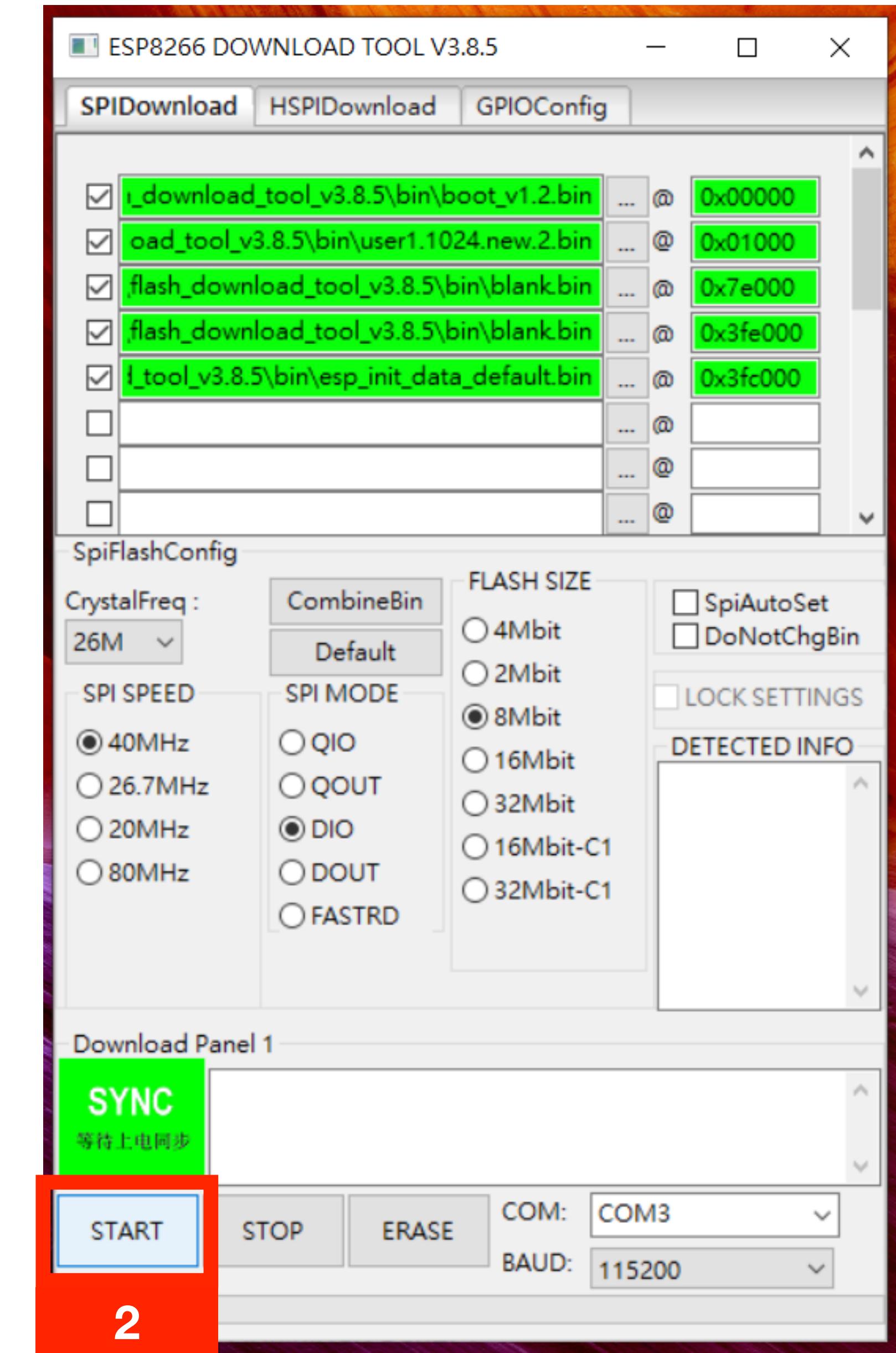
1. 將Arduino斷電

2. 點擊[START]

3. 顯示[等待上電同步]後將  
Arduino上電

4. 完成後應顯示FINISH

5. 將Arduino斷電

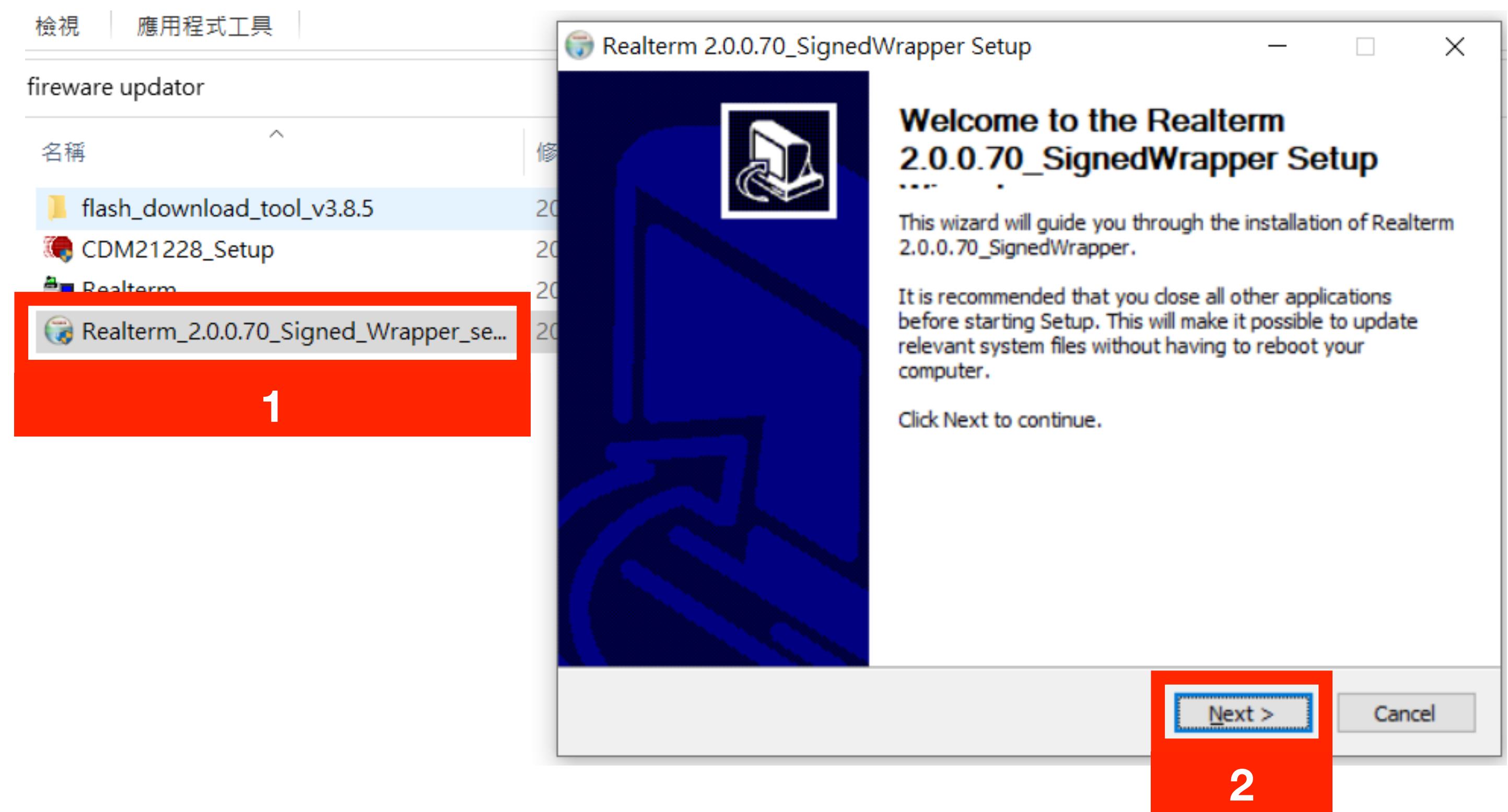


# 安裝Realterm

## 使電腦可與ESP8266溝通

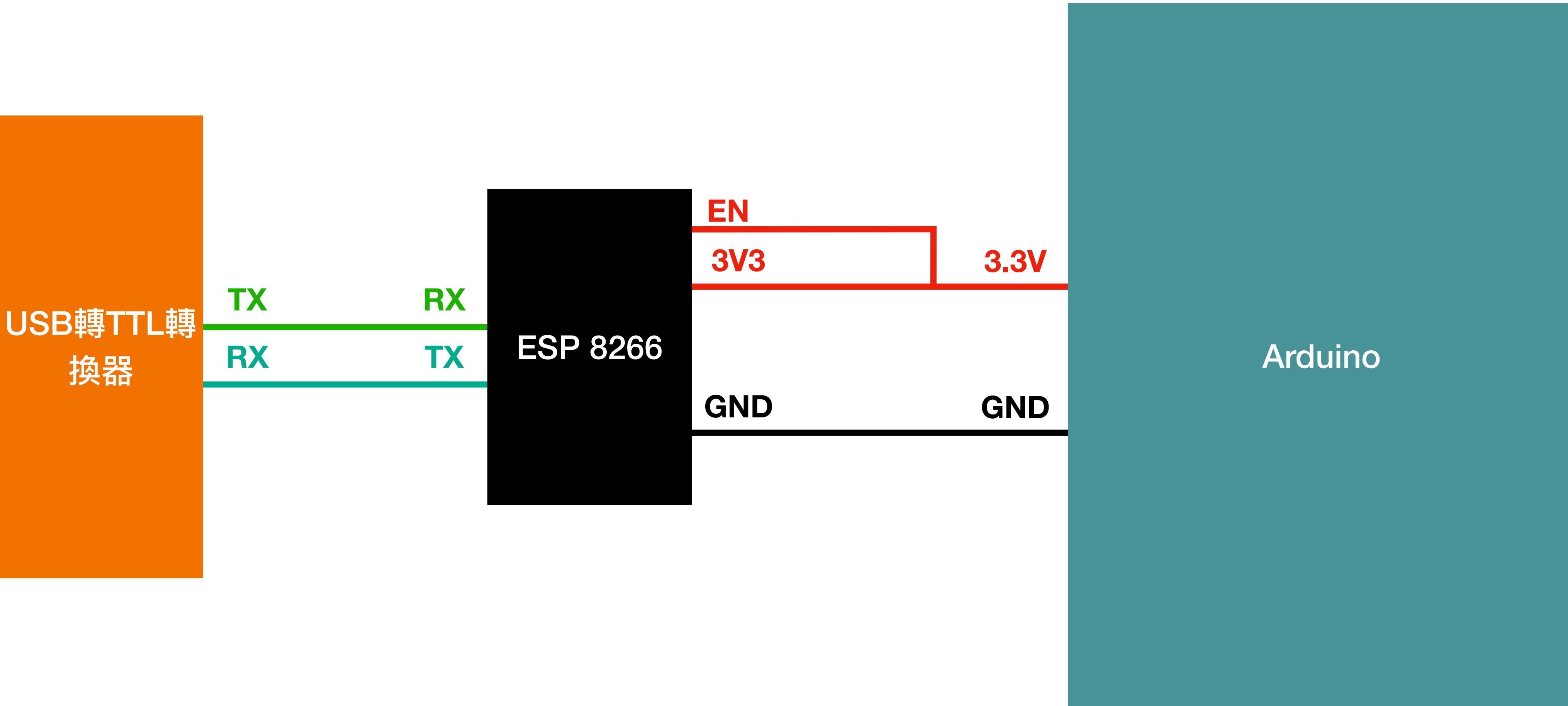
1. 雙擊[Realterm\_...\_setup]

2. 點擊Next直到安裝完成



# 更新ESP8266電路

移除ESP8266上IO0的接地線

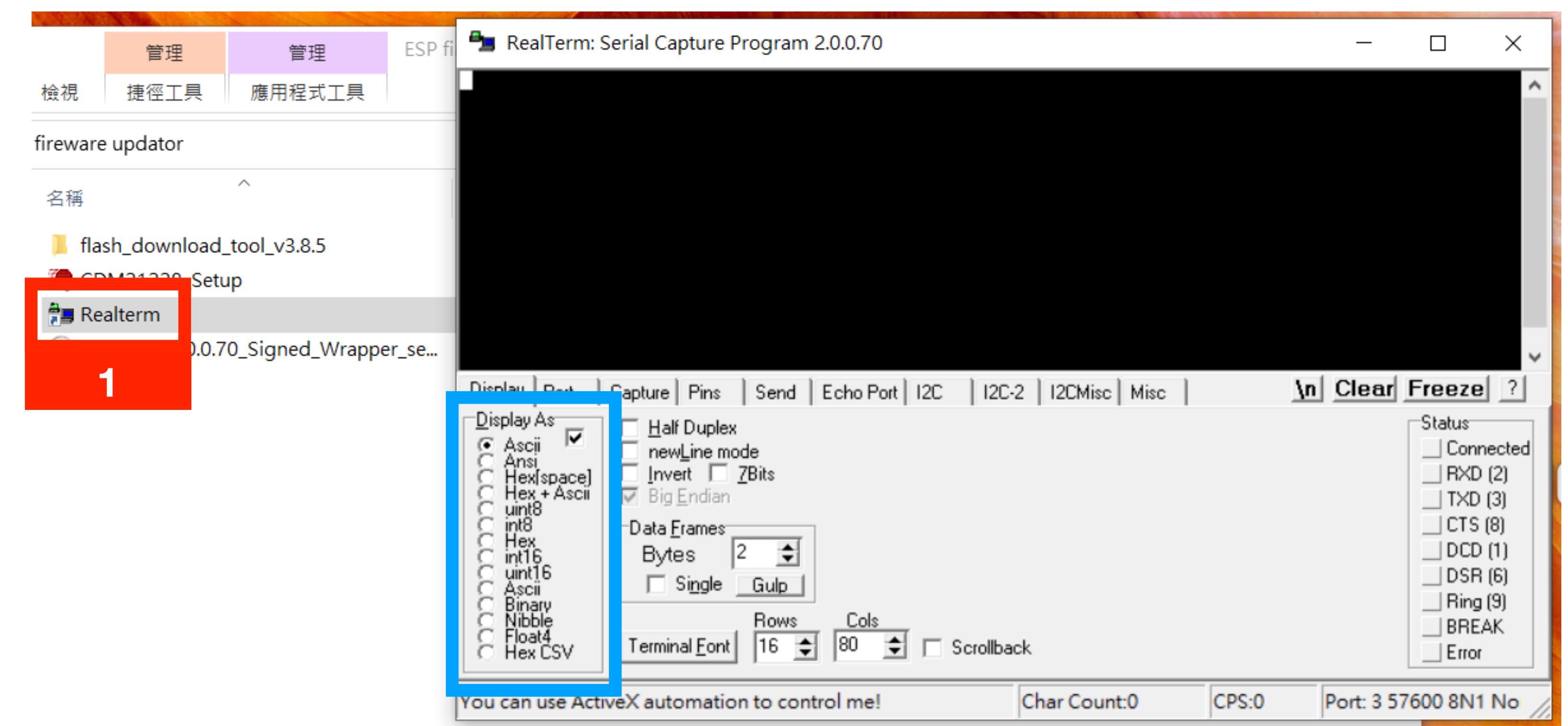


# 使用Realterm

## 確認顯示模式

1. 打開Realterm(新安裝的可能在桌面)

2. 確認[Display As]設定如藍框



# 使用Realterm

## 設定監聽端口

1. 前往[Port]

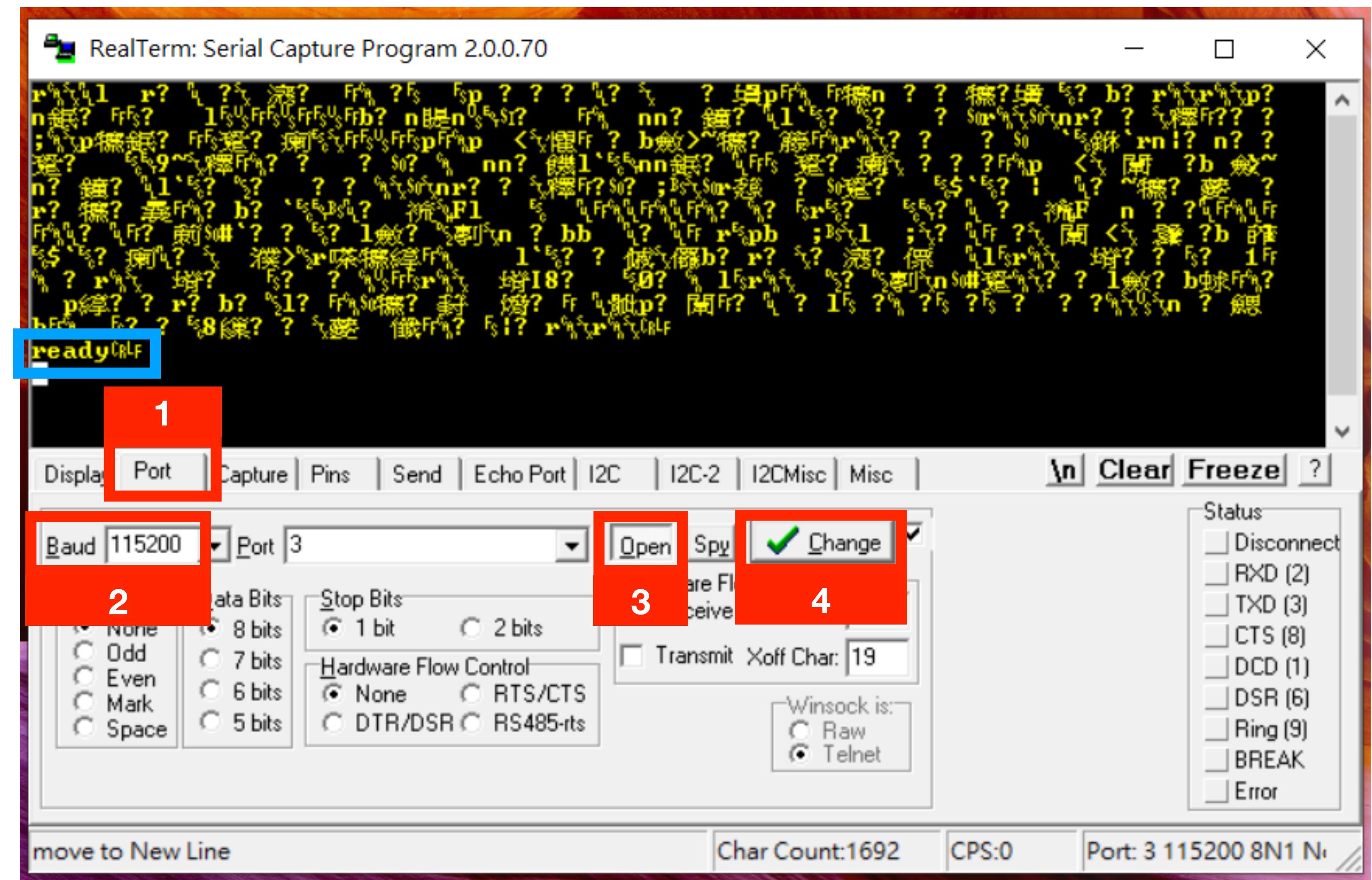
2. 更改鮑率至115200

3. 選擇端口

4. 按[Change]

5. 將Arduino上電

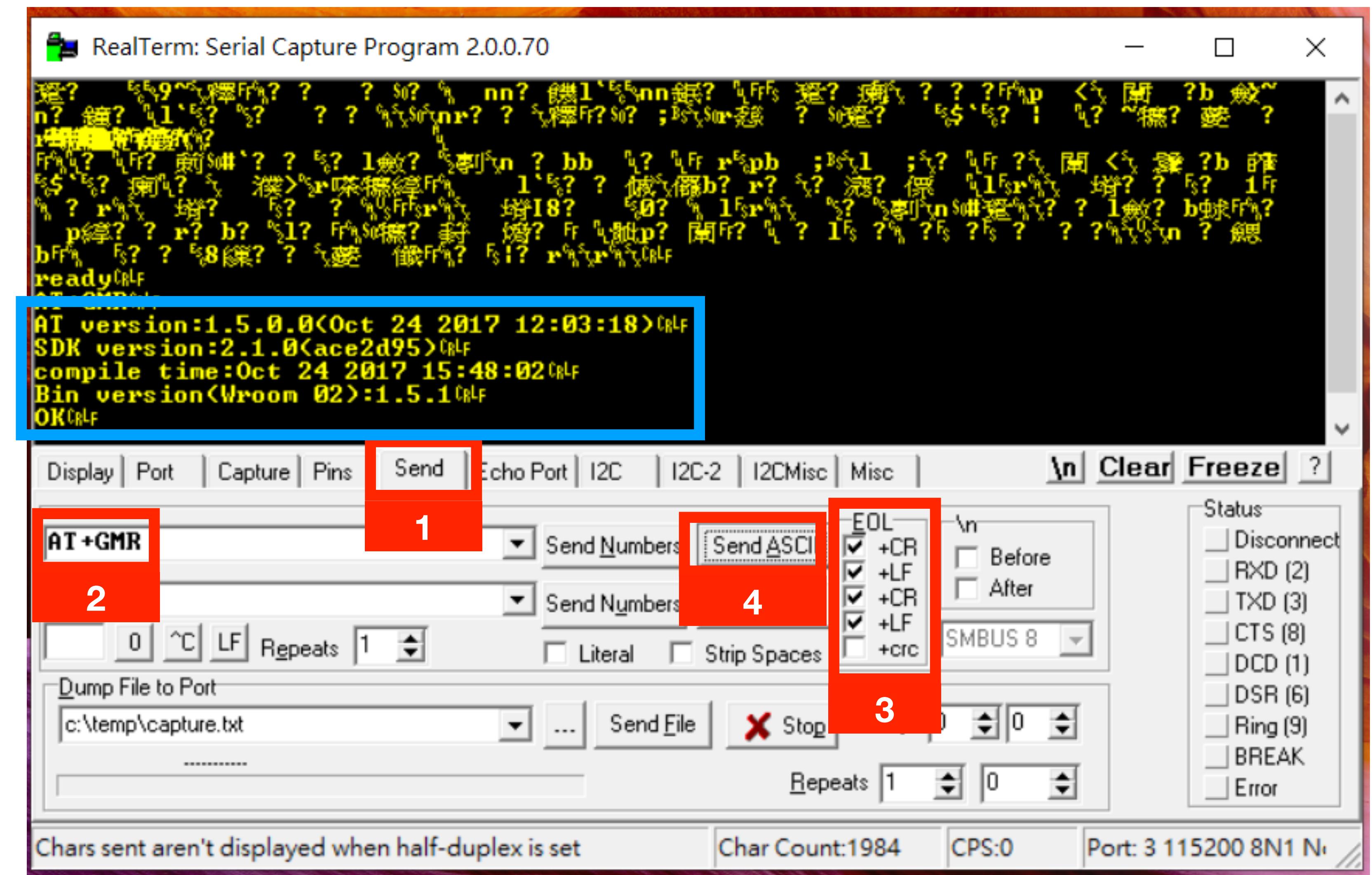
6. 應可見ready



# 使用Realterm

## 確認韌體版本

- 前往[Send]
- 輸入AT+GMR指令
- 勾選EOL設定
- 按[Send ASCII]
- 應可見藍框內的訊息



# 使用Realterm

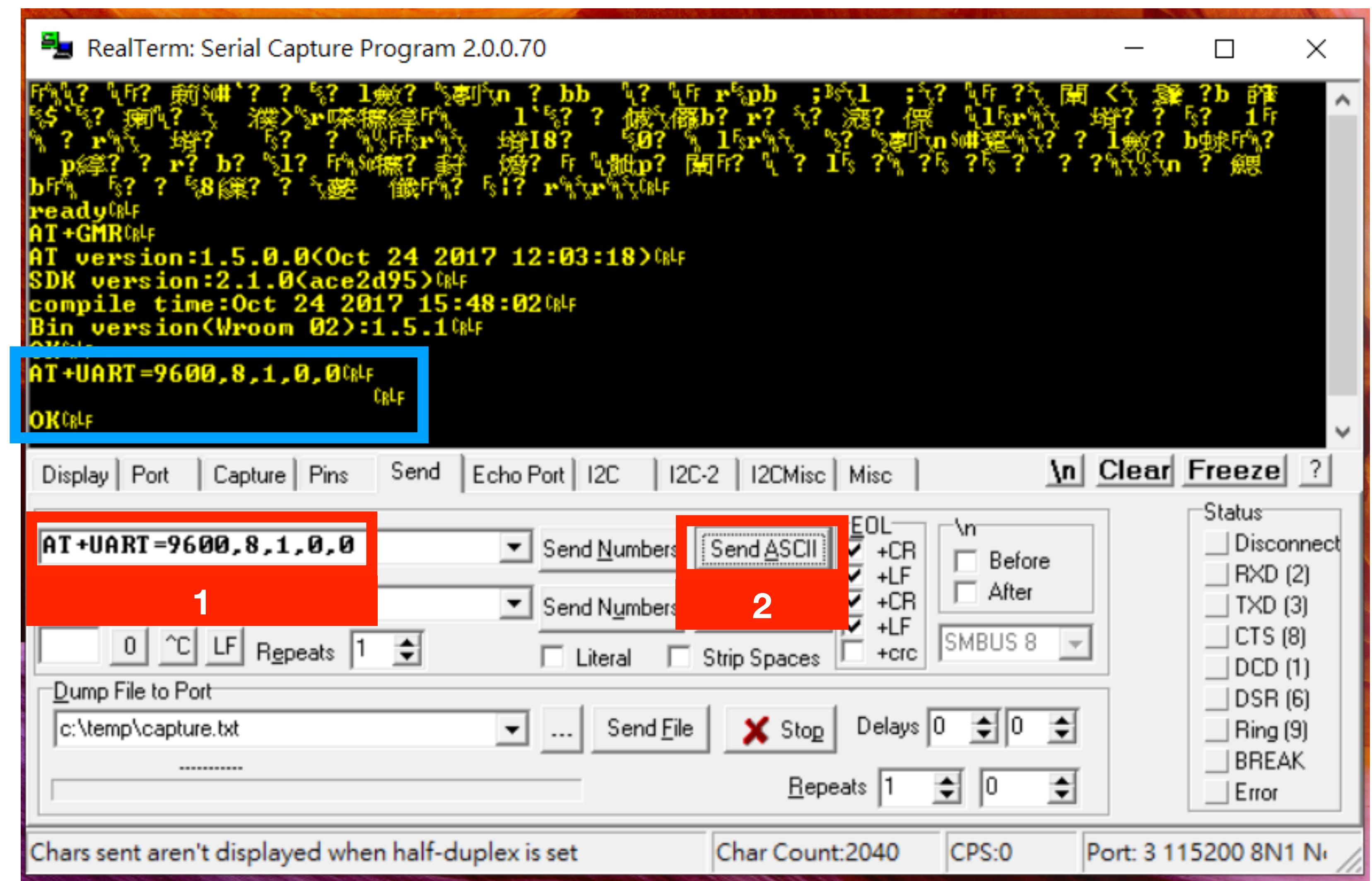
## 修改鮑率

1. 輸入

AT+UART=9600,8,1,0,0

2. 按[Send ASCII]

3. 應可見藍框內的訊息



# 使用Realterm

確認鮑率修改成功

1. 將Arduino斷電

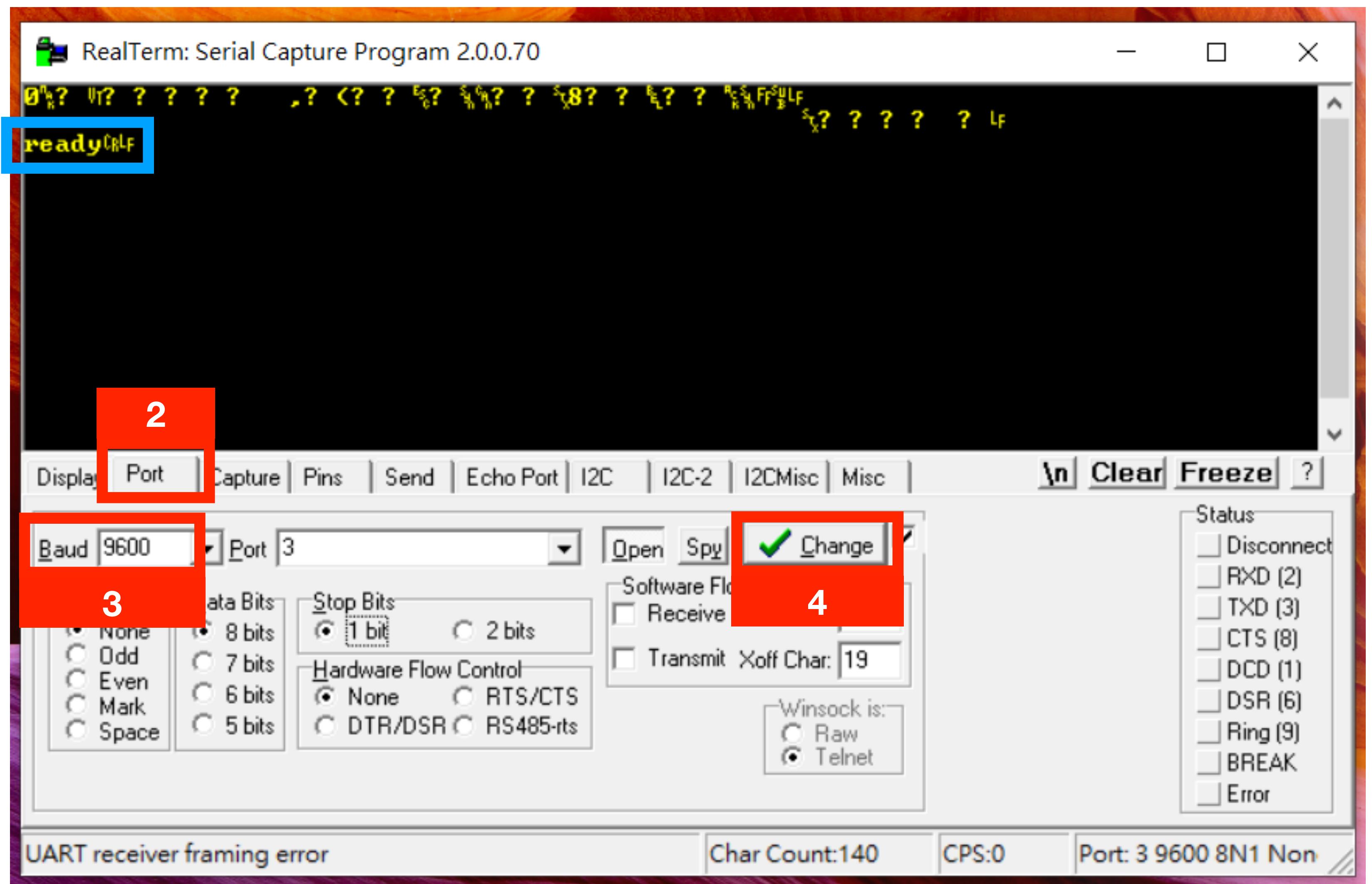
2. 前往[Port]

3. 更改鮑率至9600

4. 按[Change]

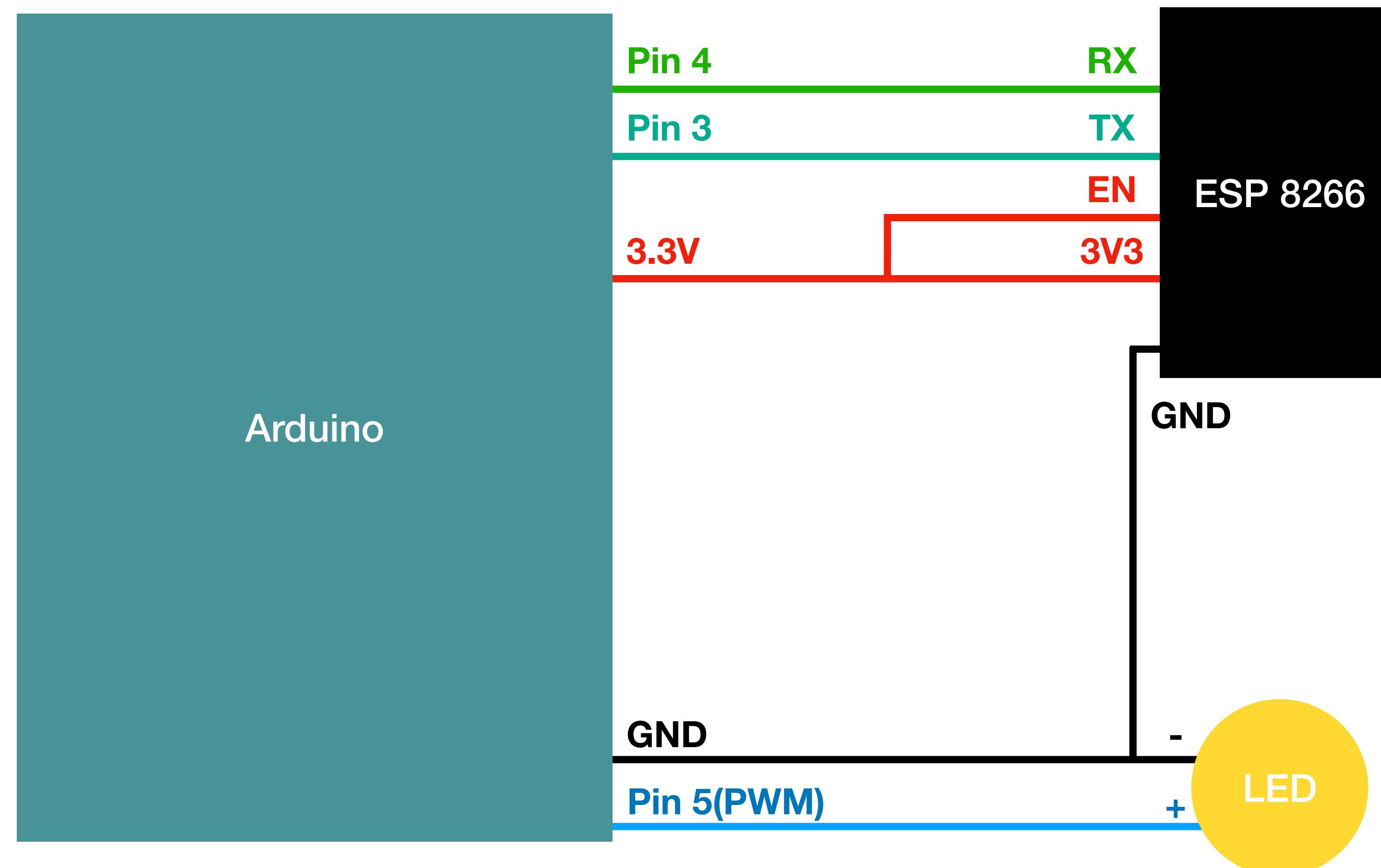
5. 將Arduino上電

6. 應可見ready



電路圖

# 電路圖





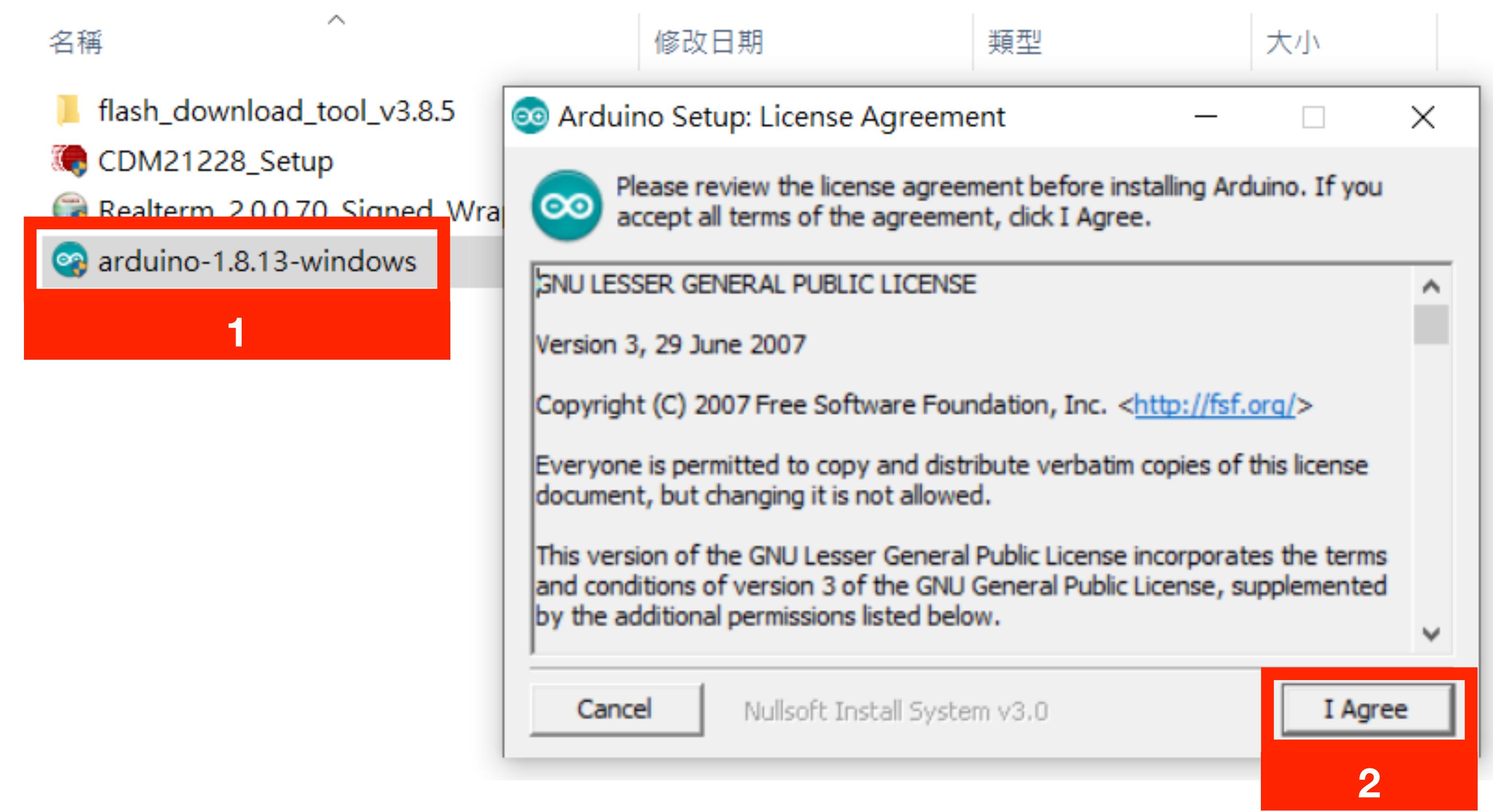
# 程式撰寫

# 安裝Arduino

## 使電腦可上傳程式至Arduino

1. 雙擊 [arduino-1.8.13-windows]

2. 按I Agree或Next等步驟  
直到安裝完成



# 安裝Arduino

## 使電腦可上傳程式至Arduino

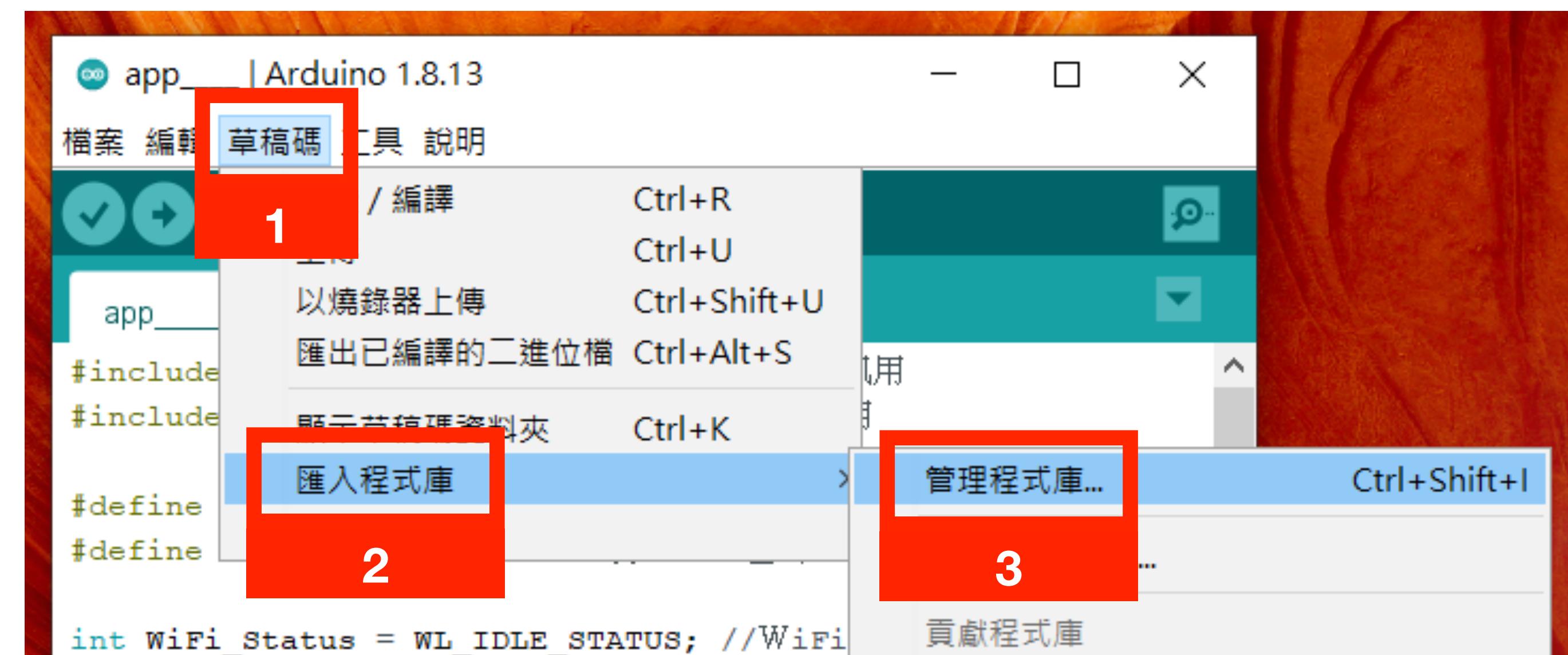
1. 若跳出安裝請求，請點擊安裝



# 安裝WifiESP Lib

## 使我們更方便操作通訊功能

1. 點擊[草稿碼]
2. 選擇匯入程式庫
3. 點擊[管理程式庫]



# 安裝WifiESP Lib

## 使我們更方便操作通訊功能

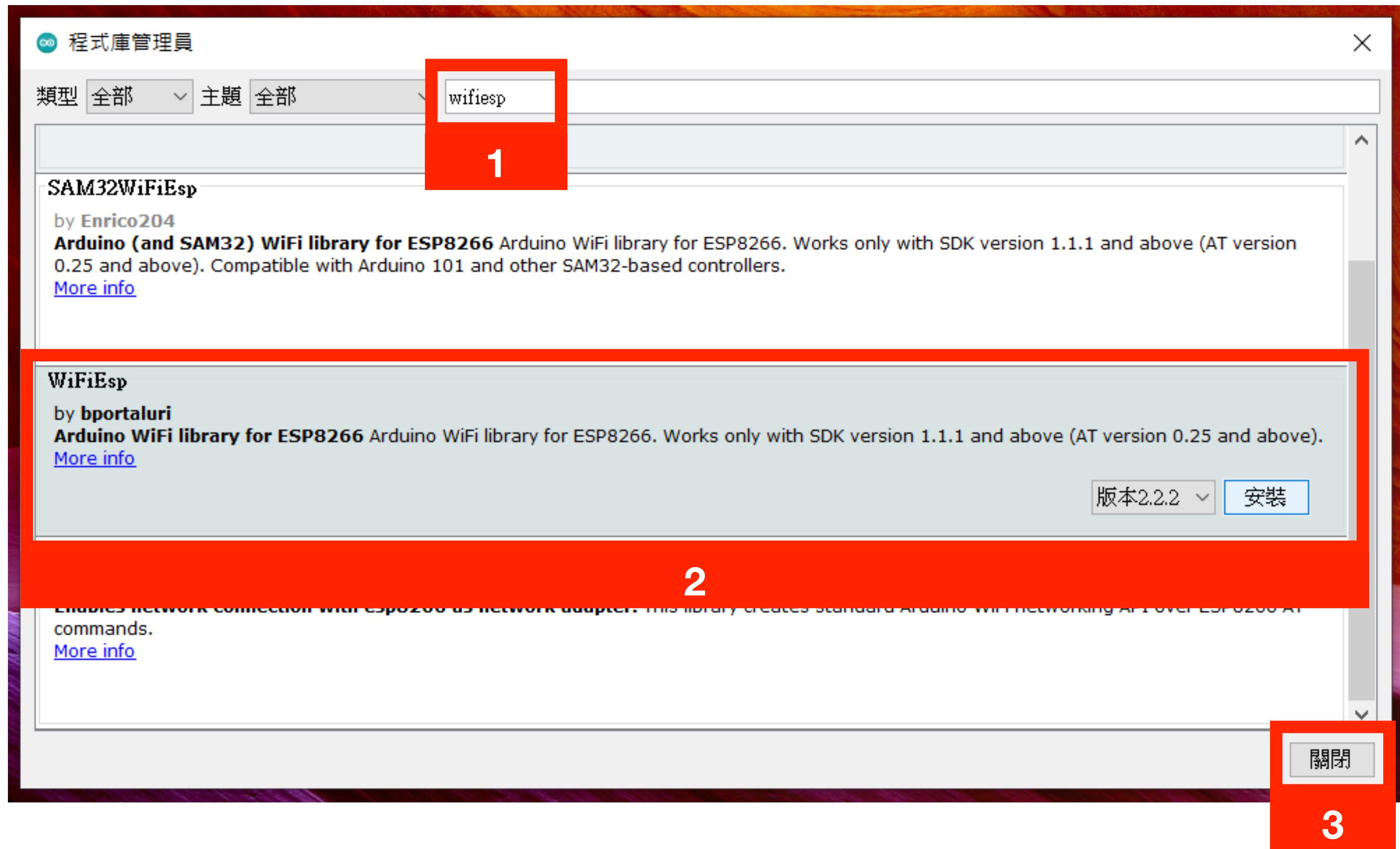
1. 至上方搜尋欄輸入

wifiesp並按enter

2. 選擇WifiEsp by

bportaluri並點擊[安裝]

3. 安裝完成後關閉視窗



# 匯入程式庫並序列埠物件

GLOBAL

```
#include <SoftwareSerial.h> //跟ESP8266通訊用
```

```
#include <WiFiEsp.h> //連接wifi跟網頁伺服器用
```

```
SoftwareSerial ESP8266(3, 4); //設定序列埠物件 (TX, RX)
```

# 設定Wifi連接資訊以及網頁伺服器

## GLOBAL

```
#define WIFI_SSID "" //填入WiFi帳號 (CCClass)
```

```
#define WIFI_PASSWORD "" //填入WiFi密碼 (CCClass123)
```

```
int WiFi_Status = WL_IDLE_STATUS; //WiFi狀態
```

```
WiFiEspClient espClient; //設定WiFiEspClient物件
```

```
WiFiEspServer server(80); //設定網頁伺服器物件
```

# 宣告相關變數

GLOBAL

```
String reqString = ""; //網頁請求資料
```

```
int brightness = 0; //呼吸燈亮度狀態
```

```
int fadeAmount = 5; //每次漸變的量
```

```
int delayDuration = 30; //脈衝間隔
```

```
int ledPin = 5; //LED腳位
```

# 初始化相關硬體

**setup()**

```
Serial.begin(9600); //設定序列埠傳輸速率(9600bps)
```

```
pinMode(ledPin, OUTPUT); // 設定燈光輸出
```

```
wifi_setting(); //wifi設定
```

```
server.begin(); //啟動網頁伺服器
```

# 撰寫wifi連接步驟 - P1

GLOBAL

```
void wifi_setting () {  
    //裡面放置等下要撰寫的程式碼  
}
```

# 撰寫wifi連接步驟 - P2

wifi\_setting()

```
ESP8266.begin(9600); //設定ESP8266傳輸速率(9600bps)
```

```
WiFi.init(&ESP8266); //初始化ESP模組
```

```
Serial.print("進行WiFi設定!\r\n");
```

# 撰寫wifi連接步驟 - P3

## wifi\_setting()

do{ //不管wifi狀態如何，都得執行一次wifi連接

    Serial.println("WiFi 連接中 ...");

    WiFi\_Status = WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //更新wifi連接狀態

    delay(500);

} while (WiFi\_Status != WL\_CONNECTED); //若wifi狀態屬沒連接，則重複迴圈

# 撰寫wifi連接步驟 - P4

wifi\_setting()

```
Serial.print("WiFi 連接成功!\r\n");
```

```
Serial.print("IP 位址: ");
```

```
Serial.println(WiFi.localIP()); //輸出Arduino所獲得的IP地址，務必記住！
```

```
Serial.println("WiFi 設定結束\r\n");
```

```
doBreathLED(); //通過呼吸燈通知wifi已連接成功
```

# 撰寫呼吸燈程式 - P1

GLOBAL

```
void doBreathLED () {
```

//裡面放置等下要撰寫的程式碼

```
}
```

# 撰寫呼吸燈程式 - P2

doBreathLED()

```
do {  
  
    analogWrite(ledPin, brightness); //發送脈衝  
  
    brightness = brightness + fadeAmount; //更新亮度狀態  
  
    if (brightness <= 0 || brightness >= 255) fadeAmount = -fadeAmount ; //更新正負  
  
    delay(delayDuration); //脈衝間隔  
  
} while (brightness != 0); //製作堵塞回圈，使呼吸燈完成一個輪迴
```

# 撰寫呼吸燈程式 - P3

doBreathLED()

```
digitalWrite(ledPin, LOW); //使LED完全熄滅
```

# Main

loop()

Too complex for beginners

```
void loop() {
    WiFiEspClient client = server.available(); //建立客戶端與伺服端連線

    if (client) { //若連線有效
        while (client.connected()) { //當客戶端與伺服端連線
            if (client.available()) { //若客戶端的請求是有用的
                char c = client.read();
                reqString += c; //把請求資料合併

                if (c == '\n') { //當請求資料收完
                    if (reqString.indexOf("?flash") > 0) { //檢查有沒有 ?flash
                        for (int i = 0; i < 3; i++) {
                            doBreathLED();
                        }
                        Serial.println("Led Flash");
                    }

                    client.print( //伺服端回覆客戶端停止連線
                    "HTTP/1.1 200 OK\r\n"
                    "Connection: close\r\n"
                    "\r\n");
                    break ;
                }

                //伺服端傳送HTML網頁給客戶端
                client.print(
                    "HTTP/1.1 200 OK\r\n"
                    "Content-Type: text/html\r\n"
                    "Connection: close\r\n" // the connection will be closed after completion of the request
                    "\r\n");
                client.print("<!DOCTYPE HTML>\r\n");
                client.print("<meta charset=UTF-8><title>Arduino Web Server</title><script>function");
                break ;
            }
        }

        delay(10); //預留傳送時間
        client.stop(); //停止這次連線
        reqString = ""; //清空請求資料
    }
}
```

# 構建HTML網頁(HyperText Markup Language)

```
<html>
|   <head></head>
|   <body></body>
</html>
```

# 構建HTML網頁

```
<html>
  <head></head>
  <body>
    <div align="center">
      <h1>呼叫機器</h1>
      <button type="button" onclick="flash2Arduino()">閃爍</button>
    </div>
  </body>
</html>
```

# 構建HTML網頁

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Arduino Web Server</title>
    <script type="text/javascript"></script>
  </head>
  <body>
    <div align="center">
```

# 構建HTML網頁

```
<script type="text/javascript">
| function flash2Arduino(){
|   var xmlhttp;
|
|   if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
|     xmlhttp = new XMLHttpRequest();
|   } else { // code for IE6, IE5
|     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
|   }
|
|   xmlhttp.open("GET","?flash", true);
|   xmlhttp.send();
|
| }
</script>
```

# 最小化HTML

1. 前往 <https://smallseotools.com/minify-html/>
2. 把HTML貼上輸入框
3. 按[Minify HTML]
4. 便取得最小化格式

The screenshot shows a web page with a red border around the input field. Inside the input field, there is some sample HTML code. Below the input field is a large orange button with the number '2' on it. At the bottom of the input field, there is a link 'Show Advanced Options' with a red plus sign icon.

```
<html>
<head>
<meta charset="UTF-8">
<title>Arduino Web Server</title>
<script type="text/javascript">
function flash2Arduino(){
var xmlhttp;
```

The screenshot shows a web page with a red border around the 'Minify HTML' button. The button is blue with white text. Below the button is the number '3'. At the top of the page, it says 'TextOptimizer.com Test your Website' with a help icon and a close icon. On the right side, there is a dark grey button with the word 'OPEN' in white.

TextOptimizer.com  
Test your Website  
OPEN

Minify HTML

3

# 最小化HTML

1. 滑至網頁下端
2. 點擊[Copy To Clipboard]
3. 複製貼上至arduino程式碼裡

**RESULTS**

Input: 816 KB      Output: 321 KB      You Save: 495 KB (60.66%)

**Output:**

```
<meta charset=UTF-8><title>Arduino Web Server</title><script>function flash2Arduino()
{var e;(e=window.XMLHttpRequest)?new XMLHttpRequest:new
ActiveXObject("Microsoft.XMLHTTP")).open("GET","?flash",!0),e.send()}</script><div
align=center><h1>呼叫機器</h1><button onclick=flash2Arduino() type=button>閃爍
</button></div><div></div>
```

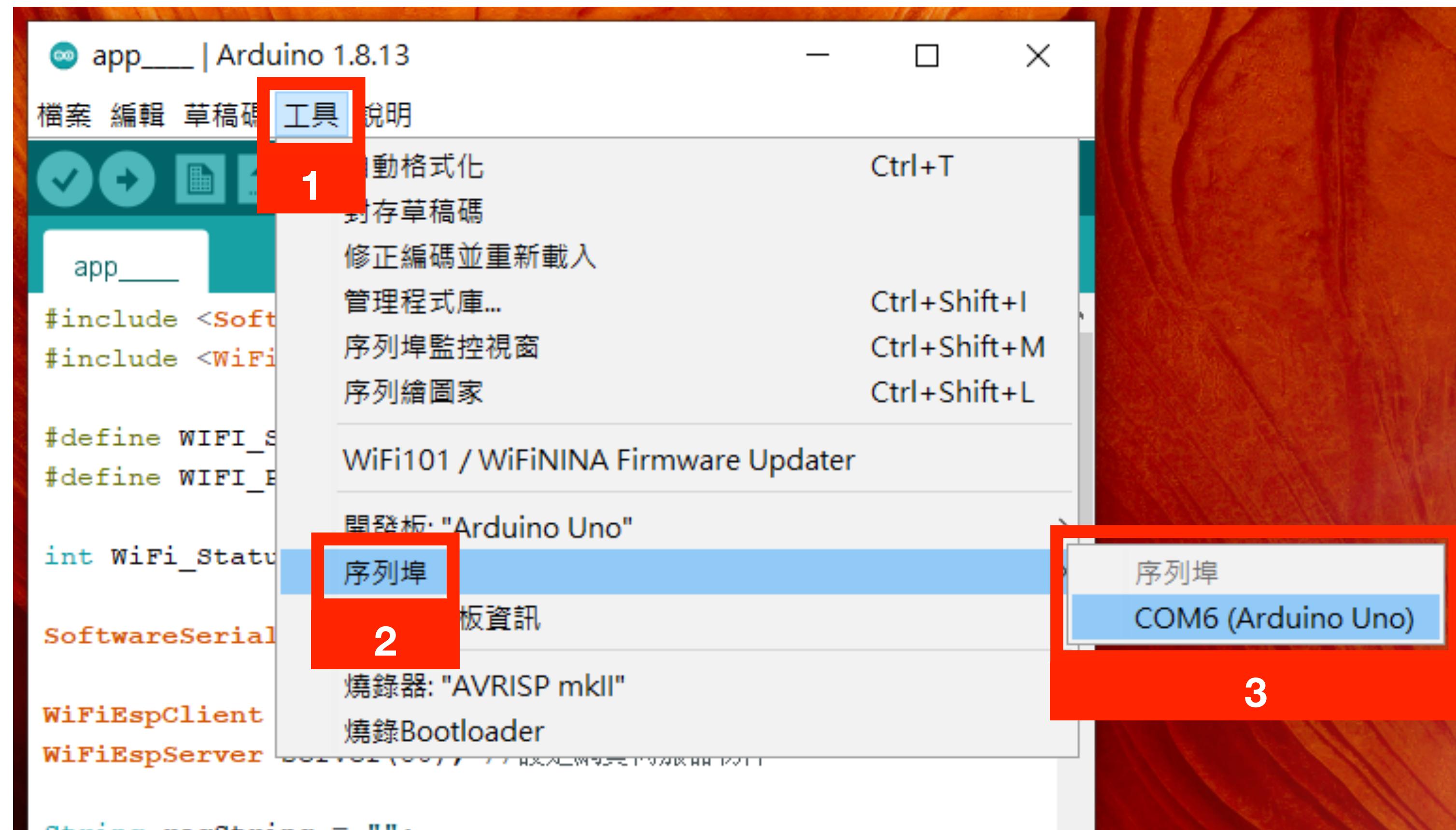
**Copy To Clipboard**

2

# 上傳程式

## 選擇通訊接口

1. 點擊[工具]
2. 選擇序列埠
3. 點擊連接著Arduino的  
COM口



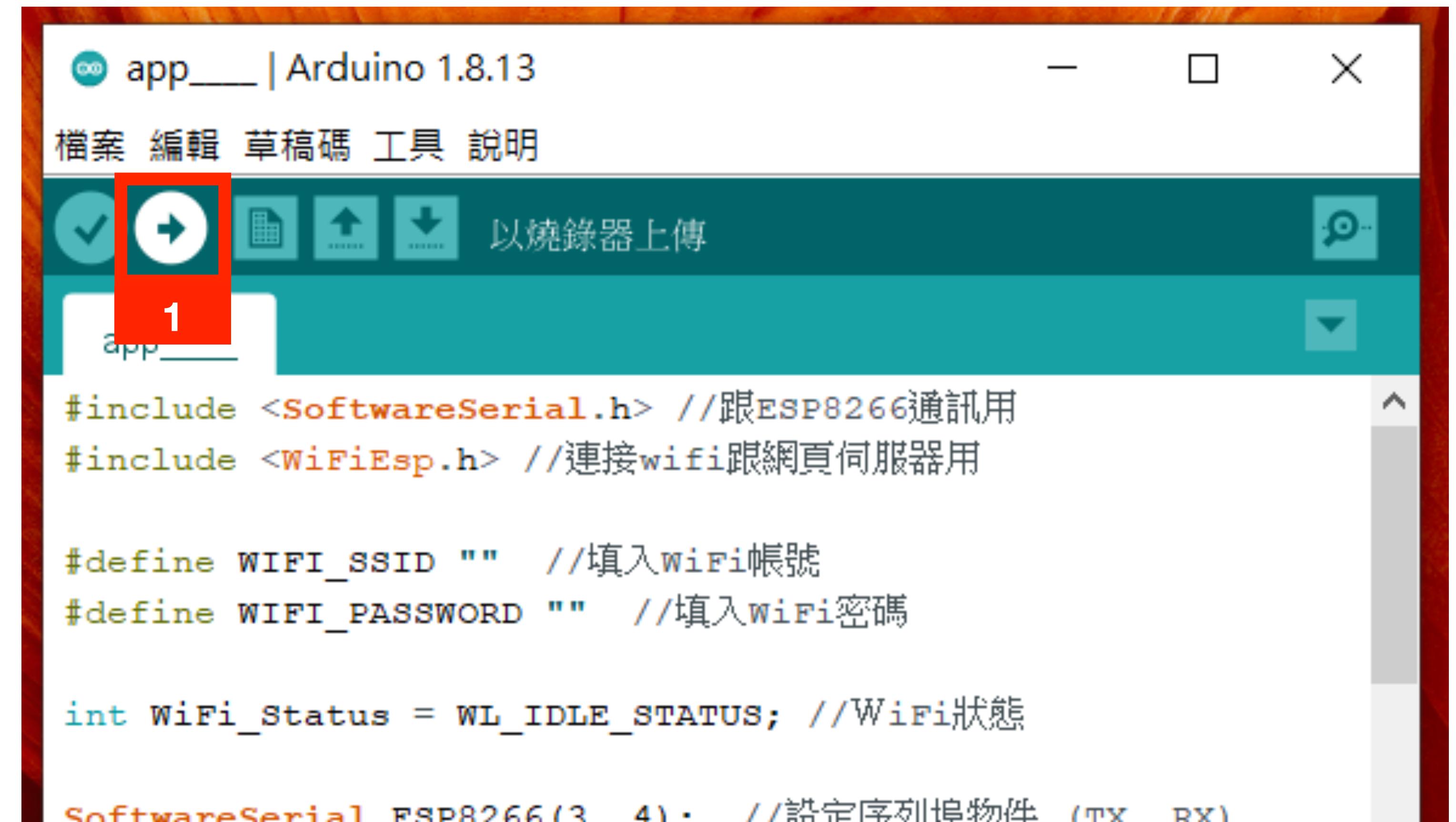
# 上傳程式

## 編譯並上傳

1. 點擊第二個符號(以燒錄器上傳)

2. 等待完成編譯，程式便會開始上傳至Arduino

\* 若底下出現紅字，請尋求協助。



The screenshot shows the Arduino IDE interface with the title bar "app\_ | Arduino 1.8.13". The menu bar includes "檔案" (File), "編輯" (Edit), "草稿碼" (Sketch), "工具" (Tools), and "說明" (Help). The toolbar features icons for checkmark, upload (highlighted with a red box and labeled "1"), and other functions. A status message "以燒錄器上傳" (Upload via programmer) is displayed. The code editor contains the following C++ code:

```
#include <SoftwareSerial.h> //跟ESP8266通訊用
#include <WiFiEsp.h> //連接wifi跟網頁伺服器用

#define WIFI_SSID "" //填入wifi帳號
#define WIFI_PASSWORD "" //填入wifi密碼

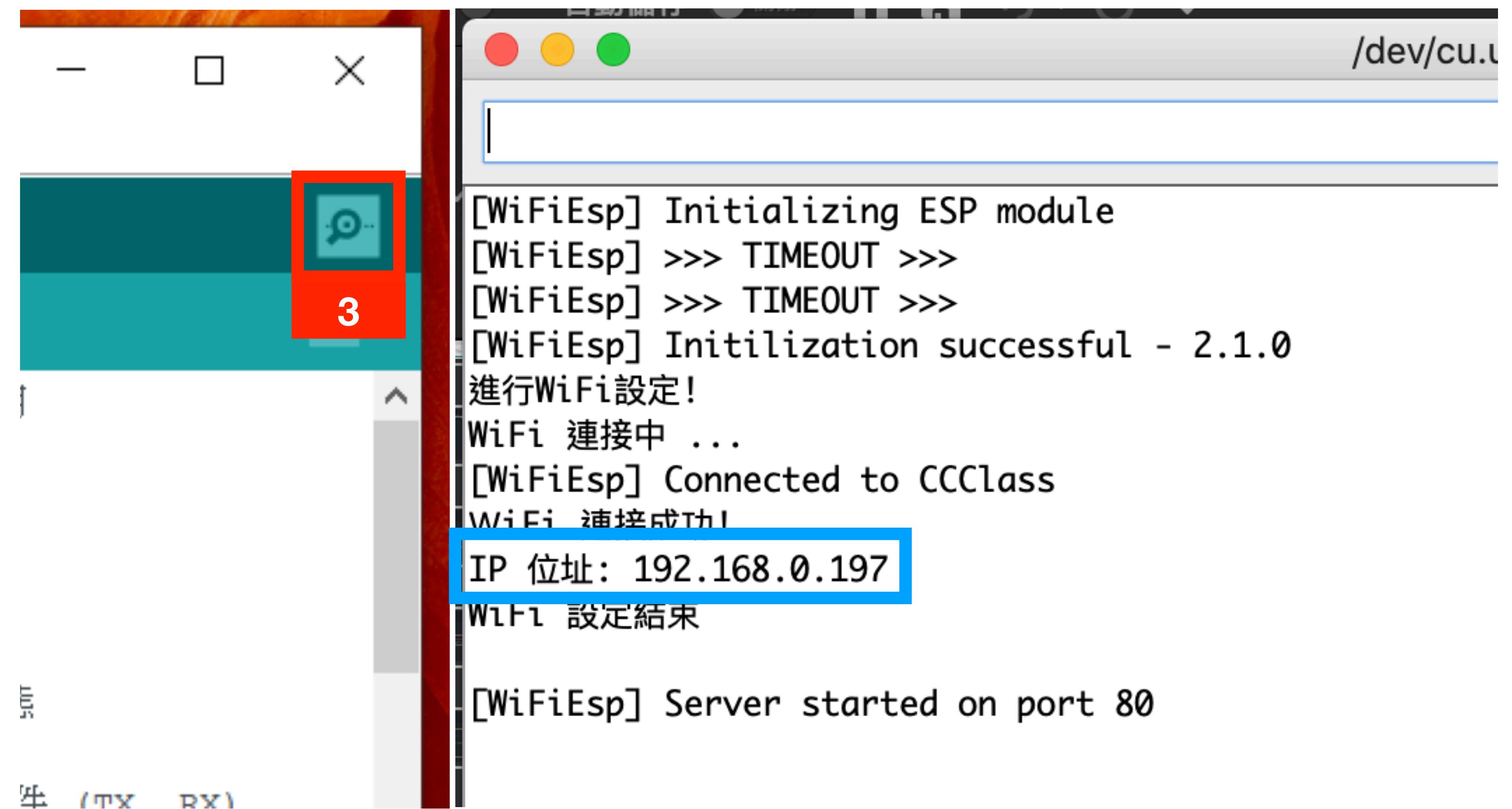
int WiFi_Status = WL_IDLE_STATUS; //WiFi狀態

SoftwareSerial ESP8266(3, 4); //設定序列埠物件 (TX RX)
```

# 查看IP地址

## 尋找網頁網址

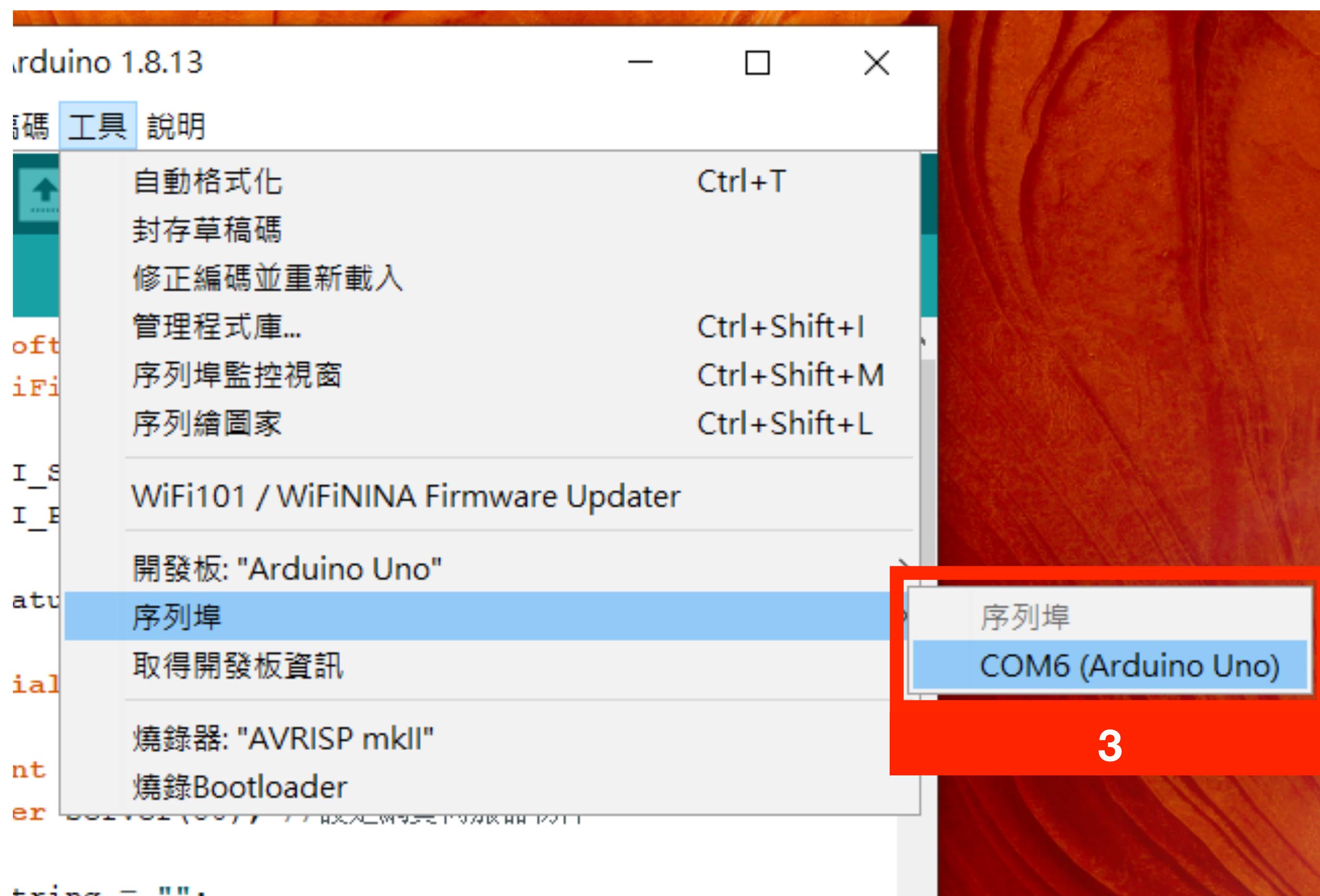
1. 確認上傳完畢後，將 Arduino斷電
2. 將Arduino上電
3. 打開序列埠監聽窗口
4. 監聽窗口應顯示連接資訊並出現IP地址，如藍框



# 查看IP地址(常見問題)

若監聽窗口未出現資訊

1. 確認上傳完畢後，將 Arduino斷電
2. 將Arduino上電
3. 立刻前往序列埠選擇表單
4. 打開序列埠監聽窗口
5. 監聽窗口應顯示連接資訊



# 操作Arduino

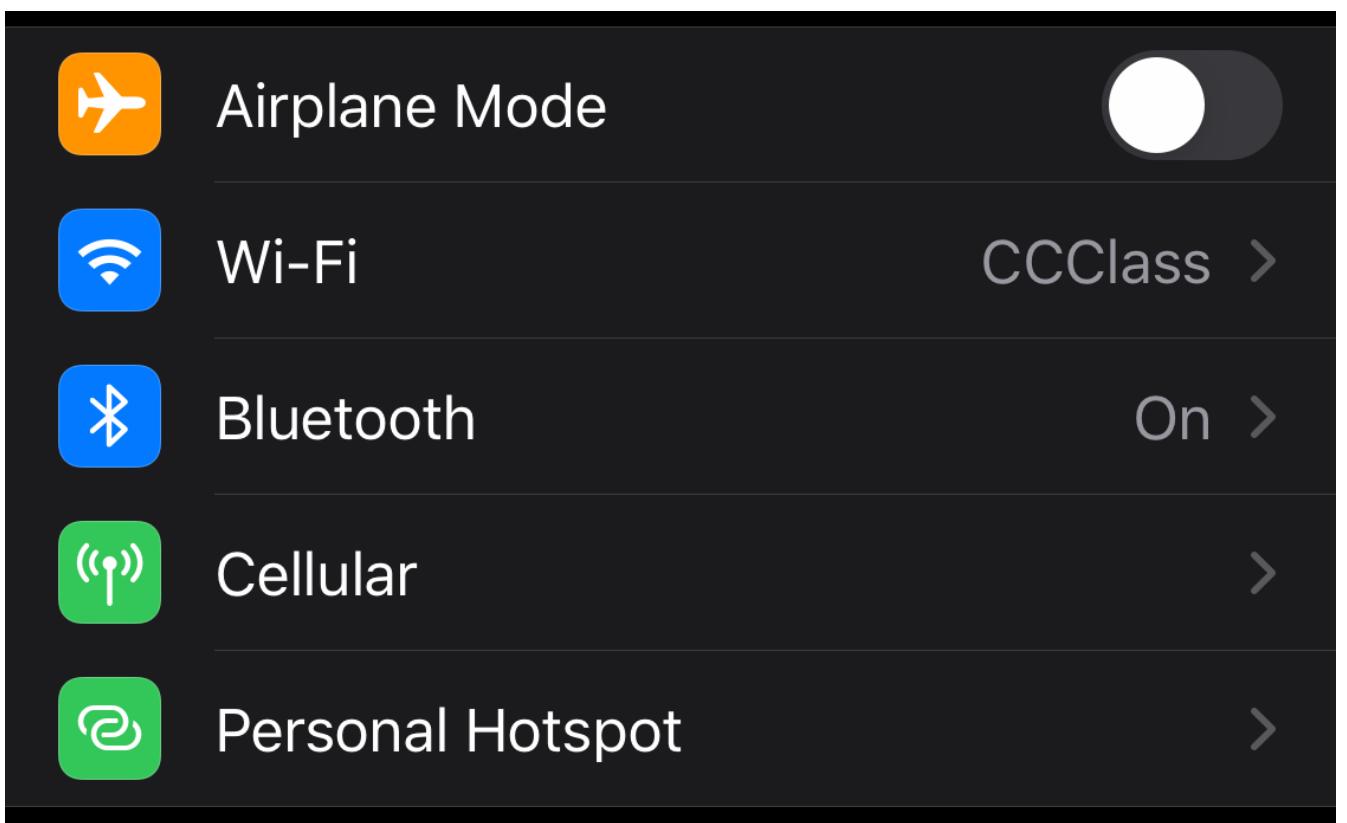
前往指定網址(iOS為例，但概念一樣喔)

WiFi帳號：CCClass

WiFi密碼：CCClass123

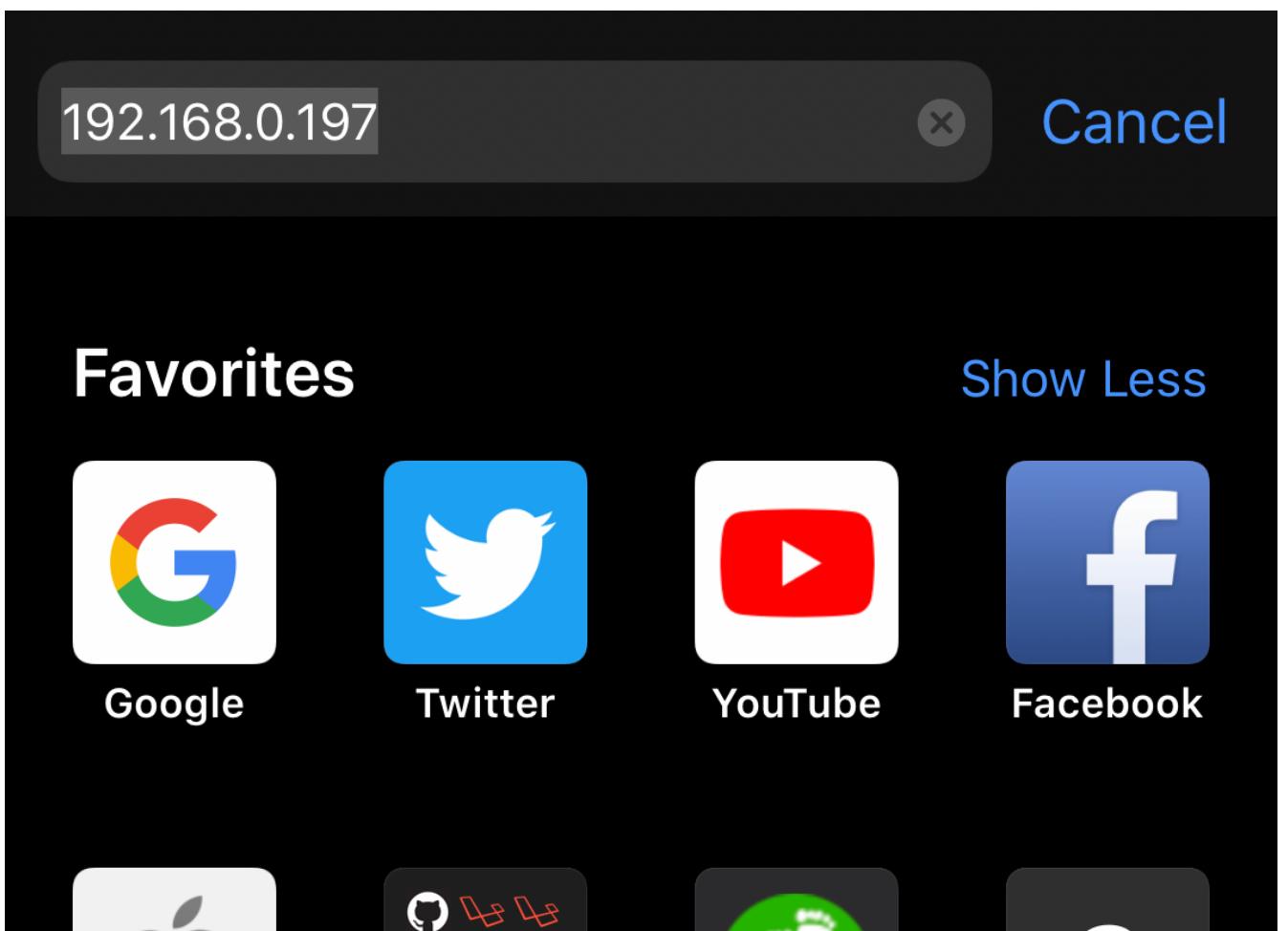
1. 確認手機已經與Arduino連接

在相同的網路裡 (CCClass)



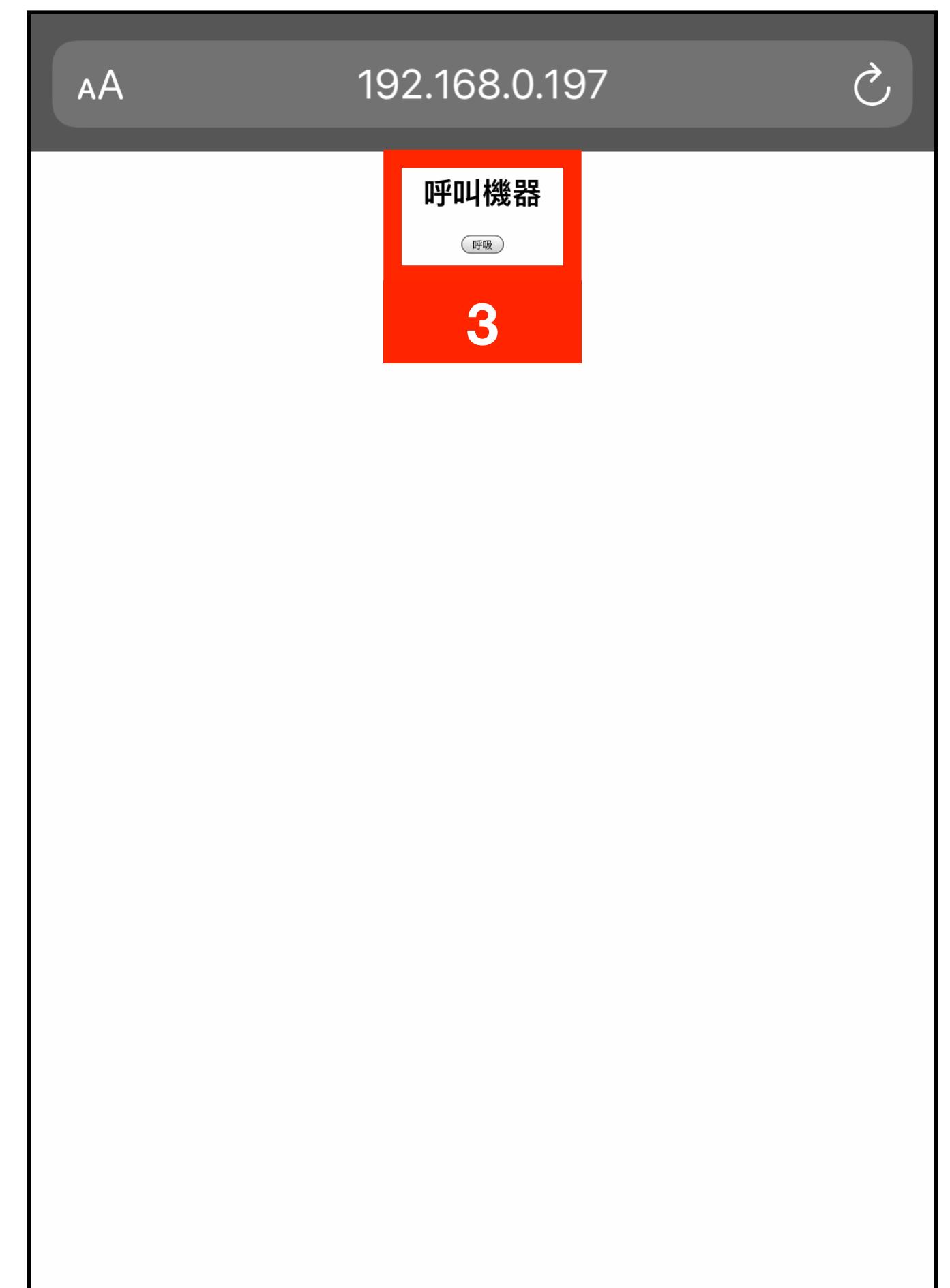
2. 前往瀏覽器並網址欄輸入

Arduino所給的IP地址



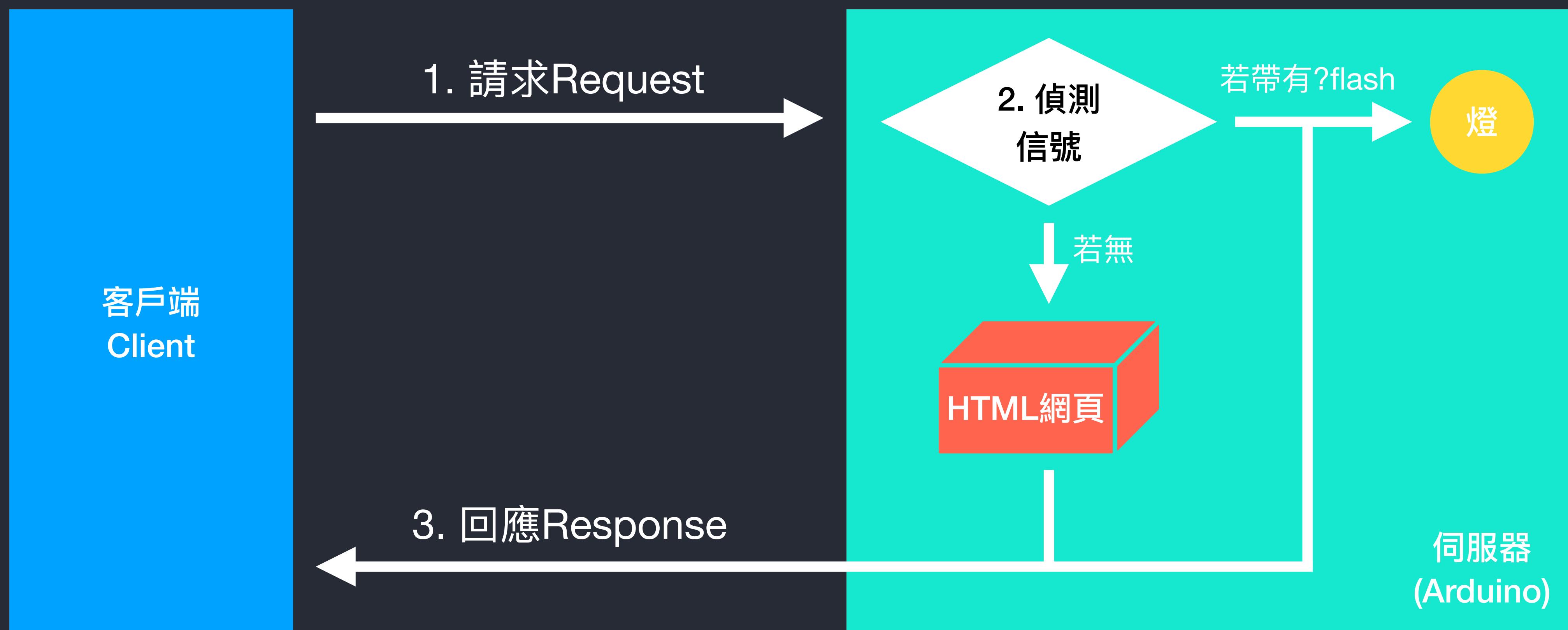
3. 點擊中間的[呼吸]按鈕，

Arduino便會執行呼吸燈3次



# 原理角解析

# 網頁解析 - 訊號流程圖



# 網頁解析 - 分工合作

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Arduino Web Server</title>
    <script type="text/javascript">
      function flash2Arduino(){
        var xmlhttp;

        if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
          xmlhttp = new XMLHttpRequest();
        } else { // code for IE6, IE5
          xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }

        xmlhttp.open("GET","?flash", true);
        xmlhttp.send();
      }
    </script>
  </head>

  <body>
    <div align="center">
      <h1>呼叫機器</h1>
      <button type="button" onclick="flash2Arduino()">閃爍</button>
    </div>
  </body>
</html>
```

發送http請求給  
Arduino

網頁介面宣告

# 網頁解析 - 遠端遙控呼吸燈的秘密

## 1. 使用Ajax技術裡的XmlHttp

技術來傳送訊息給Arduino

## 2. IE5 & IE6使用Active X，而

其他瀏覽器使用

XMLHttpRequest來操作。

```
function flash2Arduino(){
    var xmlhttp;

    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    } else { // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.open("GET","?flash", true);
    xmlhttp.send();
}
```

# 網頁解析 - 遠端遙控呼吸燈的秘密

## 1. window.XMLHttpRequest

可以判斷出當前使用的瀏

覽器是否為IE5或IE6

## 2. Open函數告訴要打開什麼

連結(?flash)

## 3. Send函數確認發送資料

```
function flash2Arduino(){
    var xmlhttp;

    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    } else { // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.open("GET","?flash", true);
    xmlhttp.send();
}
```



代表異步，不堵塞整個程式

# 相關應用 - IoT物聯網

Google Home, Alexa  
Homekit...

TaiSEIA 101...

小米手環...

公車追蹤...

居家自動化

智慧能源

健康照護

智慧交通

智慧生活

促進工業4.0的發展

