

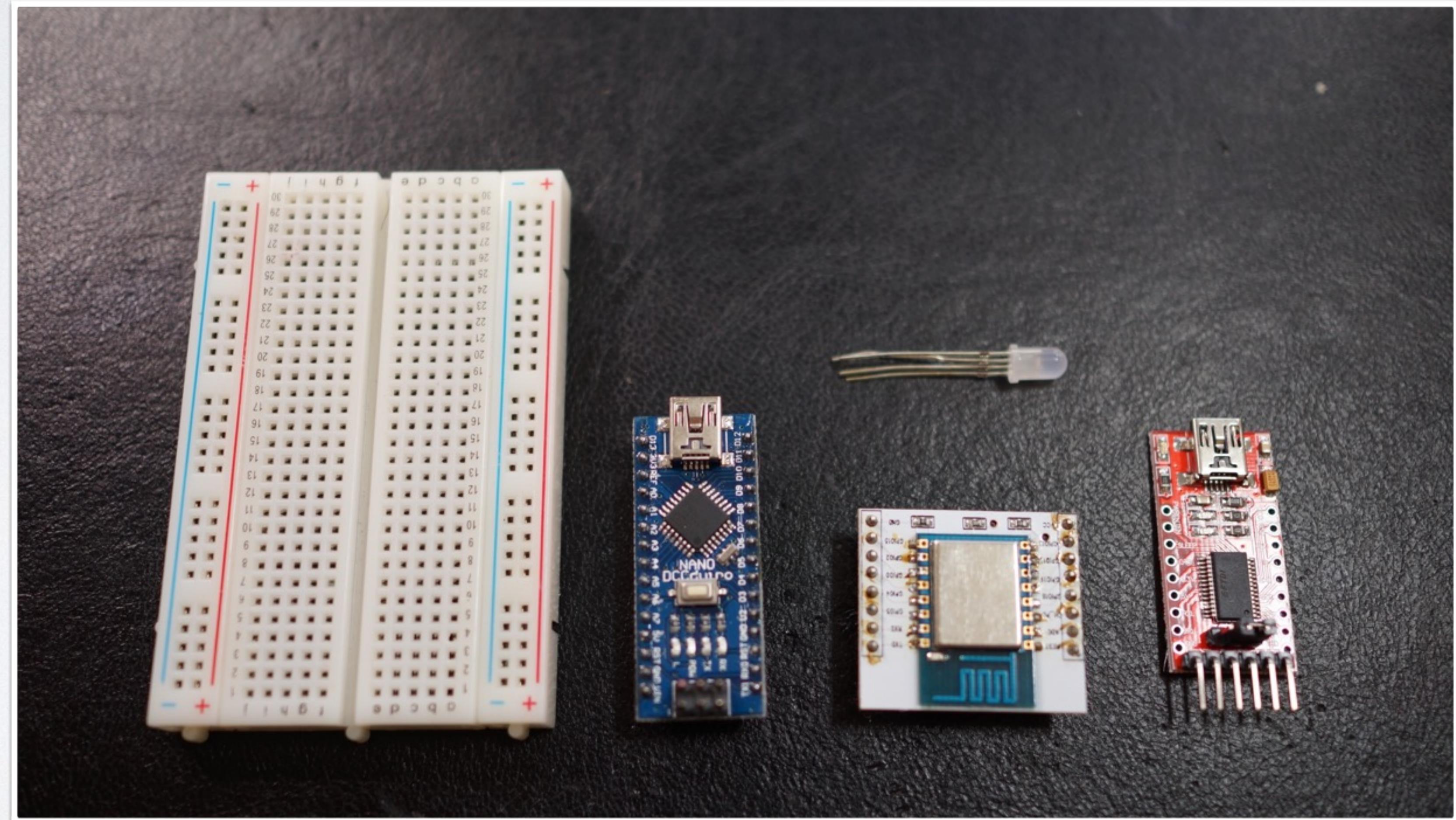
UNU

IoT

ESP8266

# 材料

- Adruino Nano × 1
- ESP8266 × 1
- FTDI232 TTL × 1
- RGB LED × 1
- 杜邦線數根



# 開發套件和平台

ESP8266 介紹



# ESP8266 介紹

- 規格 – ESP-12
  - 802.11 b/g/n
  - WiFi Direct (P2P) 、 soft-AP
  - 整合 TCP/IP 協議
  - 整合 TR switch 、 balun 、 LNA 、 PA 和 matching network
  - 整合 PLLs 、 穩壓器 、 DCXO 和電源管理單元
  - 802.11b 模式下 +19.5dBm 的輸出功率
  - 小於 10uA 的漏電流
  - 整合低功耗 32 位元 CPU ，可以兼作應用處理器
  - SDIO 1.1/2.0 、 SPI 、 UART
  - STBC 、 1×1 MIMO 、 2×1 MIMO
  - A-MPDU & A-MSDU 聚合 & 0.4ms 的保護間隔
  - 2ms 之內喚醒並開始傳送資料
  - 待機狀態消耗功率少於 1.0mW (DTIM3)



# ESP8266 功能

- 序列傳輸 : 最大速率 460800 bps
- PWM : 燈光調節 RGB LED, 馬達調速等
- GPIO : 控制開關 繼電器等

# ESP8266 工作模式

- SAT : 透過路由器連接網路
- AP : 做為 AP 提供設備連結
- SAT+AP : 混合模式

# ESP8266

## AT

- [https://github.com/espressif/esp8266\\_at/wiki](https://github.com/espressif/esp8266_at/wiki)

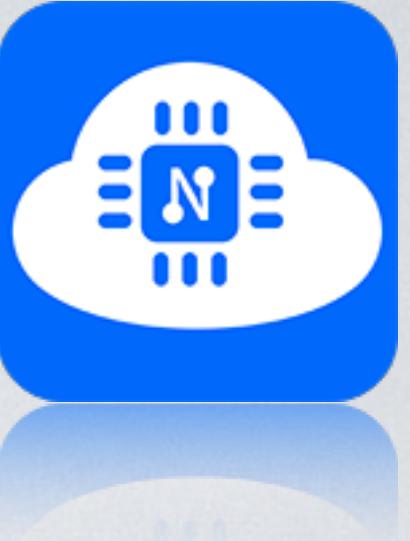
The screenshot shows the GitHub wiki page for the esp8266\_at repository. The top navigation bar includes the repository name "espressif / esp8266\_at", a "Watch" button (46), a "Star" button (83), a "Fork" button (42), and "Edit" and "New Page" buttons. To the right is a sidebar with icons for file operations like copy, paste, and search. A sidebar menu titled "Pages 50" lists several AT command categories:

- Home
- Compiler
- AT Command set
  - Basic AT
    - > AT
    - > AT+RST
    - > AT+GMR
    - > AT+GSLP
    - > ATE
  - WIFI Function AT
    - > AT+CWMODE
    - > AT+CWJAP
    - > AT+CWLAP
    - > AT+CWQAP
    - > AT+CWSAP

The main content area features a "Home" section with a note from AthenaYu about edits made on Dec 8, 2014. It also includes links to the Espressif BBS and a "Note" section containing instructions for code submission and folder structure details.

# ESP8266

## NodeMCU

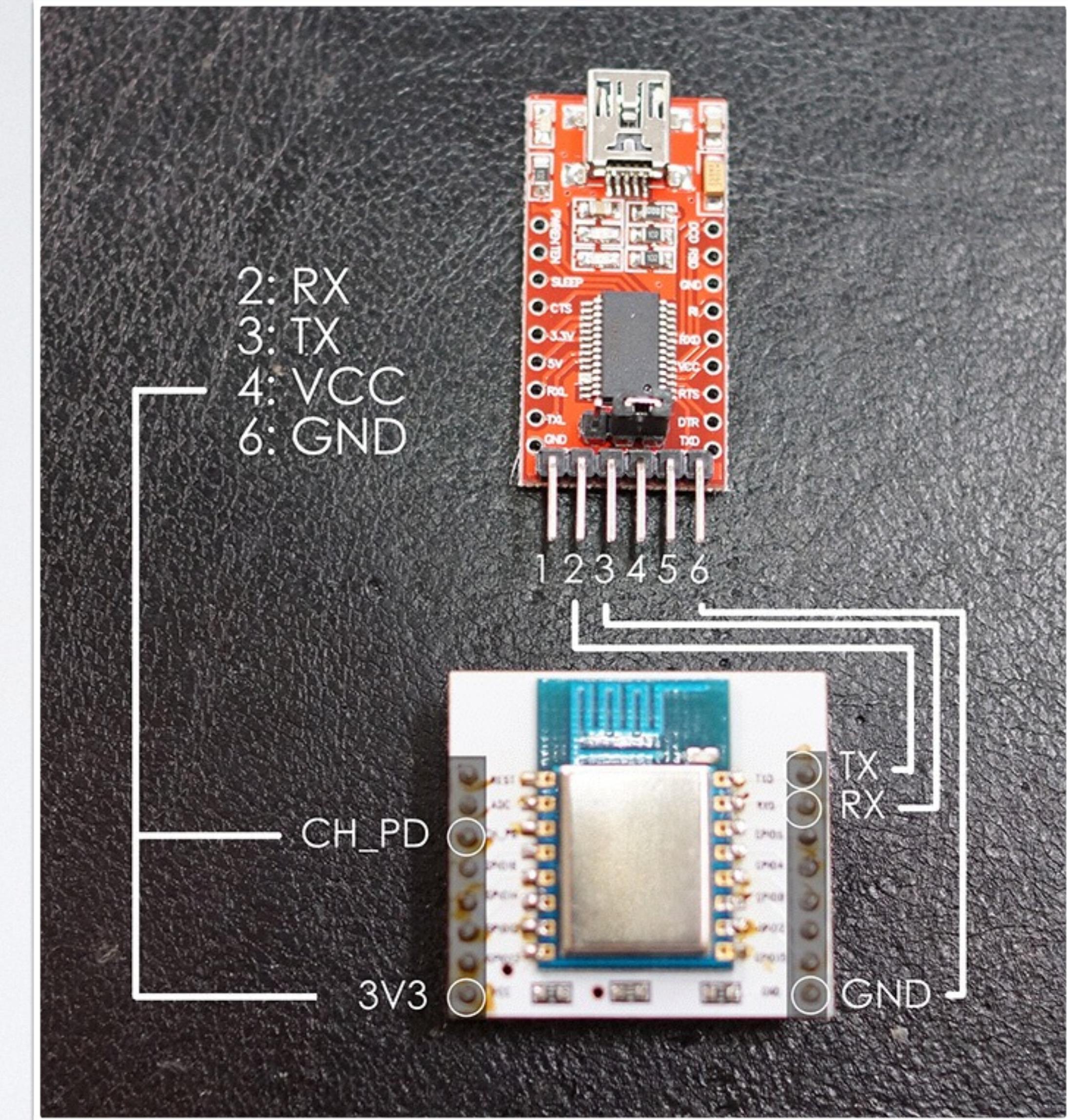


- [http://www.nodemcu.com/index\\_cn.html](http://www.nodemcu.com/index_cn.html)



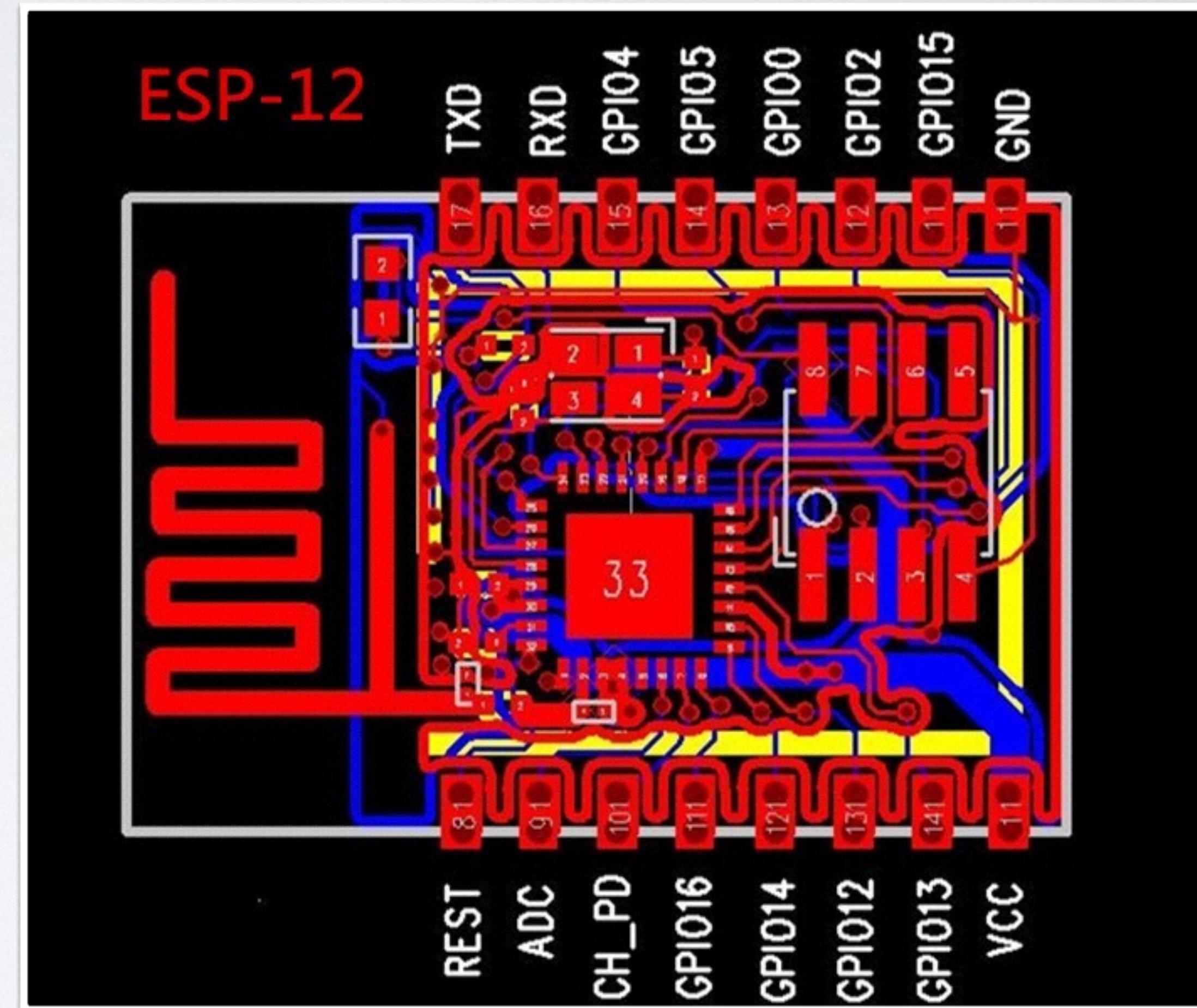
# LAB 000

## 連接線路



# 接腳說明

PIN	Function	Description
1	URXD	1.UART_RXD，接收； 2.General Purpose Input/Output : GPIO3；
2	UTXD	1.UART_TXD，發送； 2) General Purpose Input/Output : GPIO1；
5	RESET (GPIO 16)	外部Reset，低電位重開，高電位工作 (default)；
6	GND	GND
8	VCC	3.3V
9	ANT	WiFi Antenna
11	GPIO0	預設 WiFi Status : WiFi 工作指示燈 1.工作模式选择： 空接：Flash Boot，工作模式； 接地：UART Download，下載模式；
12	ADC	ADC，输入：0V-1V；
13	GPIO15	
14	CH_PD	1.高电位工作； 2.低电位關閉供电；
15	GPIO2	空接：工作模式；預設高電位



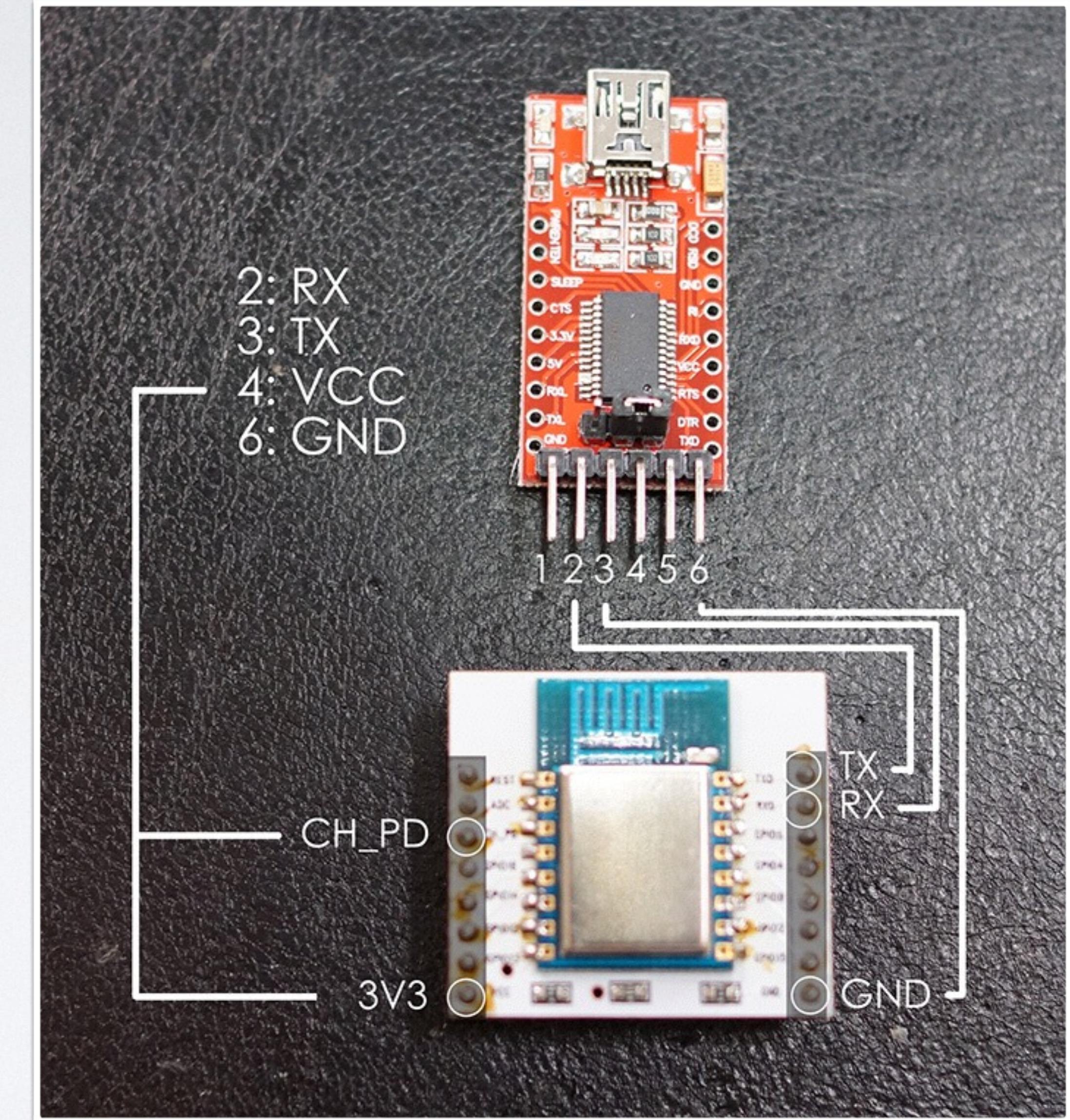
# 燒寫韌體

- 切斷ESP8266電源，將 GPIO0 接地重開進入燒寫模式

# LAB 101

## FIRMWARE

Espressif Systems



# 下載

- [https://github.com/espressif/esp8266\\_at](https://github.com/espressif/esp8266_at)

- AT 版本 0.20

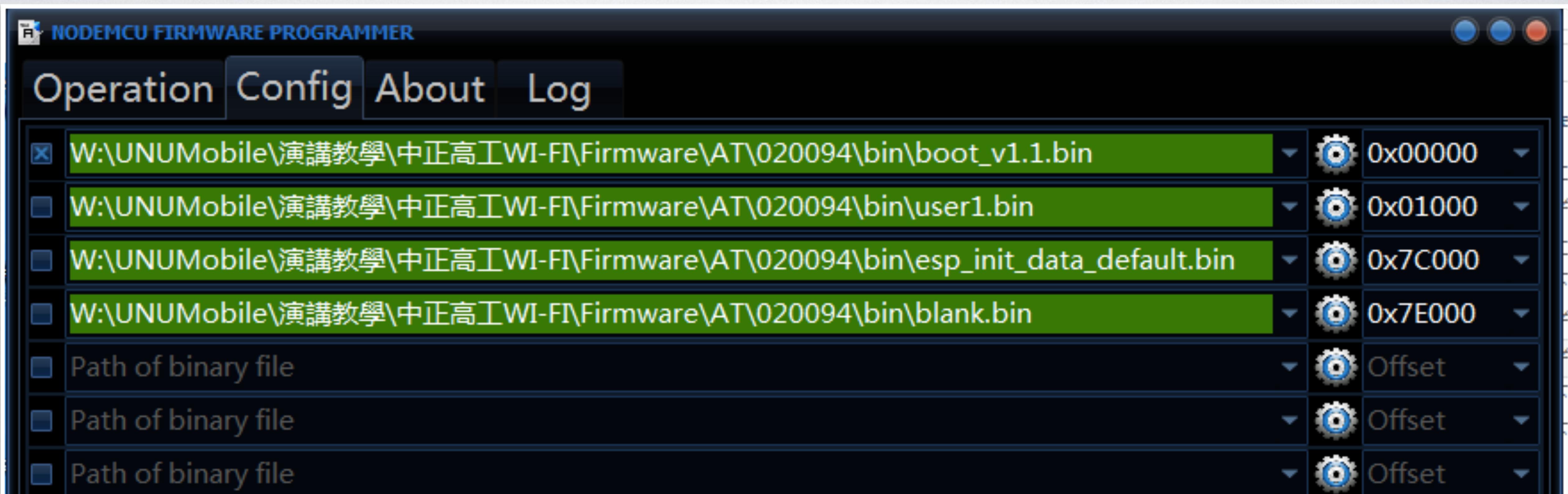
- SDK 0.9.4

The screenshot shows the GitHub repository page for `esp8266_at`. The repository is owned by `espressif`. Key statistics displayed are 17 commits, 2 branches, 1 release, and 2 contributors. The master branch is selected. The repository description is "ESP8266 Official AT+ Command <http://bbs.espressif.com>". The commit history shows two recent updates:

- update newest bin by `espressif` on 25 Dec 2014 (latest commit)
- update for sdk v0.9.4. by `at` on 4 months ago
- update newest bin by `bin` on 4 months ago

# 燒寫位址

- 路徑 : Firmware\AT\020094



# 連結

- 工具

## ESP8266 Config

The screenshot shows two windows related to ESP8266 configuration:

**AppStack ESP8266 Config V. 1.0.0.0** (Left Window):

- ESP8266 Config** tab is selected.
- Mode** section: AP (radio button), Set Mode, STA, Get Mode, FW Version: 0020000903, Reset, Enable WD, Disable WD, Set Baudrate: 9600, Get Baudrate.
- Mux** section: Single (radio button), Set Mux, Multiple, Get Mux.
- AP** section: SSID, Set AP, Get AP, Password, Channel: 1, Encryption: Open, Get join device.
- STA** section: SSID, List AP, Get AP, Password, Join AP, Quit AP, IP: -, Get IP.
- Client (TCP, UDP)** section: IP, Port, Listen, Close, Timeout, Set Timeout, Get Timeout, ID: 1, Protocol: TCP.
- Serial Monitor** section: Port: COM6, Baudrate: 115200, Disconnect, Serial Monitor.
- Bottom Bar:** http://www.facebook.com/appstack.in.th

**Serial Monitor** (Right Window):

```
AT+CWMODE? +CWMODE:3
OK
AT+CWLAP +CWLAP:(4,"HITRON-90B0",-86,"84:94:8c:e7:90:b8",1)
+CWLAP:(4,"UNUMobile.com.MI",-69,"8c:be:be:2c:f0:43",4)
+CWLAP:(3,"UNUMoblie.com.1F",-39,"78:54:2e:35:aa:48",8)
OK
```

# 更新

- AT 平台一直都在發展 尚有很多 BUG

<https://espressif.com/new-sdk-release-2/>

- 目前版本是

ESP8266 SDK (esp\_iot\_sdk\_v1.0.1\_b1\_15\_04\_02)

- 下載位址

<http://bbs.espressif.com/viewtopic.php>

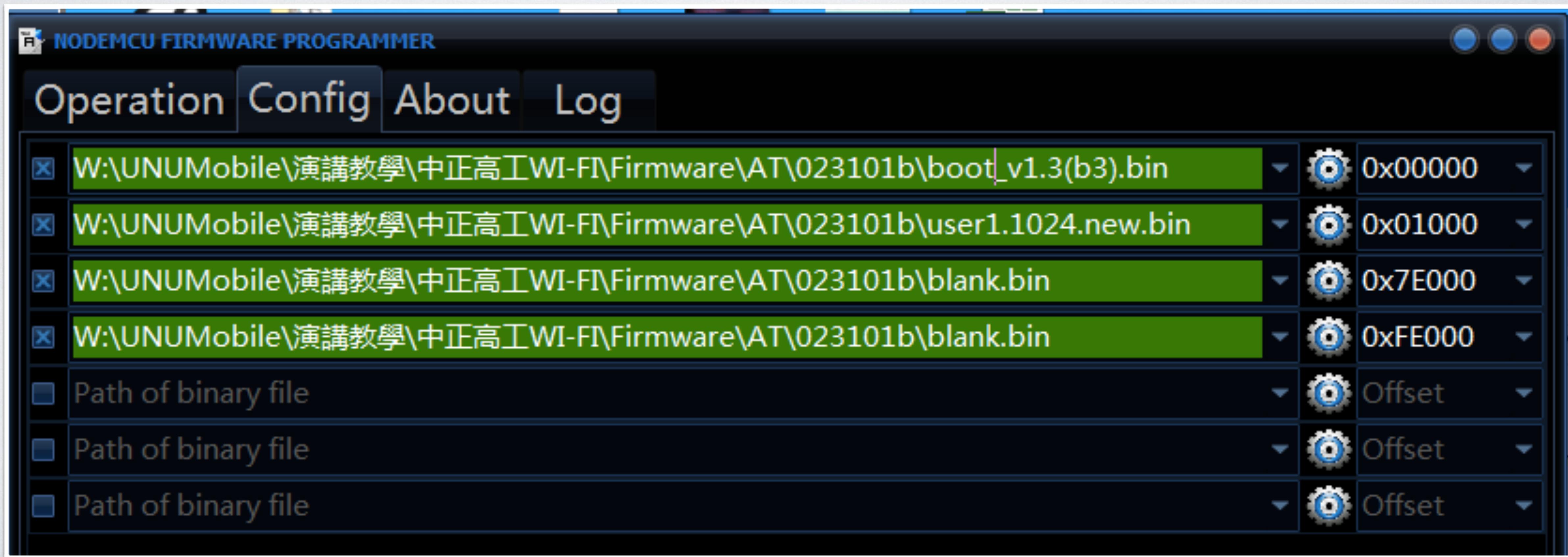
[f=5&t=321&sid=02f8c3b3a9a0045ffec0440a07d70e8e](http://bbs.espressif.com/viewtopic.php?f=5&t=321&sid=02f8c3b3a9a0045ffec0440a07d70e8e)

## TOPICS

- ESP8266 SDK (esp\_iot\_sdk\_v1.0.1\_b1\_15\_04\_02)  
0 by Admin » Fri Apr 03, 2015 12:00 am
- ESP8266 SDK (esp\_iot\_sdk\_v1.0.0\_15\_03\_20)  
0 by Admin » Fri Mar 20, 2015 6:52 pm
- ESP8266 SDK (esp\_iot\_sdk\_v0.9.5\_15\_01\_23)  
0 by Admin » Fri Jan 23, 2015 10:34 pm
- ESP8266 SDK (esp\_iot\_sdk\_v0.9.4\_14\_12\_19)  
0 by Admin » Fri Dec 19, 2014 6:14 pm
- ESP8266 Compiler On Lubuntu  
0 by Admin » Fri Oct 24, 2014 5:47 pm
- ESP8266 AT (at\_v0.20\_14\_11\_28)  
0 by Admin » Fri Nov 28, 2014 6:38 pm
- ESP8266 SDK (esp\_iot\_sdk\_v0.9.3\_14\_11\_21)  
0 by Admin » Fri Nov 21, 2014 6:13 pm
- [Emergency]:esp\_iot\_sdk\_v0.9.3\_14\_11\_21 patch1  
0 by jackson » Sat Nov 22, 2014 11:51 am
- ESP8266 AT (at\_v0.19\_14\_10\_30)  
0 by Admin » Thu Oct 30, 2014 1:55 pm
- ESP8266 SDK (esp\_iot\_sdk\_v0.9.2\_14\_10\_24)  
0 by Admin » Fri Oct 24, 2014 6:01 pm
- ESP8266EX: SDK Patch for UART RX  
by Admin » Sun Nov 02, 2014 10:51 am

# 更新並燒寫

- 路徑 : Firmware\AT\023101b



# LAB 102

## AT指令

Espressif Systems



WIFI\_Command | Arduino 1.0.6

```
#include <SoftwareSerial.h>
SoftwareSerial ESP(2, 3); // RX, TX
#define DEBUG true

void setup()
{
    Serial.begin(9600);
    ESP.begin(9600);
}

void loop() {
    if(ESP.available()) // check if the esp is sending a message
    {
        while(ESP.available())
        {
            // The esp has data so display its output to the serial window
            char c = ESP.read(); // read the next character.
            Serial.write(c);
        }
    }
}
```

/dev/tty.w

AT+GMR

y  
ERROR  
AT

OK  
AT+GMR  
AT version:0.23.b1.0(Apr 2 2015 23:27:52)  
SDK version:1.0.1(b1)  
compile time:Apr 2 2015 23:39:06

OK

# AT 指令練習

- 路徑 : Firmware\AT\023101b\4A-ESP8266 AT 指令\_V0.23b1.pdf
- 原本晶片傳輸是 115200 需要改為 9600 才能供 Arduino 使用  
AT+UART\_DEF = 9600,8,1,0,0
- 開啟 Arduino 燒寫  
Labs\Lab102\Arduino\  
WIFI\_Command

The screenshot shows the Arduino IDE interface. The sketch window contains the following code:

```
WIFI_Command | Arduino 1.0.6
WIFI_Command
#include <SoftwareSerial.h>
SoftwareSerial ESP(2, 3); // RX, TX
#define DEBUG true

void setup()
{
    Serial.begin(9600);
    ESP.begin(9600);
}

void loop()
{
    if(ESP.available()) // check if the esp is sending a message
    {
        while(ESP.available())
        {
            // The esp has data so display its output to the serial window
            char c = ESP.read(); // read the next character.
            Serial.write(c);
        }
    }
}
```

The serial monitor window on the right shows the following output:

```
/dev/tty.w
AT+GMR
y
ERROR
AT
OK
AT+GMR
AT version:0.23.b1.0(Apr 2 2015 23:27:52)
SDK version:1.0.1(b1)
compile time:Apr 2 2015 23:39:06
OK
```

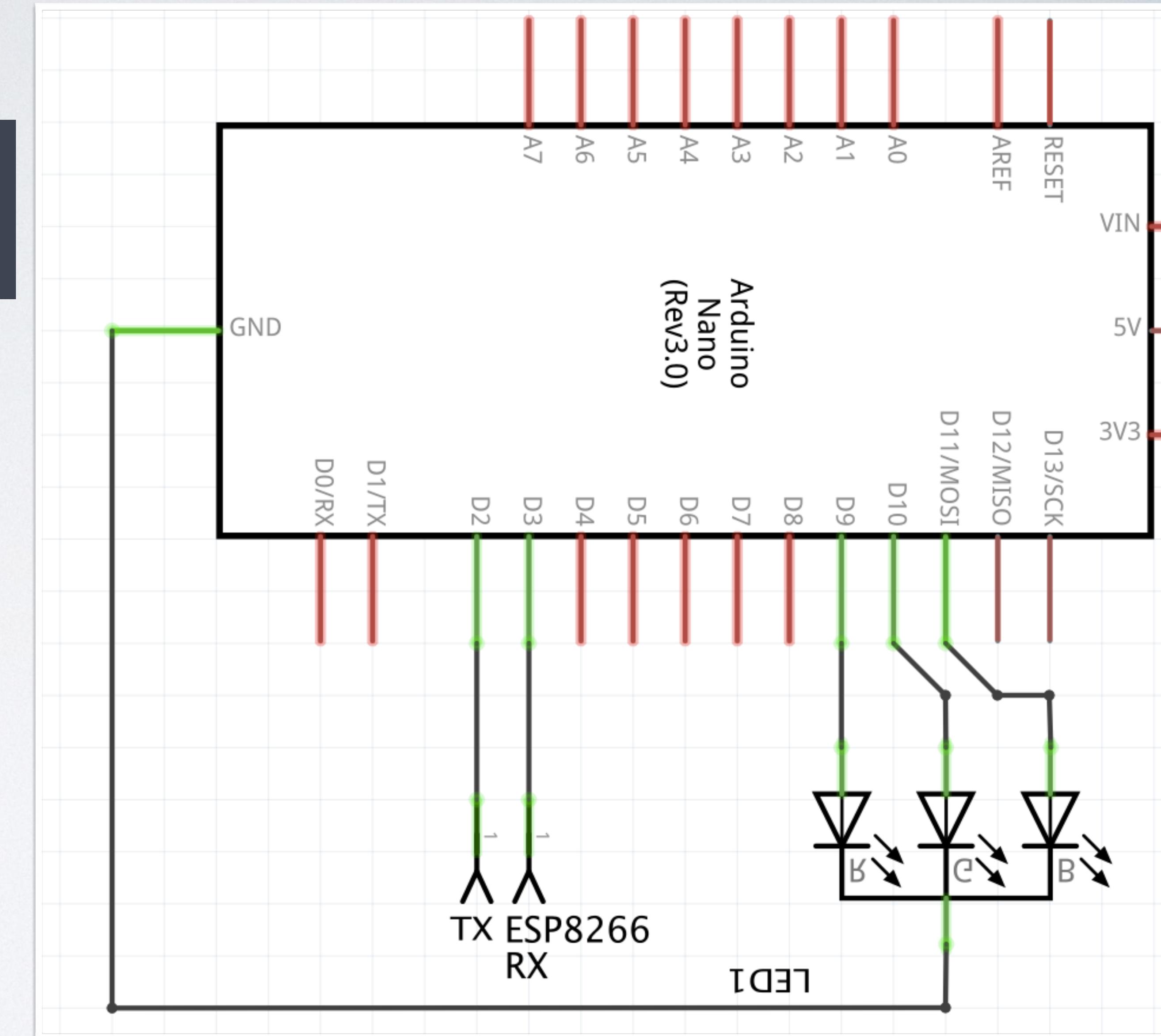
# AT

指令	描述
AT	測試
AT+RST	重啟
AT+GMR	版本
AT+GSLP	進入 deep-sleep
ATE	開關 ECHO 模式
AT+RESTORE	恢複出廠設置
AT+UART_CUR	暫時更改目前傳輸率
AT+UART_DEF	更改目前傳輸率 寫到 FLASH

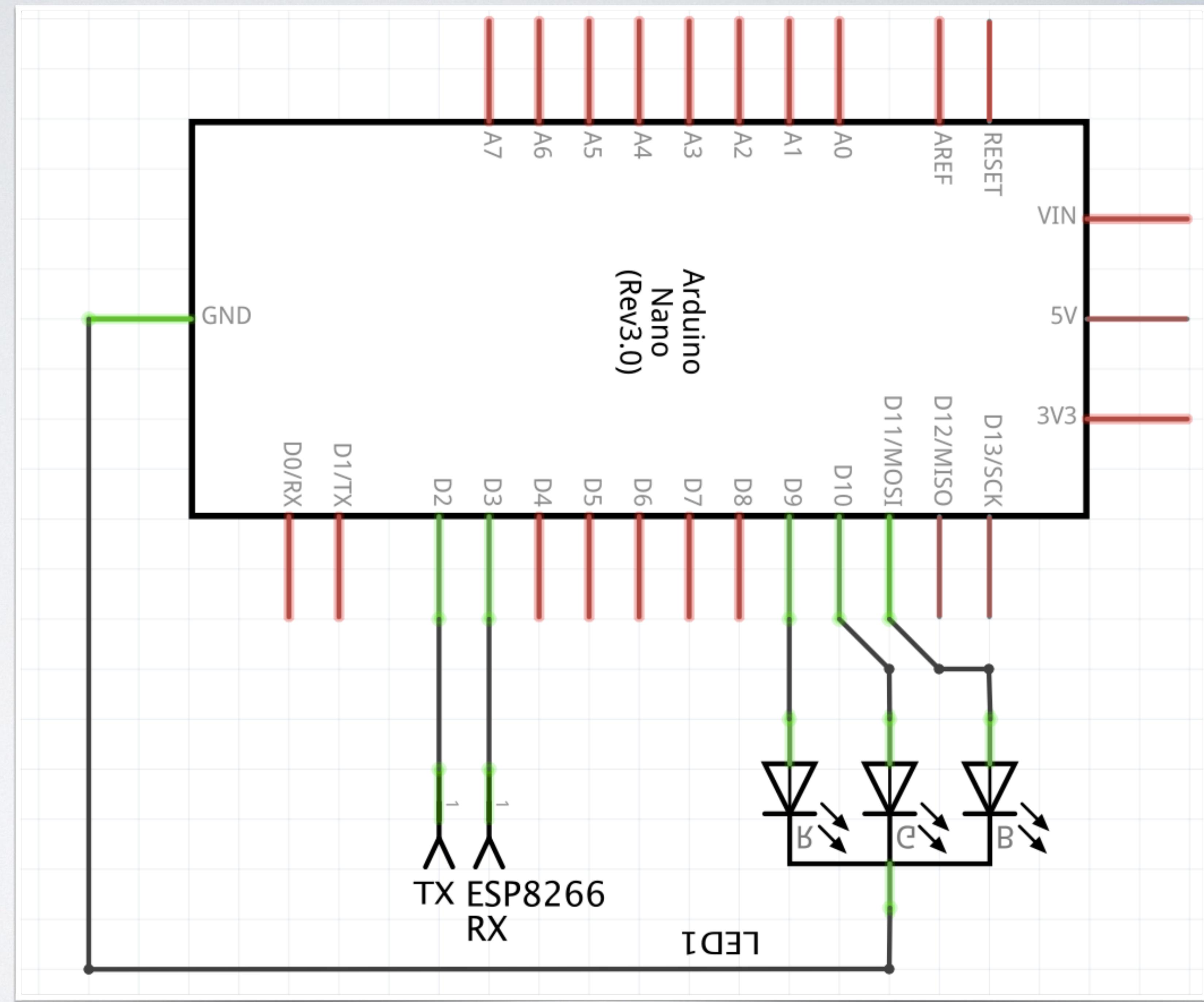
# LAB 103

## SERVER

建立 Server 並提供內容



# 線路連接



# 撰寫 ARDUINO 和網頁 程式

- 路徑

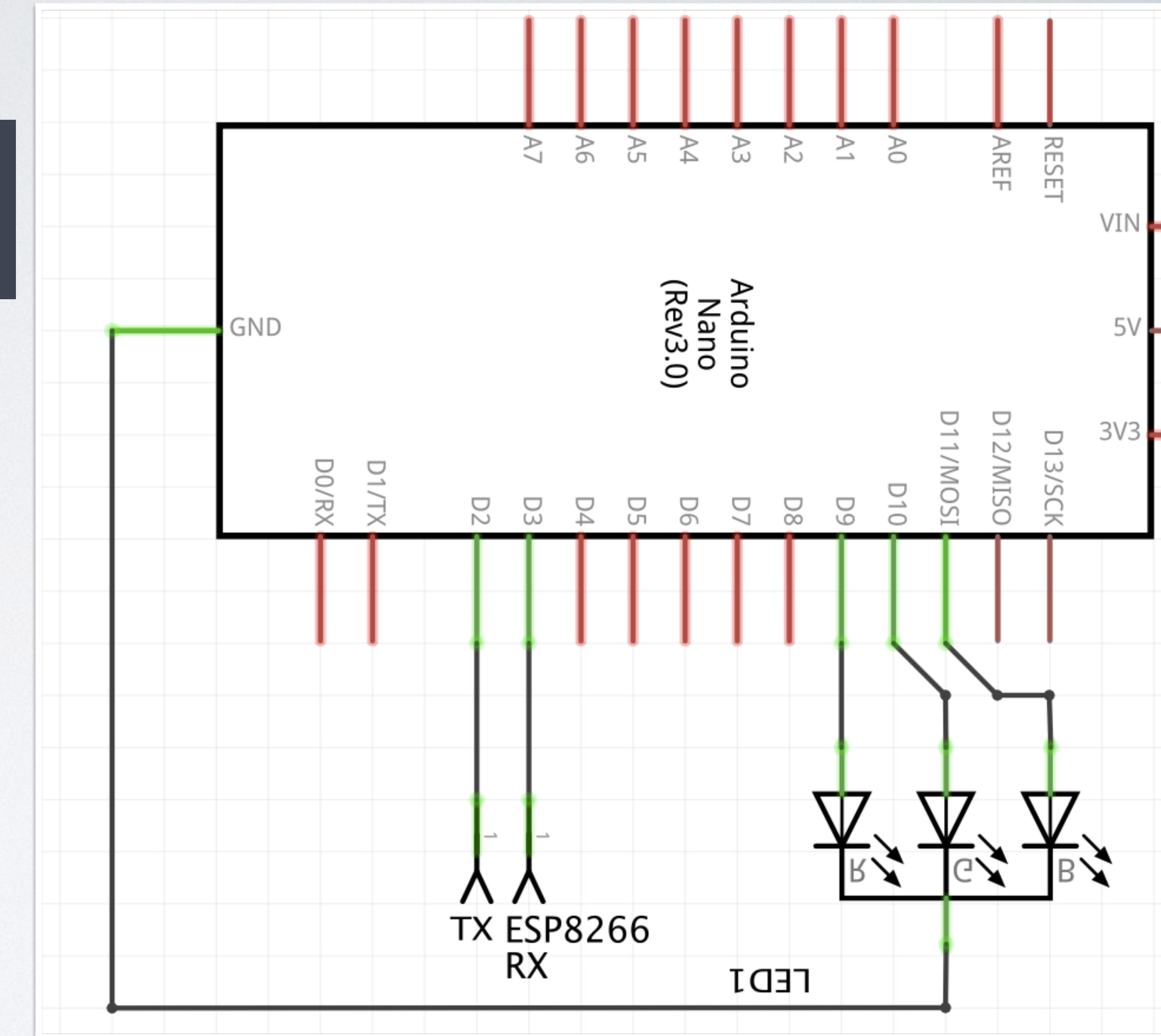
Labs\Lab103\Arduino\WIFI\_SERVER



# LAB 104

## RGB LED

Arduino 透過 AT 將 ESP8266 設置為 SERVER, 透過將頁控制 RGB LED



# 撰寫 ARDUINO 和網頁 程式

- 路徑

Labs\Lab104\Arduino\WIFI\_LED



# 改寫為 PWM 選色

- 路徑

Labs\Lab104\Arduino  
\WIFI\_LEDPWM



# LAB 105

NODEMCU

透過 LUA 腳本開發 IoT



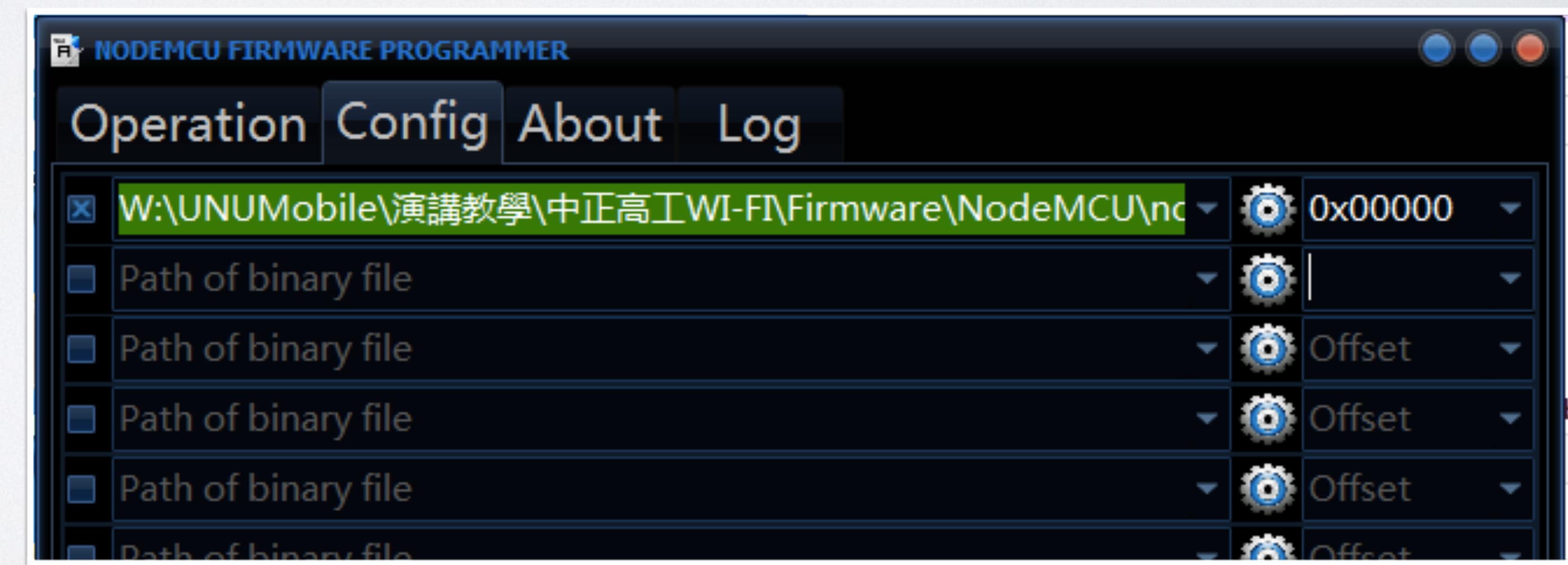
# 下載並燒寫 NODEMCU

- 下載 Firmware

<http://bbs.nodemcu.com/t/nodemcu-firmware-download-build-20150318-new-location/27>

- 路徑

Firmware\NodeMCU



# LUA 腳本

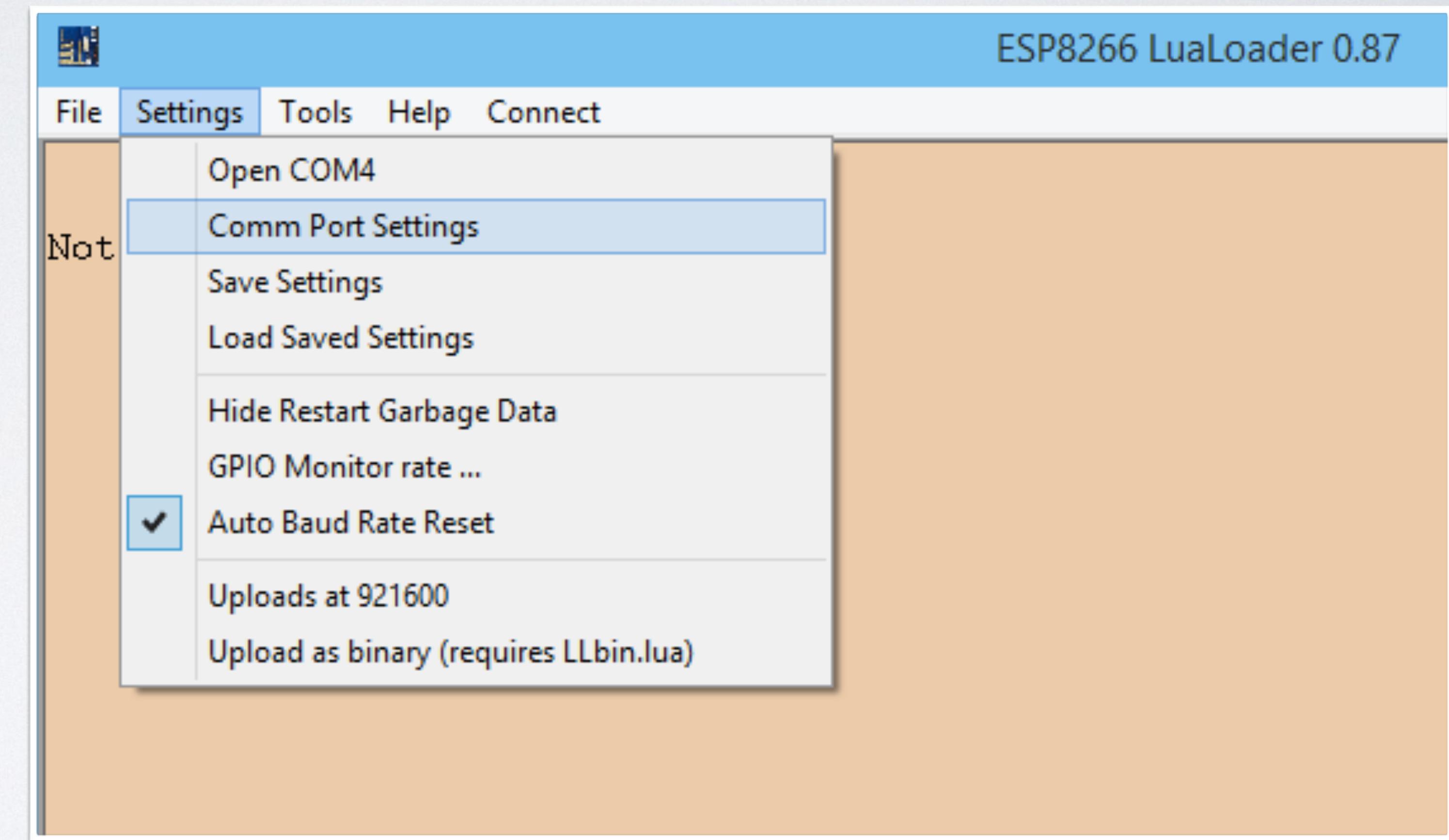
- <http://www.lua.org/>
- NodeMCU API 手冊  
路徑：Firmware/NodeMCU/  
nodeMcuAPI簡中.pdf  
NodeMCU API Instruction En.pdf



# 工具

## LuaLoader

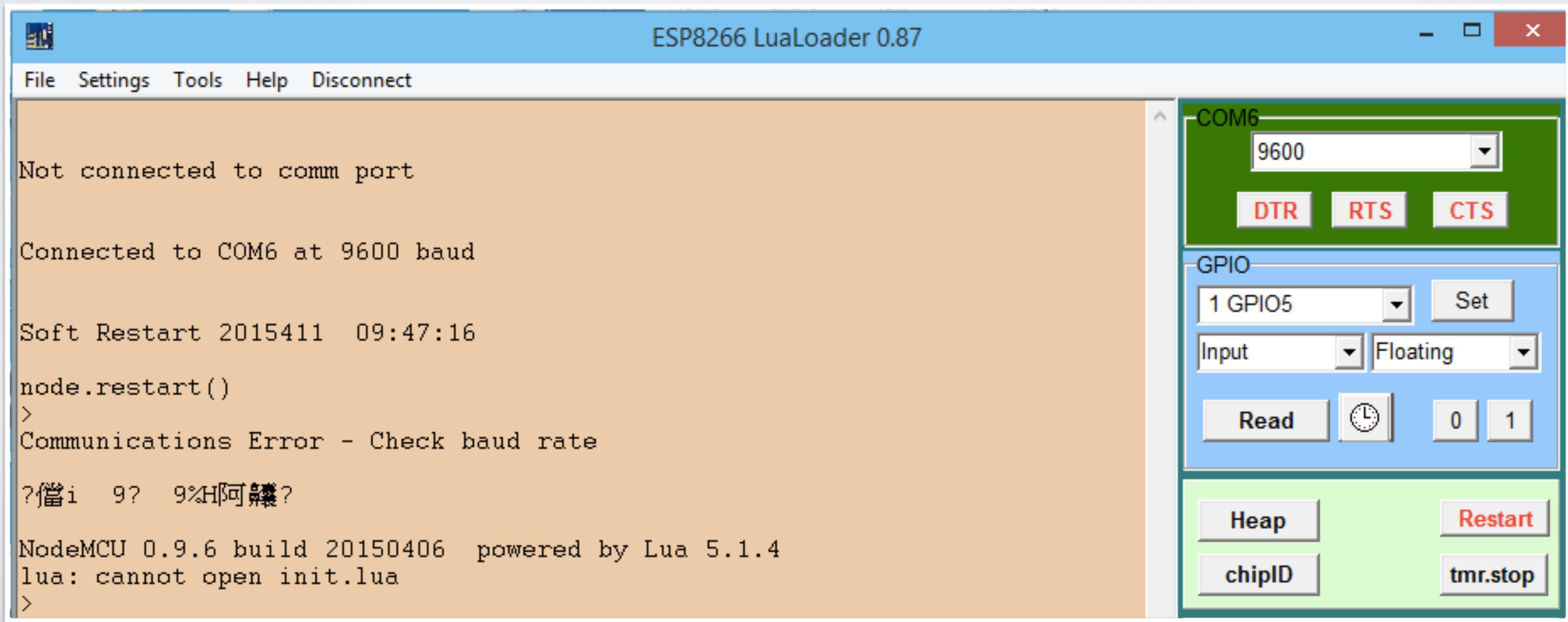
- 設定 PORT
- 再點選 Open Comx



# 工具

## LuaLoader

- 按下右側的 Restart 或是輸入 node.restart()  
正常情況下會看到 NodeMCU x.x.x build ... 等資訊



# GPIO

- 輸出GPIO0  
`gpio.mode(3, gpio.OUTPUT)`
- 將 GPIO0 輸出高電位  
`gpio.write(3, gpio.HIGH)`
- 讀取  
`gpio.mode(3, gpio.INPUT, gpio.FLOAT)`  
`gpio.read(3)`

index	pin
0	[*] GPIO16
1	GPIO4
2	GPIO5
3	GPIO0
4	GPIO2
5	GPIO14
6	GPIO12
7	GPIO13
8	GPIO15
9	GPIO3
10	GPIO1
11	GPIO9
12	GPIO10

# 原始碼

- <https://github.com/nodemcu/nodemcu-firmware>

# LAB 106

## NODEMCU

撰寫 Lua 讓ESP8266一啟動即連網並取得IP



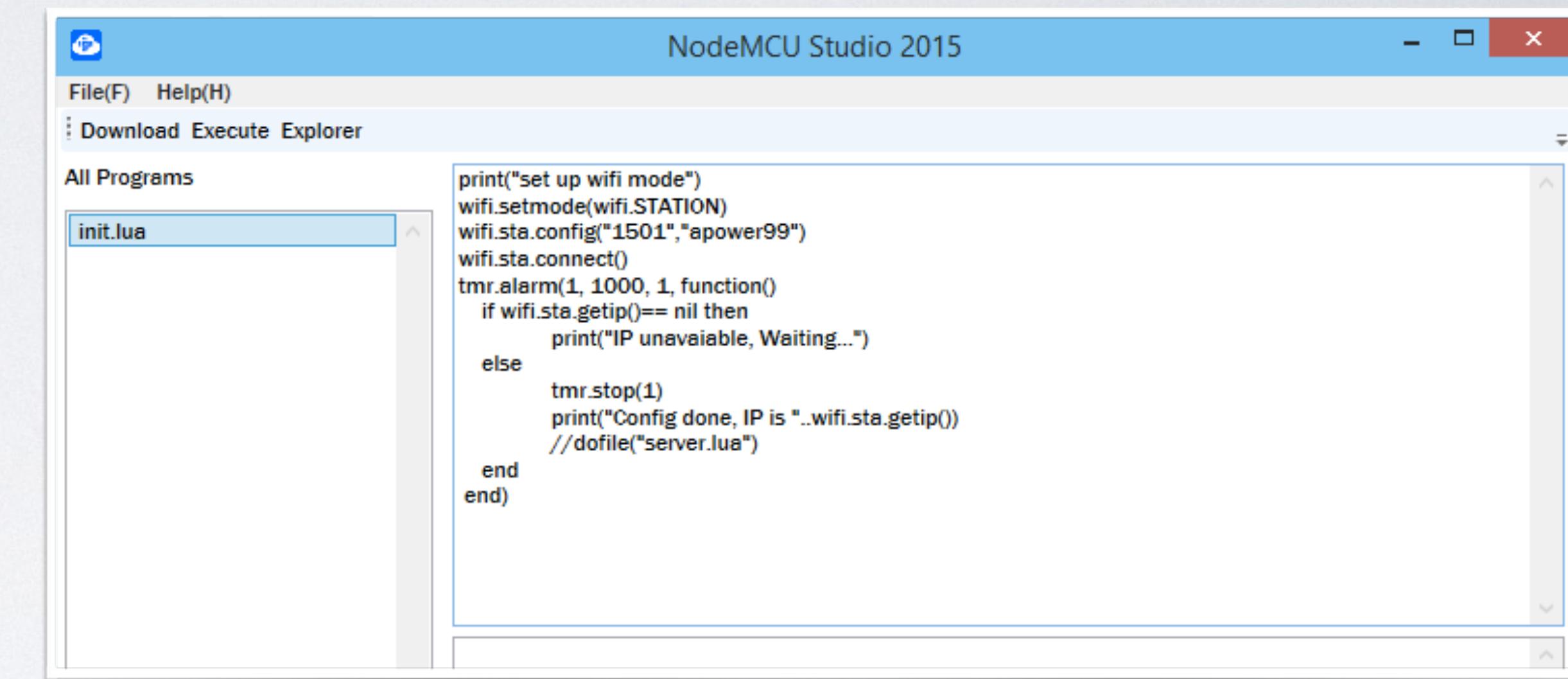
# INIT.LUA

- NodeMCU — 啟動自動執行的腳本

```
print("set up wifi mode")
wifi.setmode(wifi.STATION)
wifi.sta.config("1501","apower99")
wifi.sta.connect()
tmr.alarm(1, 1000, 1, function()
    if wifi.sta.getip()== nil then
        print("IP unavaiable, Waiting...")
    else
        tmr.stop(1)
        print("Config done, IP is "..wifi.sta.getip())
        dofile("server.lua")
    end
end)
```

# 燒寫

- 將 Lab106 的 \*.lua 覆寫到 Tools  
\\Lua Programs
- 工具 NodeMCU Studio 2015  
路徑 : Tools/NodeMCU Studio  
2015\_beta0.2.exe
- 按下 Download 即可寫入



The screenshot shows the NodeMCU Studio 2015 interface. The title bar reads "NodeMCU Studio 2015". The menu bar includes "File(F)", "Help(H)", "Download", "Execute", and "Explorer". The "All Programs" list on the left shows "init.lua" selected. The main code editor window displays the following Lua script:

```
print("set up wifi mode")
wifi.setmode(wifi.STATION)
wifi.sta.config("1501","apower99")
wifi.sta.connect()
tmr.alarm(1, 1000, 1, function()
    if wifi.sta.getip()== nil then
        print("IP unavailable, Waiting...")
    else
        tmr.stop(1)
        print("Config done, IP is "..wifi.sta.getip())
        //dofile("server.lua")
    end
end)
```

# LAB 107

NODEMCU

取得遠端 API 服務並解析資料



# 解析字串

- 字串處理教學

[https://www.ptt.cc/bbs/mud\\_sanc/M.1371321418.A.995.html](https://www.ptt.cc/bbs/mud_sanc/M.1371321418.A.995.html)

- JSON 物件

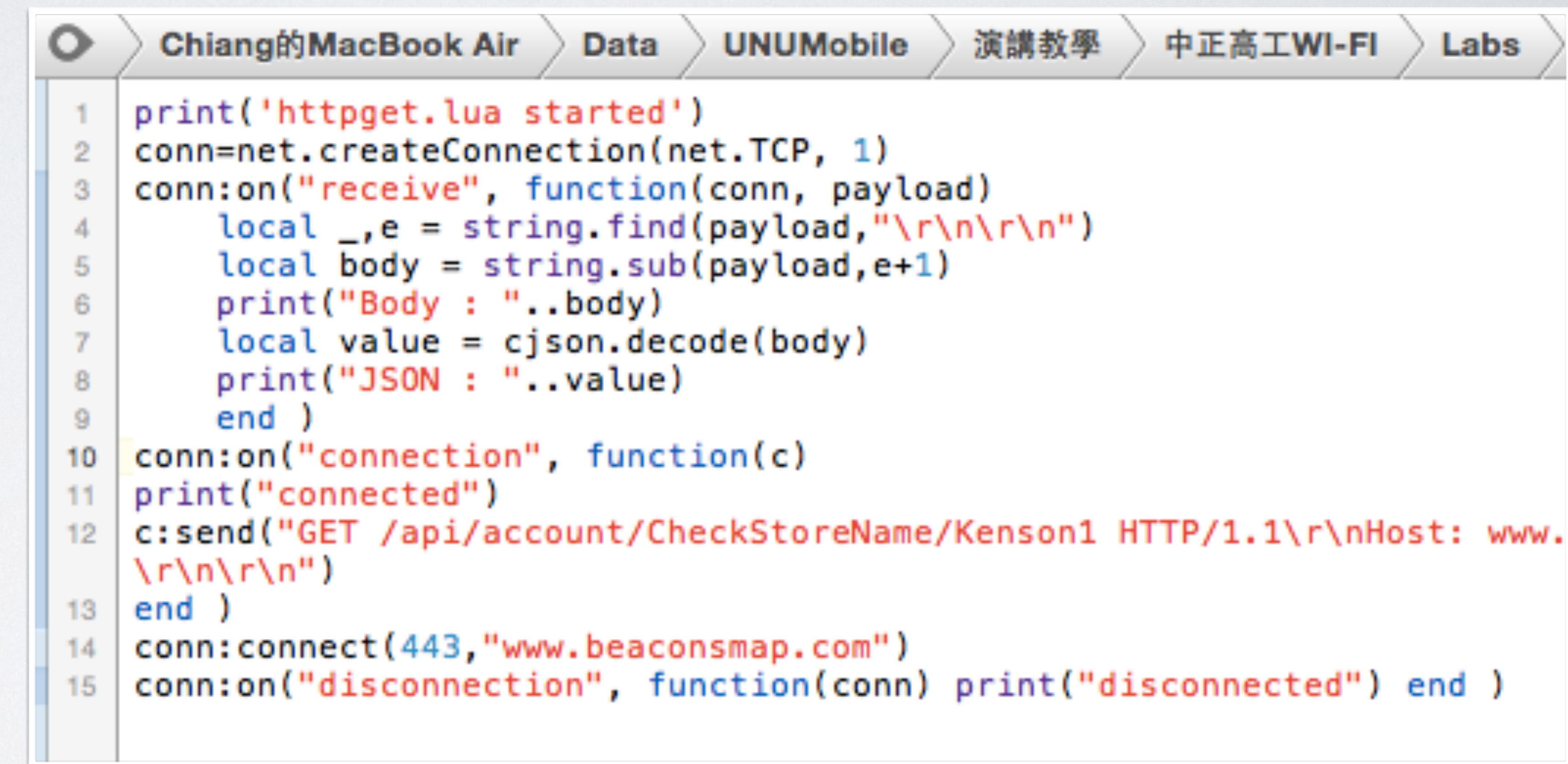
<http://www.kyne.com.au/~mark/software/lua-cjson-manual.html>

- Table 物件

<http://lua-users.org/wiki/TablesTutorial>

# 處理取得物件

- 4~8行處理  
HEADER 和 JSON



The screenshot shows a mobile browser interface with the following navigation bar:

- Chiang's MacBook Air
- Data
- UNUMobile
- 演講教學
- 中正高工WI-FI
- Labs

The main content area displays a Lua script:

```
1 print('httpget.lua started')
2 conn=net.createConnection(net.TCP, 1)
3 conn:on("receive", function(conn, payload)
4     local _,e = string.find(payload, "\r\n\r\n")
5     local body = string.sub(payload,e+1)
6     print("Body : "..body)
7     local value = cjson.decode(body)
8     print("JSON : "..value)
9     end )
10 conn:on("connection", function(c)
11     print("connected")
12     c:send("GET /api/account/CheckStoreName/Kenson1 HTTP/1.1\r\nHost: www.
13 \r\n\r\n")
14     end )
15 conn:connect(443,"www.beaconsmap.com")
16 conn:on("disconnection", function(conn) print("disconnected") end )
```

# LAB 108

NODEMCU

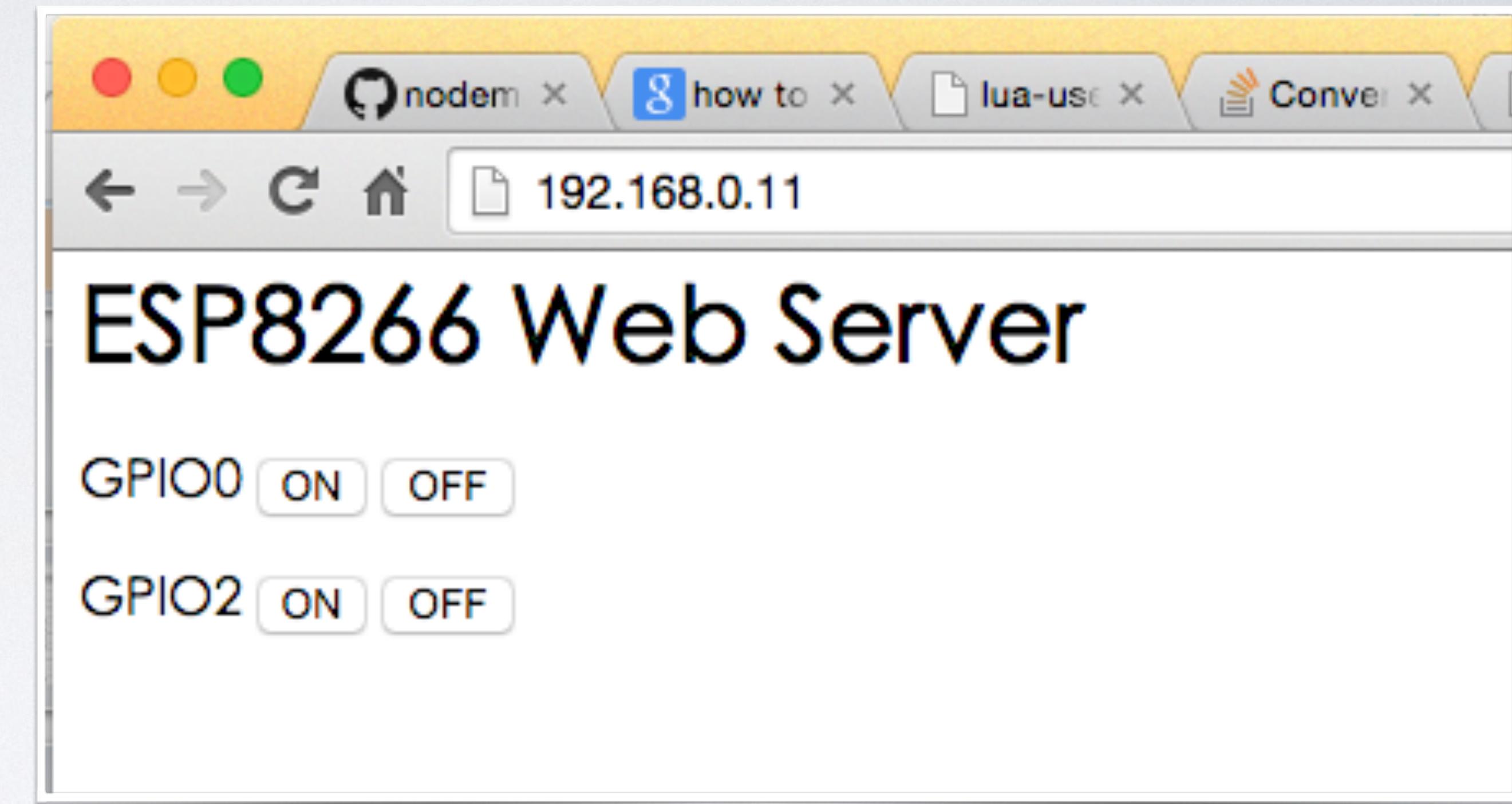
獨立運作的 WWW Server  
控制 GPIO 0 和 GPIO 4



# WEB SERVER

- 路徑  
Labs\Lab\08
- 執行並取得 IP

```
NodeMCU 0.9.6 build 20150406  powered by Lua 5.1.4
set up wifi mode
> IP unavailable, Waiting...
Config done, IP is 192.168.0.11
```





<http://www.unumobile.com>

通聯行動創意科技有限公司  
UNION U INC.

83144 高雄市大寮區承德街5號1F / 電郵 : [service@unumobile.com](mailto:service@unumobile.com) / 統一編號 : 54631308  
No.5, Chengde St., Daliao Dist., Kaohsiung City 83144, Taiwan (R.O.C.)  
© Union U Inc. 通聯行動創意科技有限公司 2014 All Rights Reserved