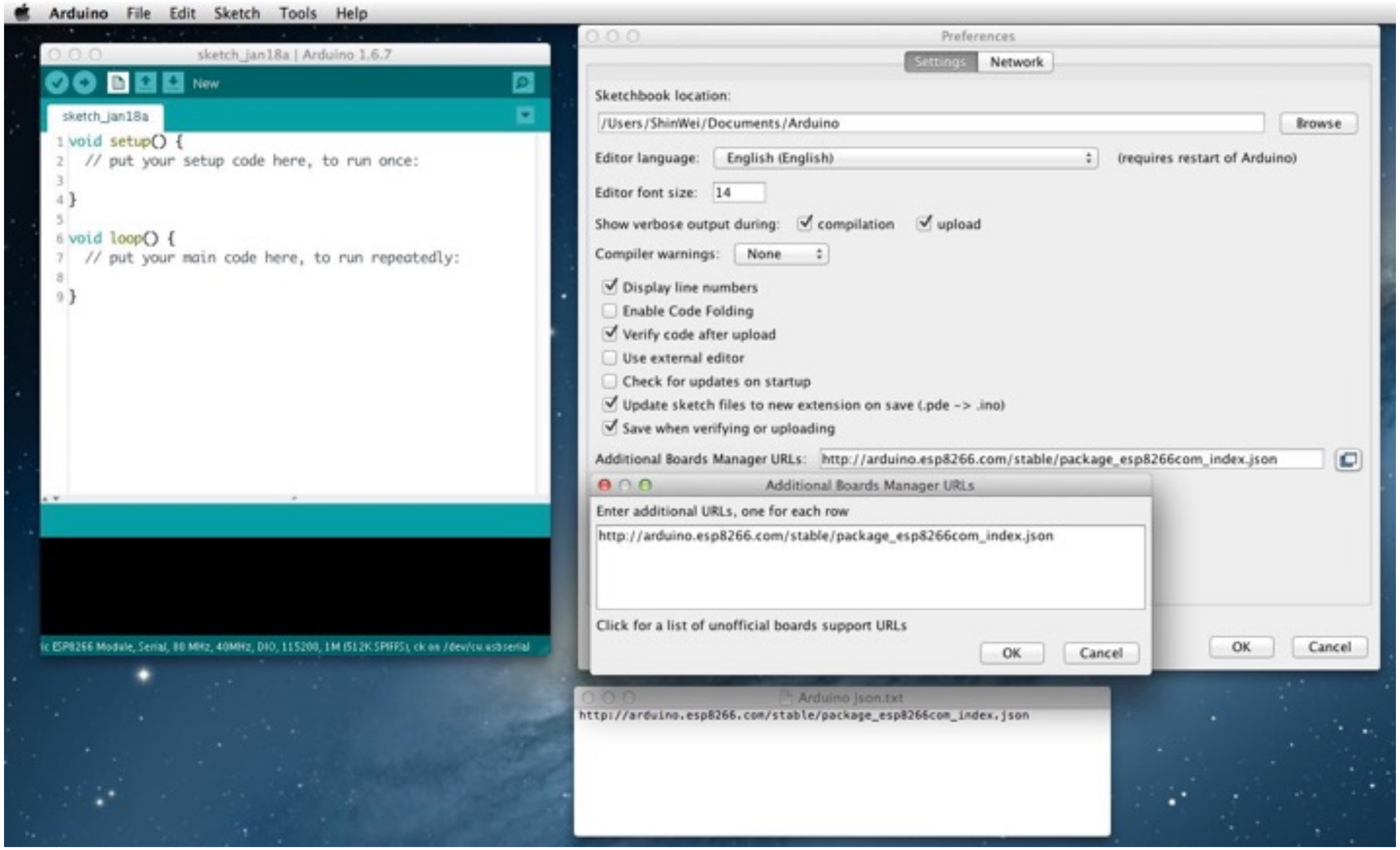
MiniPlan 軟體篇

Roy Chen

設定Arduino-ESP8266

- 額外的板子管理員網址
  http://arduino.esp8266.com/stable/package_esp8266com_index.json


Arduino File Edit Sketch Tools Help
sketch_jan18a | Arduino 1.6.7
New
sketch_jan18a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

ic ESP8266 Module, Serial, 80 MHz, 40MHz, DIO, 115200, 1M (512K SPIFFS), ck on /dev/cu.usbserial

Preferences
Settings
Network

Sketchbook location: /Users/ShinWei/Documents/Arduino Browse

Editor language: English (English) (requires restart of Arduino)

Editor font size: 14

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

☑ Display line numbers
☐ Enable Code Folding
☑ Verify code after upload
☐ Use external editor
☐ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Enter additional URLs, one for each row

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Click for a list of unofficial boards support URLs

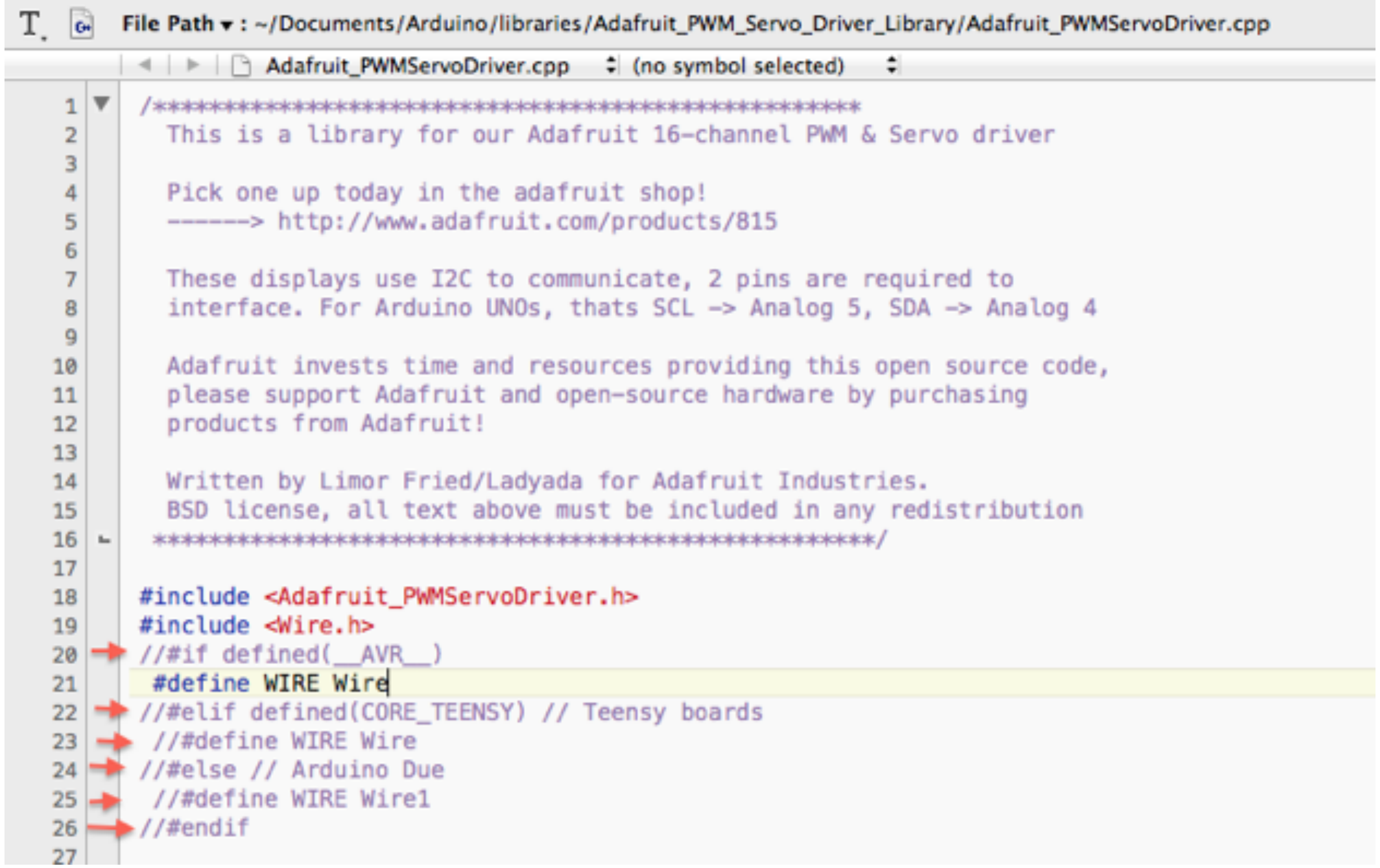
http://arduino.esp8266.com/stable/package_esp8266com_index.json

設定Arduino-程式庫

- 加入函式庫 -> 程式庫管理員 ->搜尋”adafruit pwm servo”
- 安裝”Adafruit PWM Servo Driver Library”

設定Arduino-程式庫

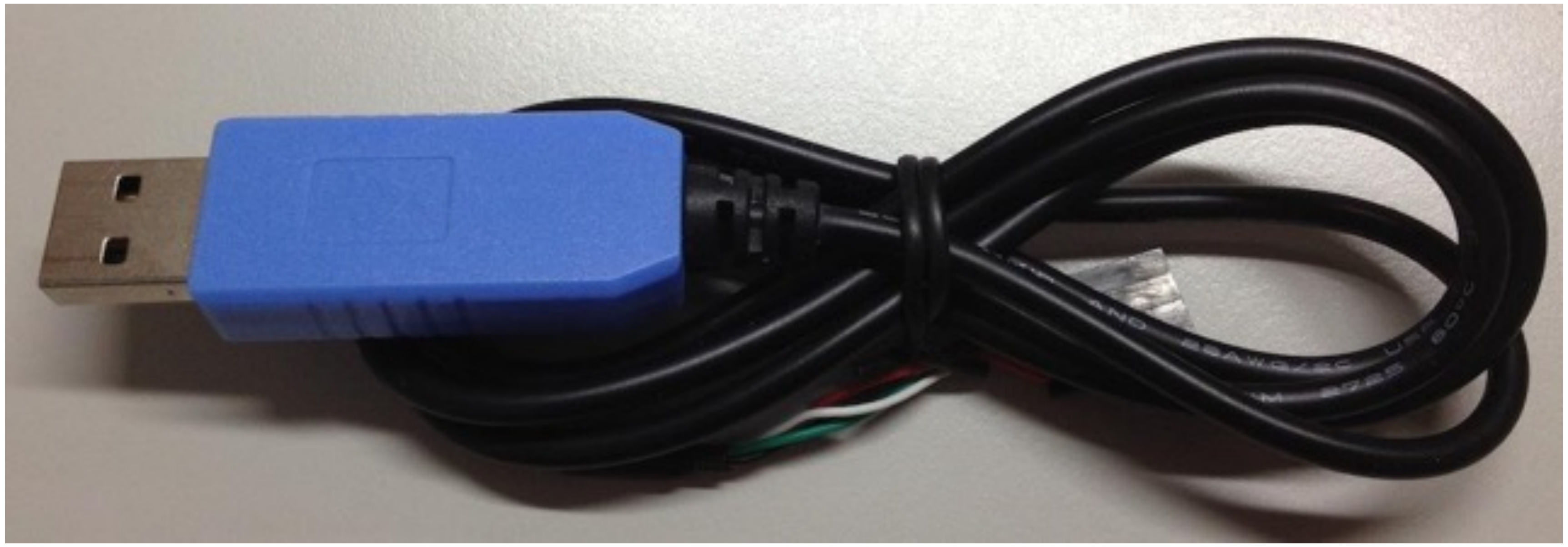
- 開啟/Arduino/libraries/Adafruit_PWM_Servo_Driver_Library/Adafruit_PWMServoDriver.cpp, 並且做下列修正：


File Path ▾ : ~/Documents/Arduino/libraries/Adafruit_PWM_Servo_Driver_Library/Adafruit_PWMServoDriver.cpp
Adafruit_PWMServoDriver.cpp
(no symbol selected)

```
/***************************************************
  This is a library for our Adafruit 16-channel PWM & Servo driver

  Pick one up today in the adafruit shop!
  ------> http://www.adafruit.com/products/815

  These displays use I2C to communicate, 2 pins are required to
  interface. For Arduino UNOs, thats SCL -> Analog 5, SDA -> Analog 4

  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!

  Written by Limor Fried/Ladyada for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ****************************************************/

#include <Adafruit_PWMServoDriver.h>
#include <Wire.h>
//#if defined(__AVR__)
 #define WIRE Wire
//#elif defined(CORE_TEENSY) // Teensy boards
 //#define WIRE Wire
//#else // Arduino Due
 //#define WIRE Wire1
//#endif
```

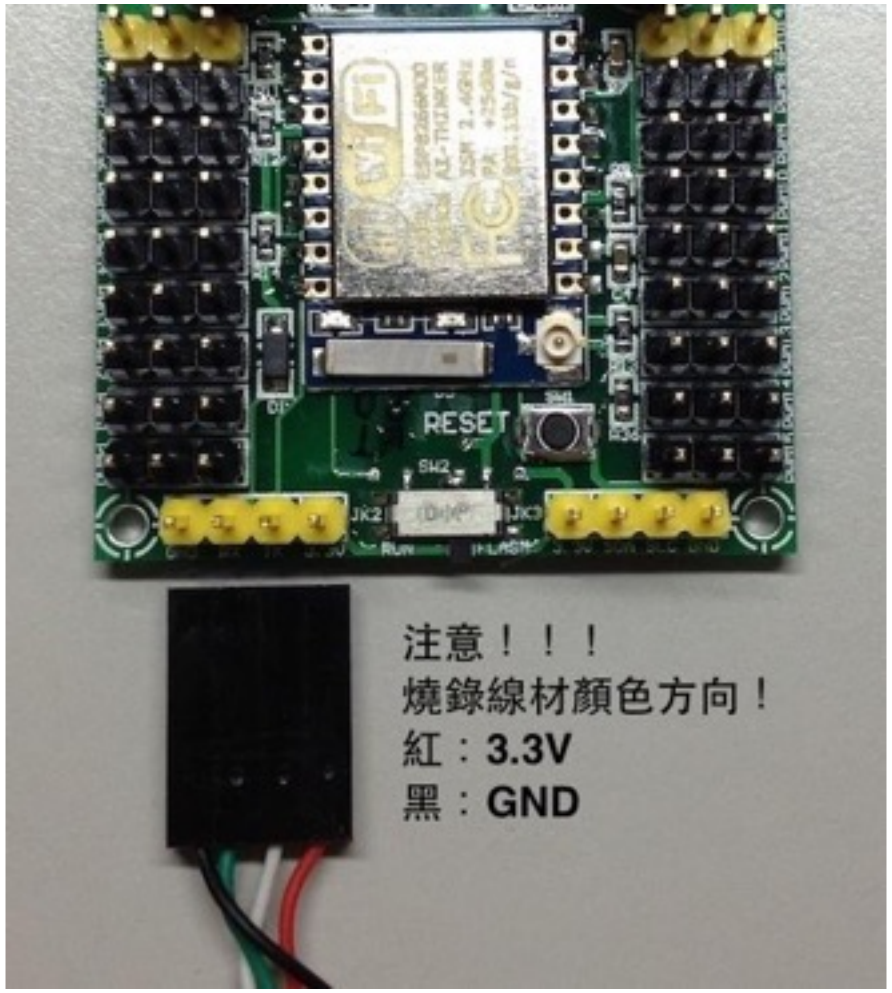
透過Arduino-IDE與機器人進行通訊

- 安裝驅動程式 http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41



透過Arduino-IDE與機器人進行通訊

- 設定Arduino-IDE連結方式

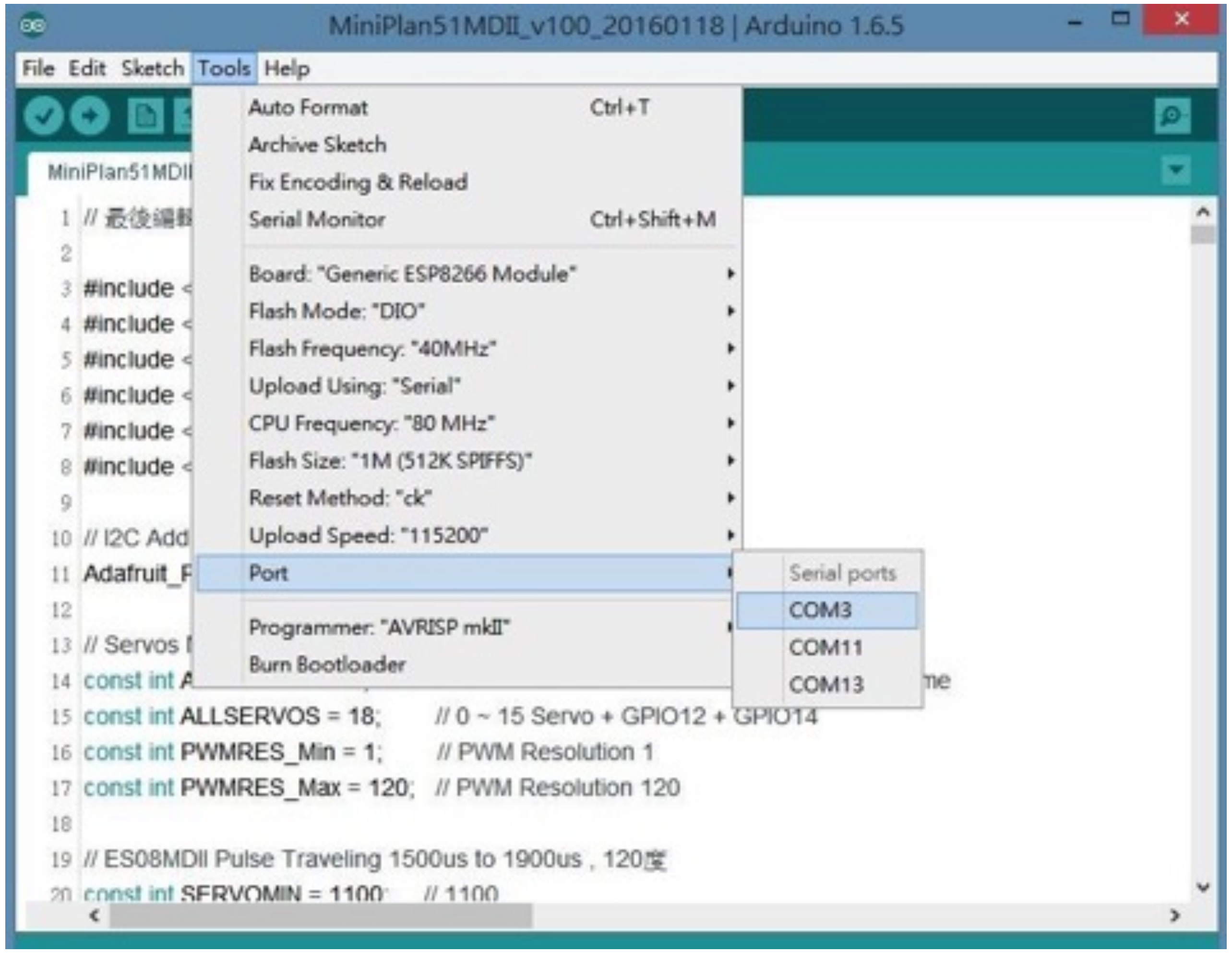


注意！！！

燒錄線材顏色方向！

紅：3.3V

黑：GND



Board: "Generic ESP8266 Module"

Flash Mode: "DIO"

Flash Frequency: "40MHz"

Upload Using: "Serial"

CPU Frequency: "80 MHz"

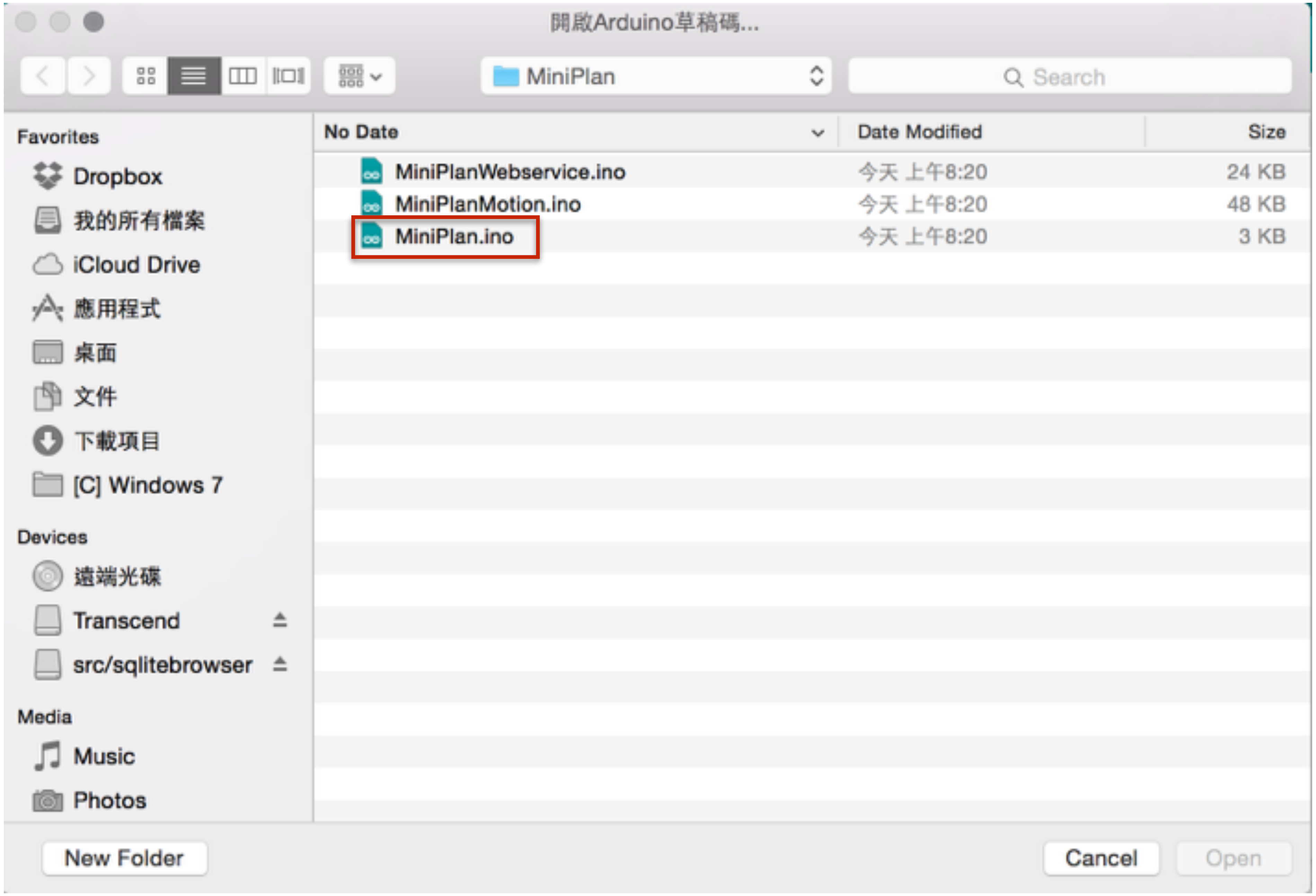
Flash Size: "1M (512K SPIFFS)"

Reset Method: "ck"

Upload Speed: "115200"

```
// 最後編輯

#include <
#include <
#include <
#include <
#include <
#include <

// I2C Add
Adafruit_P

// Servos I
const int A                                                me
const int ALLSERVOS = 18;      // 0 ~ 15 Servo + GPIO12 + GPIO14
const int PWMRES_Min = 1;      // PWM Resolution 1
const int PWMRES_Max = 120;    // PWM Resolution 120

// ES08MDII Pulse Traveling 1500us to 1900us , 120度
const int SERVOMIN = 1100;     // 1100
```

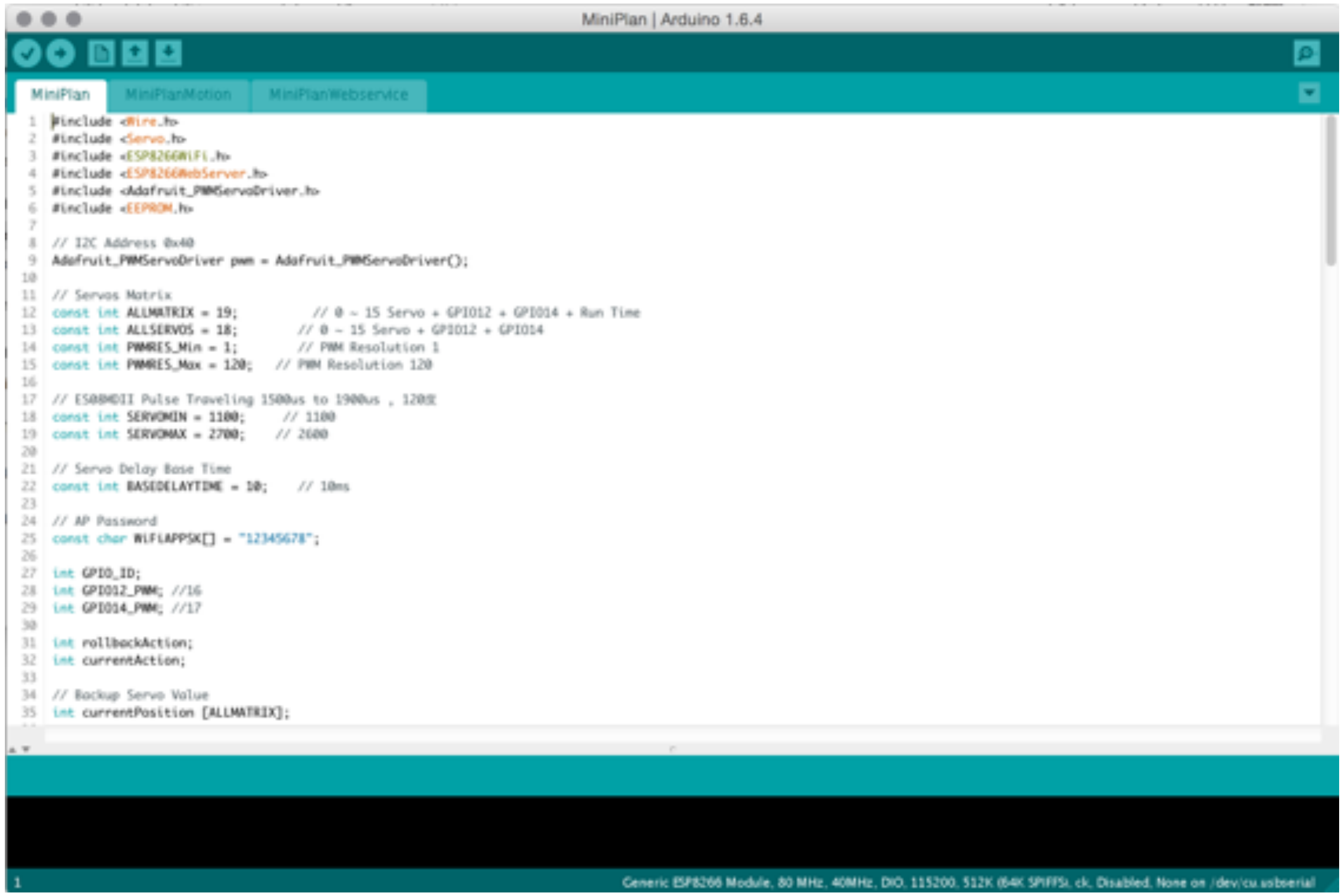
開啟MiniPlan專案



No Date	Date Modified	Size
MiniPlanWebservice.ino	今天 上午8:20	24 KB
MiniPlanMotion.ino	今天 上午8:20	48 KB
MiniPlan.ino	今天 上午8:20	3 KB

Arduino-主畫面

- MiniPlan
- MiniPlanMotion
- MinPlanWebservice



```
#include <Wire.h>
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Adafruit_PWMServoDriver.h>
#include <EEPROM.h>

// I2C Address 0x40
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

// Servos Matrix
const int ALLMATRIX = 19;          // 0 ~ 15 Servo + GPIO12 + GPIO14 + Run Time
const int ALLSERVOS = 18;        // 0 ~ 15 Servo + GPIO12 + GPIO14
const int PWMRES_Min = 1;        // PWM Resolution 1
const int PWMRES_Max = 120;   // PWM Resolution 120

// ES08MDII Pulse Traveling 1500us to 1900us , 120度
const int SERVOMIN = 1100;     // 1100
const int SERVOMAX = 2700;    // 2600

// Servo Delay Base Time
const int BASEDELAYTIME = 10;    // 10ms

// AP Password
const char WiFiAPPSK[] = "12345678";

int GPIO_ID;
int GPIO12_PWM; //16
int GPIO14_PWM; //17

int rollbackAction;
int currentAction;

// Backup Servo Value
int currentPosition [ALLMATRIX];
```

```
1
Generic ESP8266 Module, 80 MHz, 40MHz, DIO, 115200, 512K (64K SPIFFS), ck, Disabled, None on /dev/cu.usbserial
```

MiniPlan

- 初始化MiniPlan的設定
    - AP Mode設定
    - PWM元件初始化
    - 啟用網頁伺服器

MiniPlanMotion

- 設定MiniPlan的動作

MiniPlanWebservice

- 實作Webservice API

如何增加動作

- 設計Webservice API
- 設計動作矩陣
- 將API與動作矩陣做連結

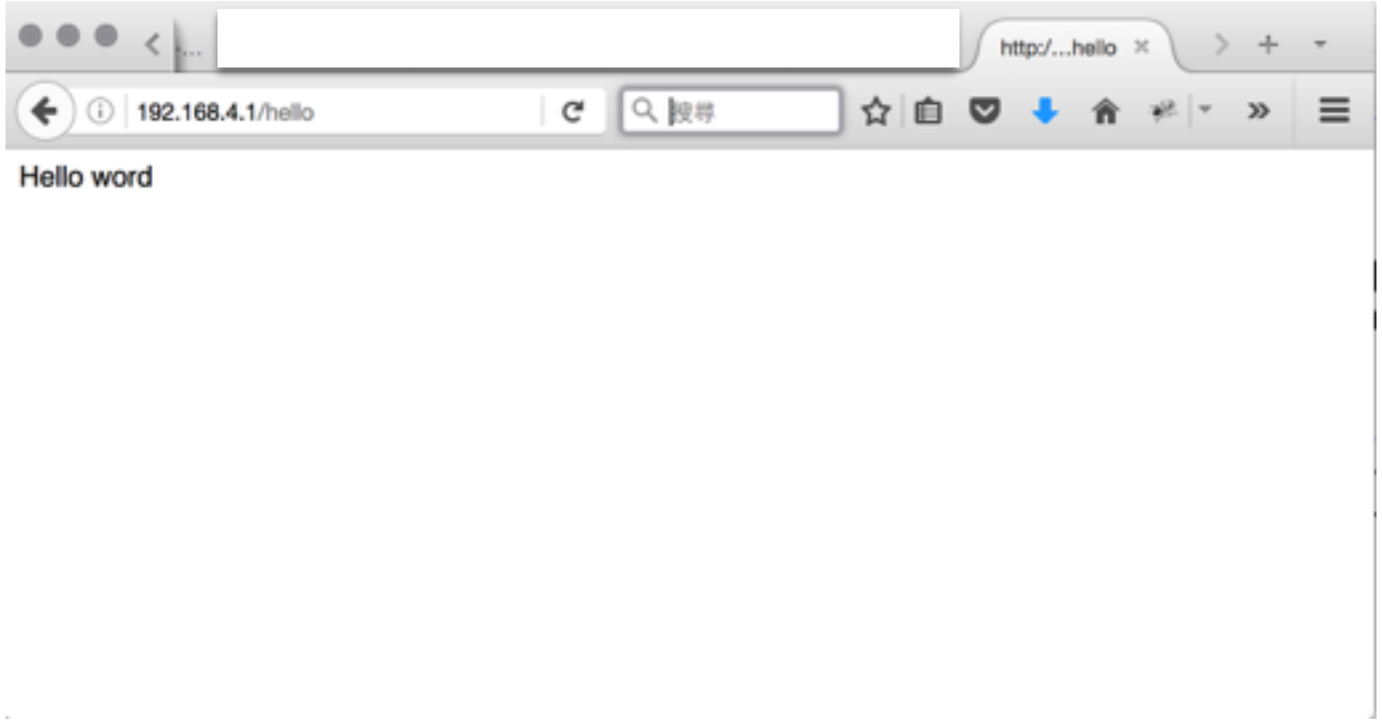
增加Webservice API

- 在MiniPlanWebservice加入下列的程式碼

```
void handleHello(){
  String content = "";
  content = "<html>";
  content += "<body>";
  content += "Hello word";
  content += "</body>";
  content += "</html>";
  server.send(200, "text/html", content);
}

void enableWebServer(){
  HTTPMethod getMethod = HTTP_GET;
  server.on("/", getMethod, handleIndex);
  server.on("/zero", getMethod, handleZero);
  server.on("/editor", getMethod, handlerEditor);
  server.on("/save", getMethod, handleSave);
  server.on("/controller", getMethod, handleController);
  server.on("/action", getMethod, handleAction);
  server.on("/motor", getMethod, handleMotor);
  server.on("/reset", getMethod, handleReset);
  server.on("/setting", getMethod, handleSetting);
  server.on("/hello", getMethod, handleHello);
  server.begin();
  Serial.println("service enable");
}
```

Webservice API結果

- 在url輸入192.168.4.1/hello會顯示下列的畫面



Hello word

增加動作矩陣-1

- 何謂動作矩陣

```
int action00 [ ] PROGMEM = {   65,      35,      80,      60,
  80,      100,      95,       80,
  40,       25,        20,        30,
  55,        35,        75,        50,
  90,        90,        0        };
```

Servo:0-15

Servo:16, 17　執行時間

增加動作矩陣-2

- 如何觸發動作

```
void resetToStand()
{
  for ( int index = 0; index < ALLMATRIX; index++)
  {
    currentPosition[index] = action00[index] + readKeyValue(index);
  }
  for (int iServo = 0; iServo < ALLSERVOS; iServo++)
  {
    setPWMVal(iServo, currentPosition[iServo]);
    delay(10);
  }
  Serial.print("resetToStand complete:");
}
```

Servo 控制

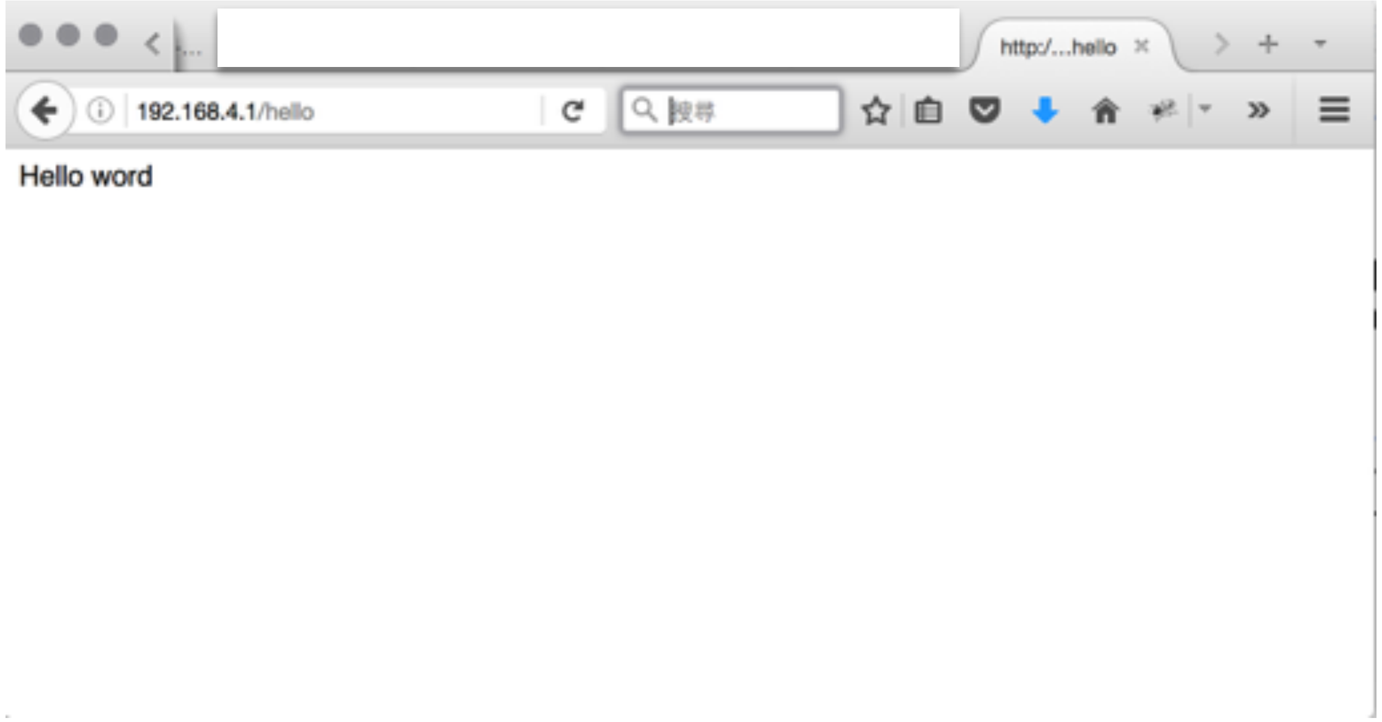
連結API與動作矩陣

- 在MiniPlan加入下列程式碼

```
void handleHelloAction(){
  String content = "";
  content = "<html>";
  content += "<body>";
  content += "Hello word";
  content += "</body>";
  content += "</html>";
  resetToHello();
  server.send(200, "text/html", content);
}
```

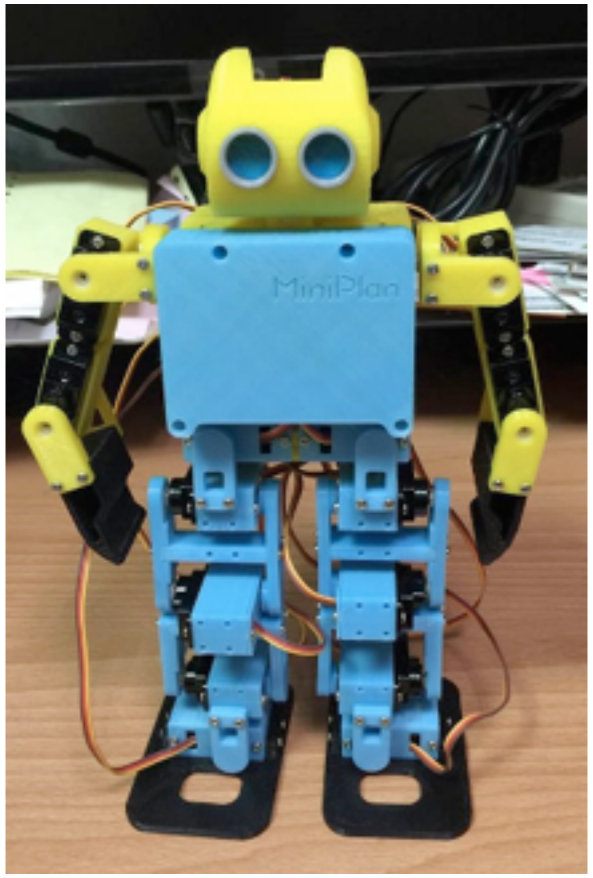
Webservice API結果

- 在url輸入192.168.4.1/hello會顯示下列的畫面以及機器人會做動作



192.168.4.1/hello

Hello word



https://github.com/makee-workshop/Miniplan-workshop

Makee