

# MiniPlan 軟體篇

Roy Chen

# 相關影片

- <https://www.youtube.com/watch?v=L6SdtgnziMU>
- <https://www.facebook.com/unwirehk/videos/10157574962865495/>

# 蜘蛛機器人



# 機械手臂

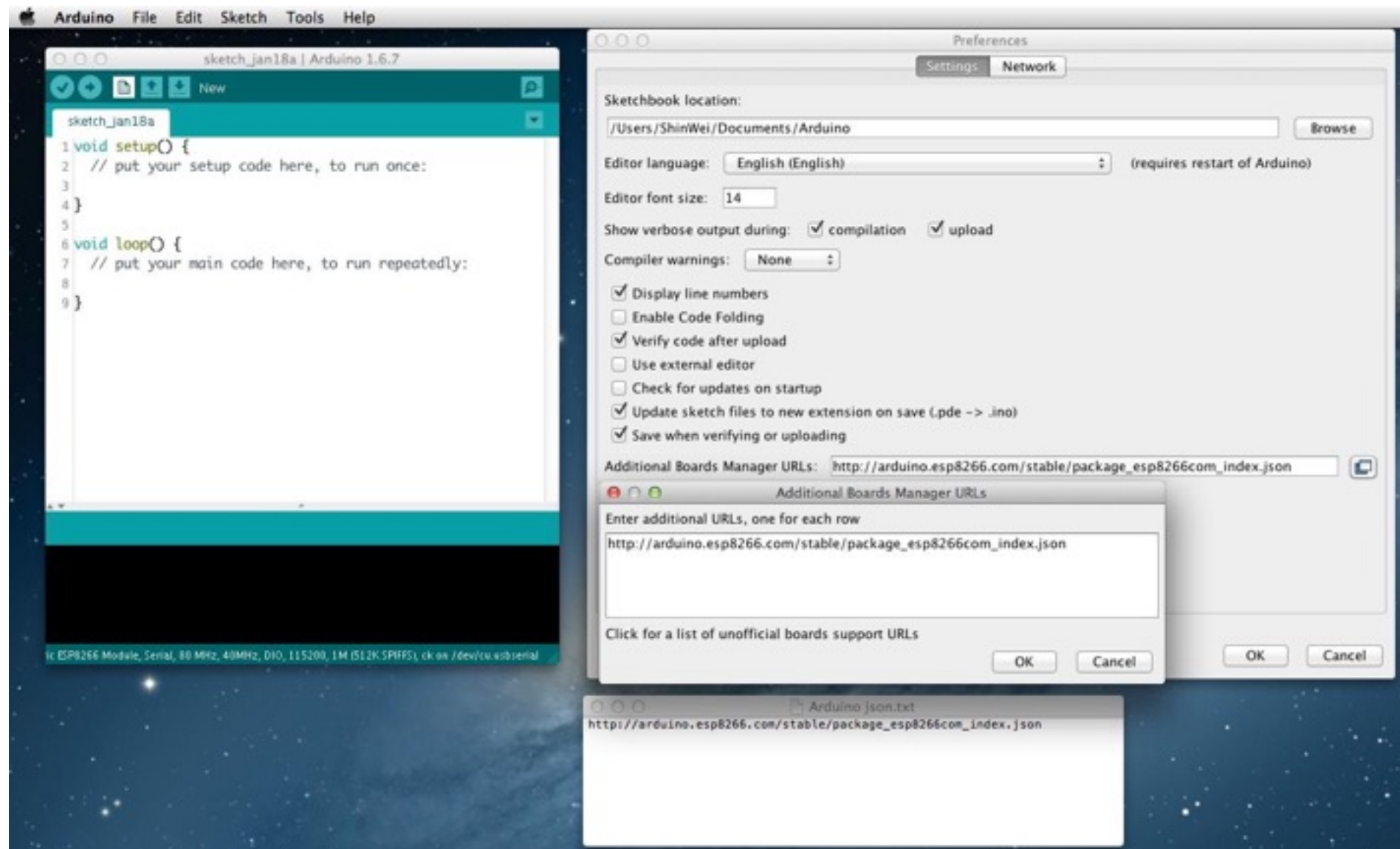




# 設定Arduino-ESP8266

- 額外的板子管理員網址

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

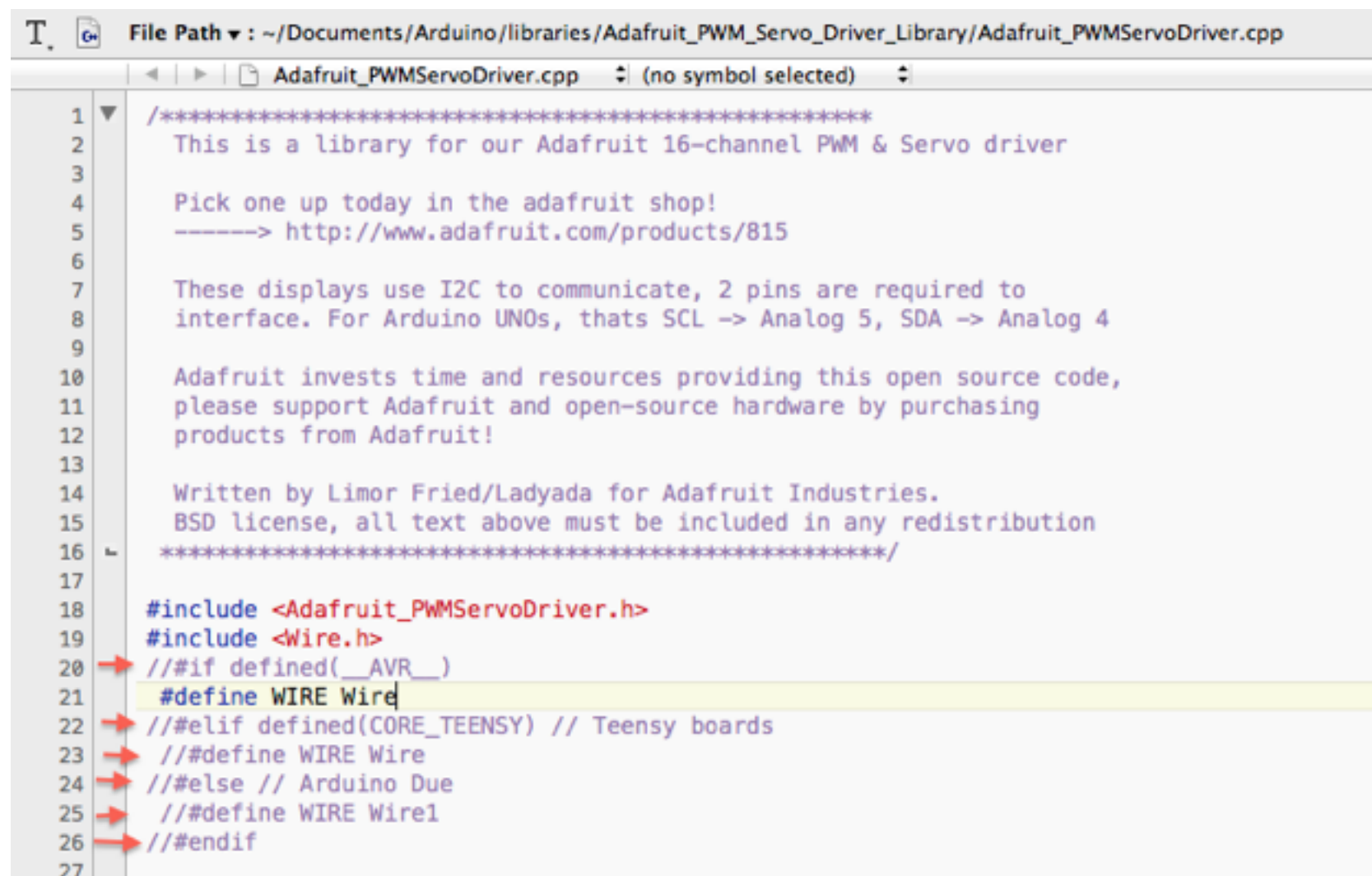


# 設定Arduino-程式庫

- 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：



```
1  /*****
2   This is a library for our Adafruit 16-channel PWM & Servo driver
3
4   Pick one up today in the adafruit shop!
5   -----> http://www.adafruit.com/products/815
6
7   These displays use I2C to communicate, 2 pins are required to
8   interface. For Arduino UNOs, thats SCL -> Analog 5, SDA -> Analog 4
9
10  Adafruit invests time and resources providing this open source code,
11  please support Adafruit and open-source hardware by purchasing
12  products from Adafruit!
13
14  Written by Limor Fried/Ladyada for Adafruit Industries.
15  BSD license, all text above must be included in any redistribution
16  *****/
17
18  #include <Adafruit_PWM_ServoDriver.h>
19  #include <Wire.h>
20  → //#if defined(__AVR__)
21  → #define WIRE Wire
22  → //#elif defined(CORE_TEENSY) // Teensy boards
23  → //#define WIRE Wire
24  → //#else // Arduino Due
25  → //#define WIRE Wire1
26  → //#endif
27
```

# 透過Arduino-IDE與機器人 進行通訊

- 安裝驅動程式

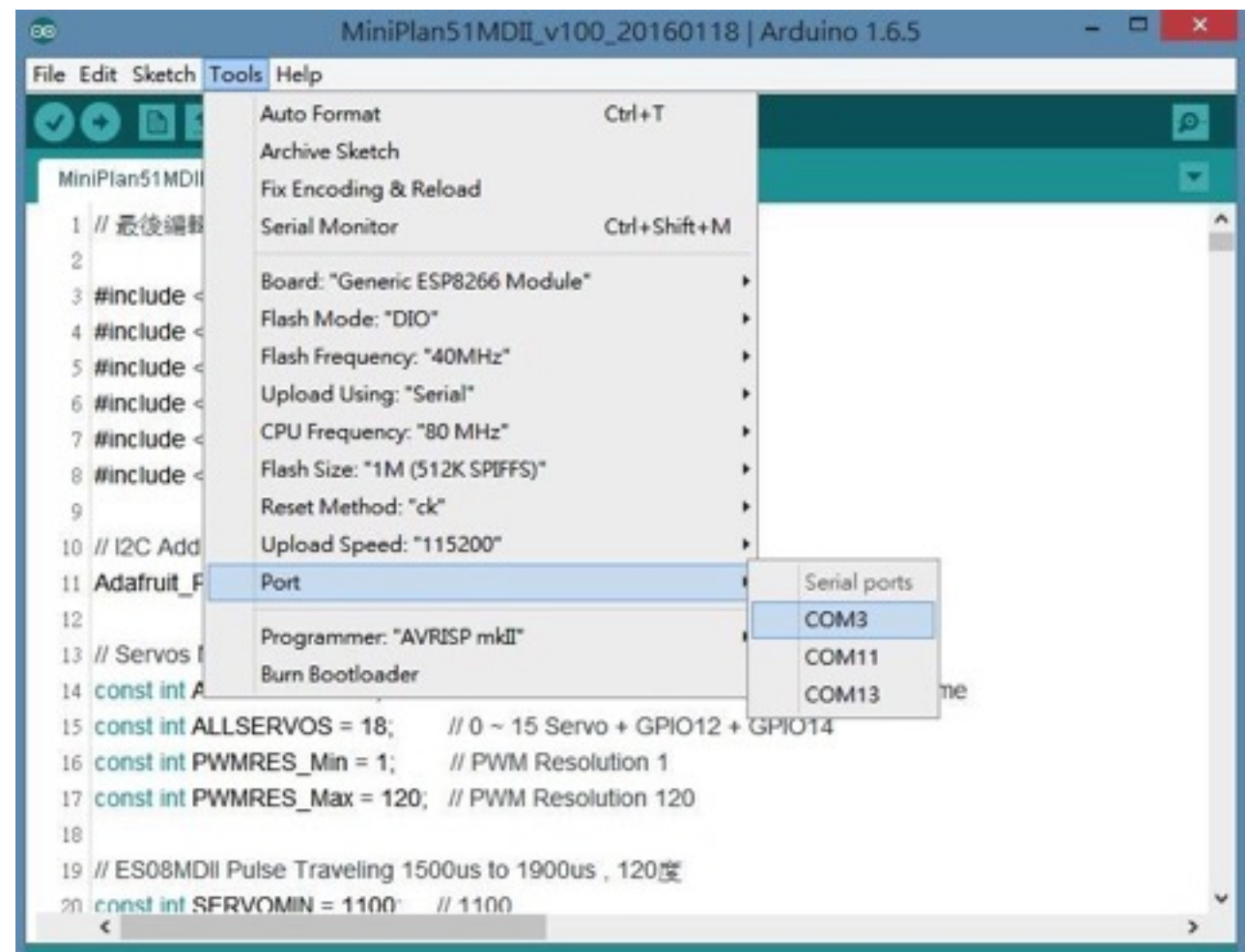
[http://www.prolific.com.tw/US/ShowProduct.aspx?  
p\\_id=225&pcid=41](http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41)





# 透過Arduino-IDE與機器人進行通訊

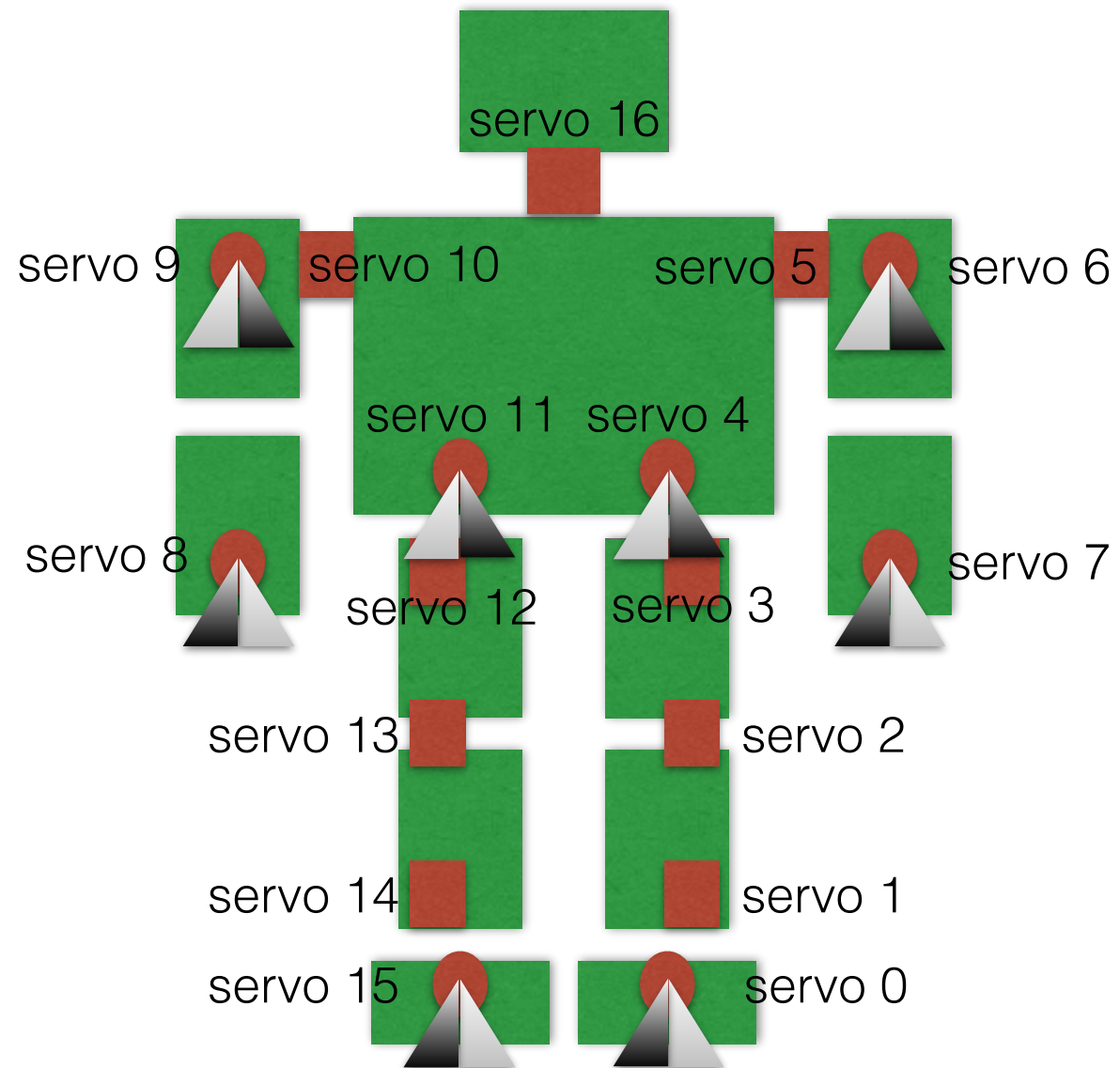
- 設定Arduino-IDE連結方式



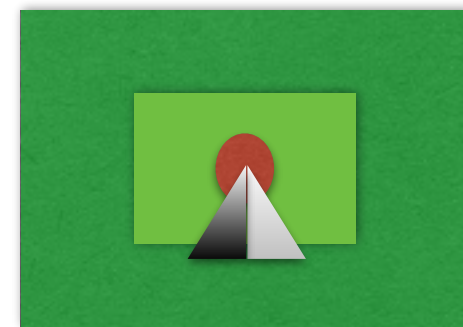
<https://github.com/makee-workshop/Miniplan-workshop>

Makee

# MiniPlan

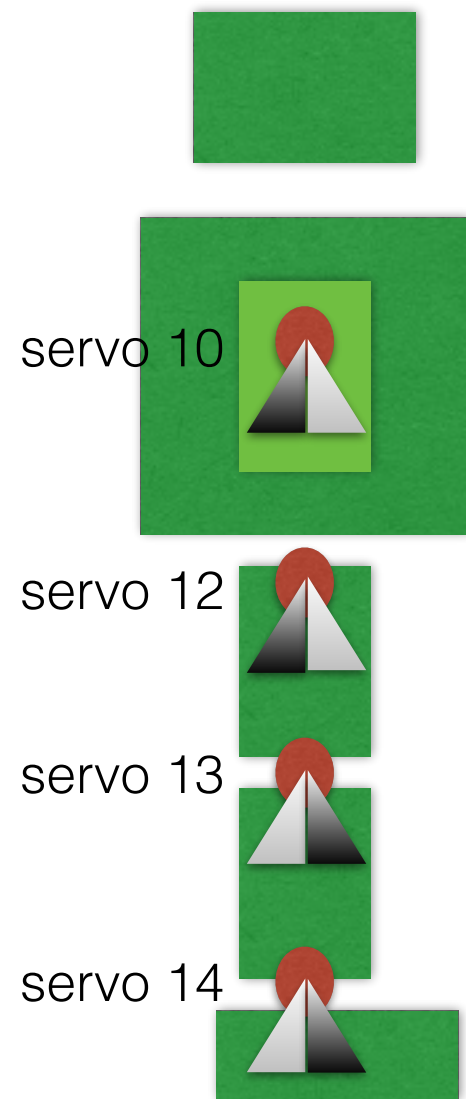


Front

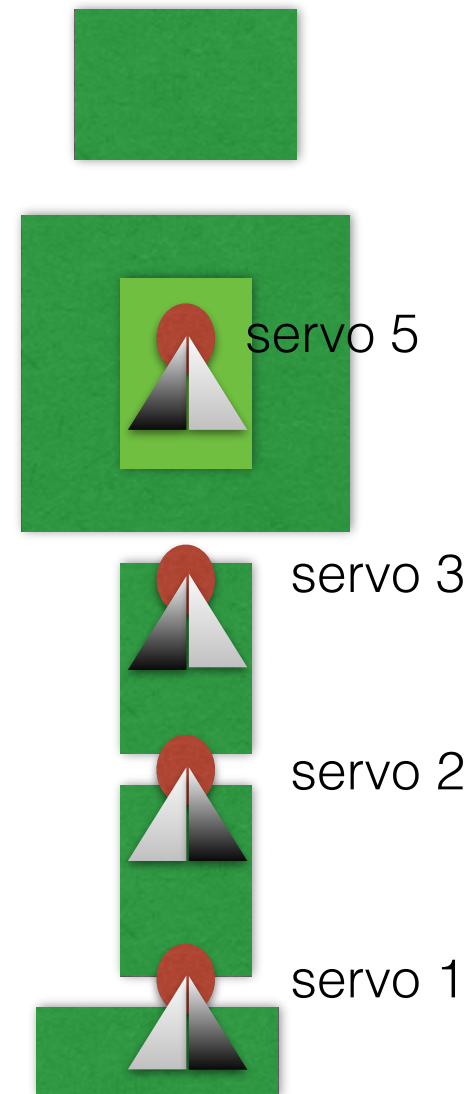


Top

# MiniPlan

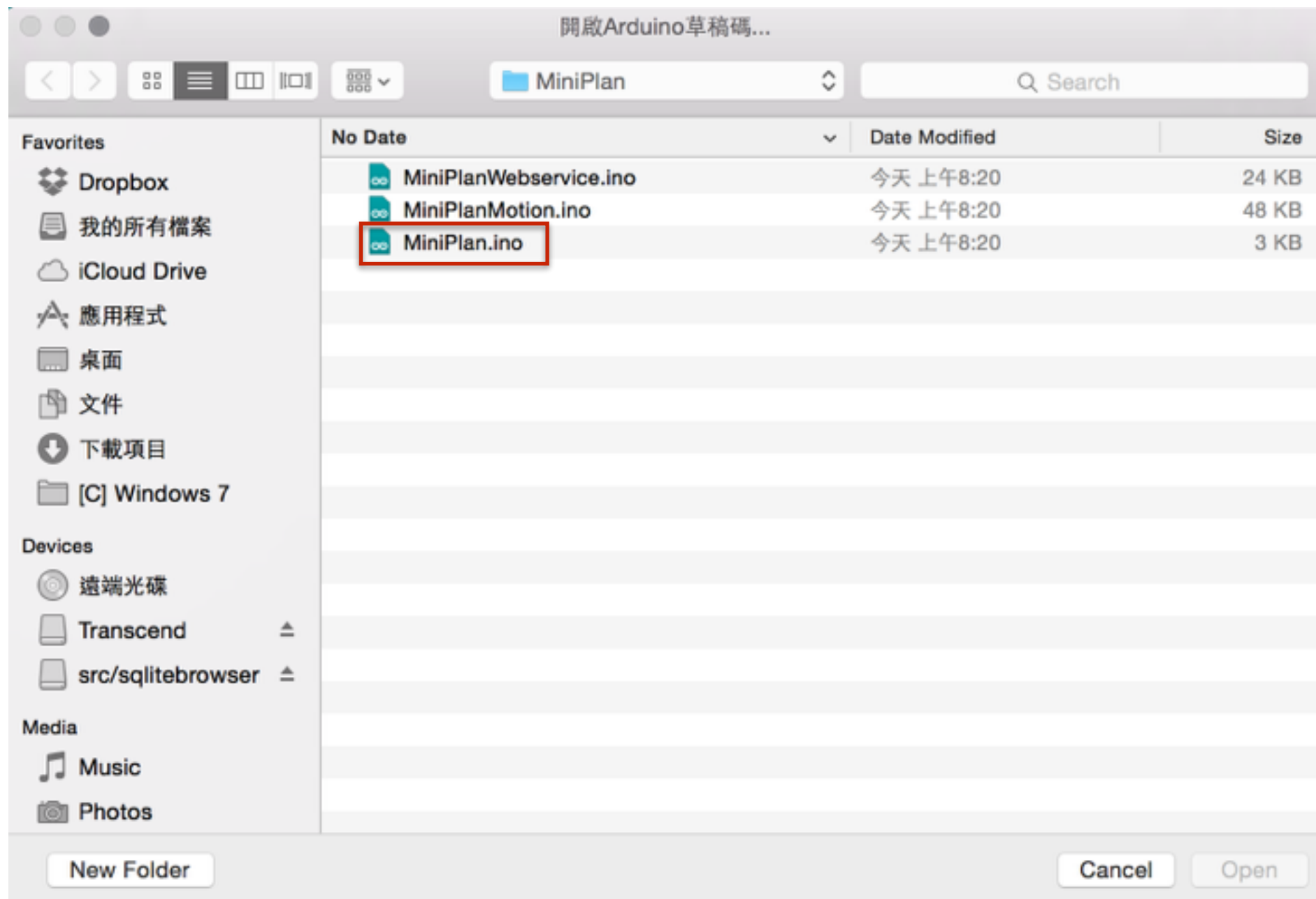


Left



Right

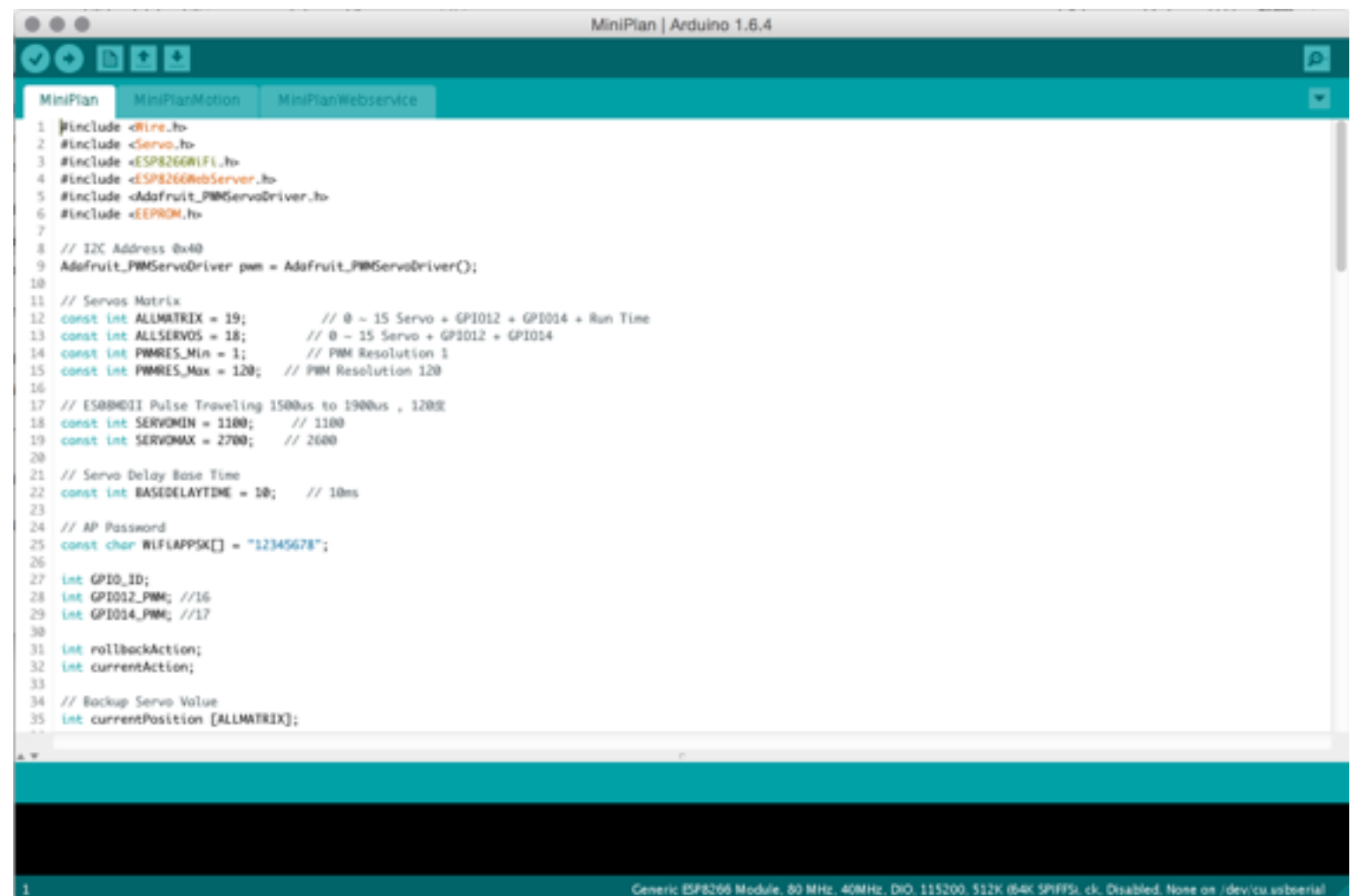
# 開啟MiniPlan專案





# Arduino-主畫面

- MiniPlan
- MiniPlanMotion
- MiniPlanWebservice



The screenshot shows the Arduino IDE interface with the 'MiniPlan' sketch loaded. The sketch is written in C++ and includes several libraries and constants. The code is as follows:

```
1 #include <Wire.h>
2 #include <Servo.h>
3 #include <ESP8266WiFi.h>
4 #include <ESP8266WebServer.h>
5 #include <Adafruit_PWMServoDriver.h>
6 #include <EEPROM.h>
7
8 // I2C Address 0x40
9 Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
10
11 // Servos Matrix
12 const int ALLMATRIX = 19; // 0 ~ 15 Servo + GPIO12 + GPIO14 + Run Time
13 const int ALLSERVOS = 18; // 0 ~ 15 Servo + GPIO12 + GPIO14
14 const int PWMRES_Min = 1; // PWM Resolution 1
15 const int PWMRES_Max = 120; // PWM Resolution 120
16
17 // ES08B011 Pulse Traveling 1500us to 1900us , 1200
18 const int SERVOMIN = 1100; // 1100
19 const int SERVOMAX = 2700; // 2600
20
21 // Servo Delay Base Time
22 const int BASEDELAYTIME = 10; // 10ms
23
24 // AP Password
25 const char WIFLAPPSK[] = "12345678";
26
27 int GPIO_ID;
28 int GPIO12_PWM; //16
29 int GPIO14_PWM; //17
30
31 int rollbackAction;
32 int currentAction;
33
34 // Backup Servo Value
35 int currentPosition [ALLMATRIX];
36
```

The status bar at the bottom indicates the hardware details: "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

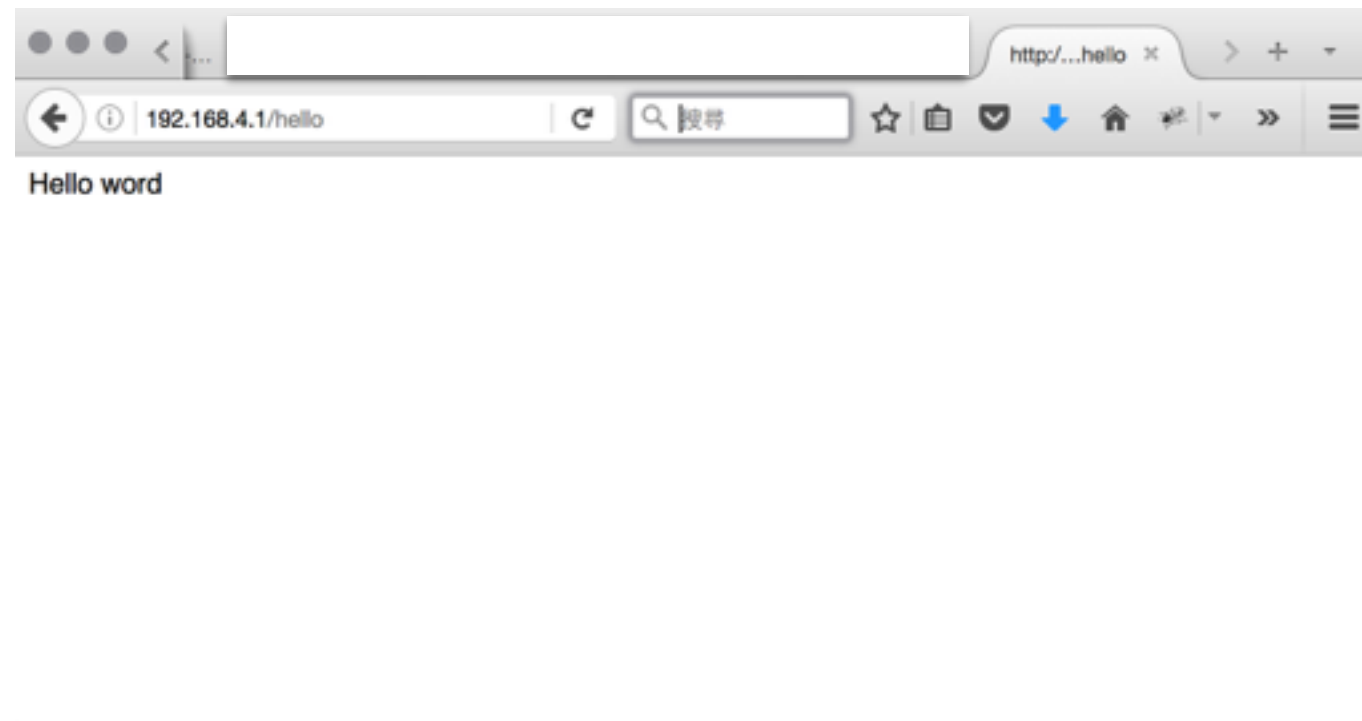
- 在MiniPlanWebservice加入下列的程式碼

```
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, handleEditor);
    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



# 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

