Raspberry Pi

互動科技藝術與感測設計

劉士達

課程大綱

- **1　電子材料清單介紹、認識Raspberry Pi 3 硬體、安裝Raspbian系統、GPIO控制(LED元件、按鈕元件)**
- 2　感測器模組與Raspberry Pi連接(繼電器、溫度、超音波距離感測器)
- 3　攝影機模組、SimpleCV影像處理、MCP3008+可變電阻
- 4　網路連線、伺服器架設、遠距馬達控制
- 5　物聯網、文字轉語音、Arduino應用開發
- 6　蘑菇總動員 - 範例製作

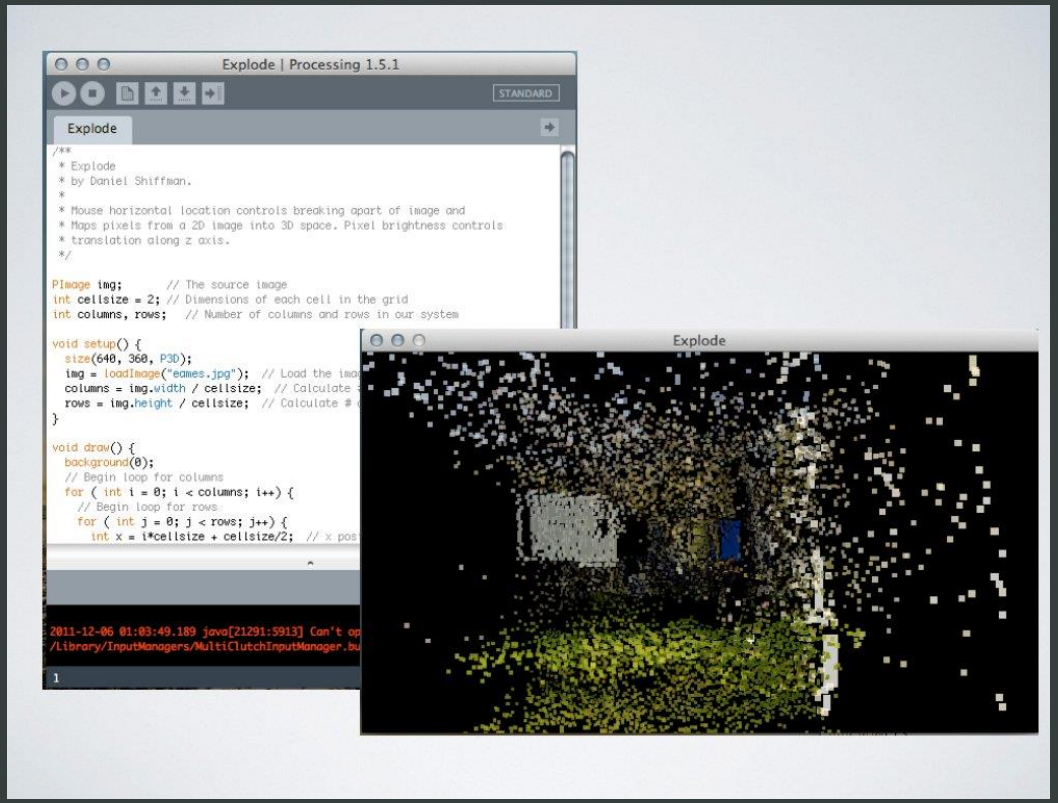
課前環境設定

- 無線AP帳號/密碼：
  - SSID = RaspberryPi_AP
  - 密碼 = raspberry
- 預設的樹梅派帳號/密碼：
  - 帳號：pi
  - 密碼：raspberry
- IP位置查找：
  - Advanced IP Scanner
- 有線網路連接：
  - 拿出2M長的網路線連接至Hub(用此方法比較穩定)

Arduino IDE

- Processing.org
- Wiring.org.co
- Arduino.cc

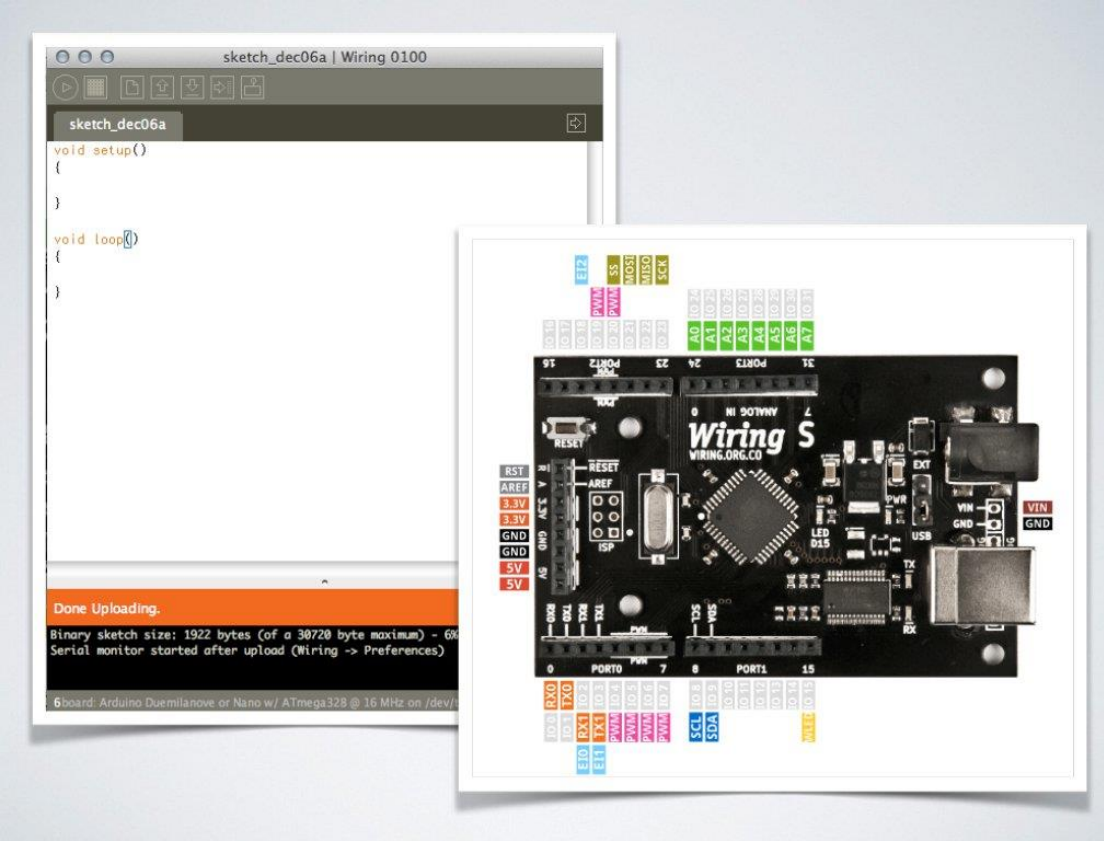
起源於2001年
- 以Java程式語言言為基礎的視覺藝術軟體
- OpenSource
- MIT Media Lab
- Windows、Mac OS、Linux
- Coding is Art



Arduino IDE

- Processing.org
- Wiring.org.co
- Arduino.cc

起源於2003年
- Arduino的前身身
- 首首創OpenSource電路原型開發平台
- 使用用Processing IDE作為程式開發環境
- 公開bootloader、電路設計、編譯器、燒錄器軟體、IDE介面原始碼、眾多感測器連接範例

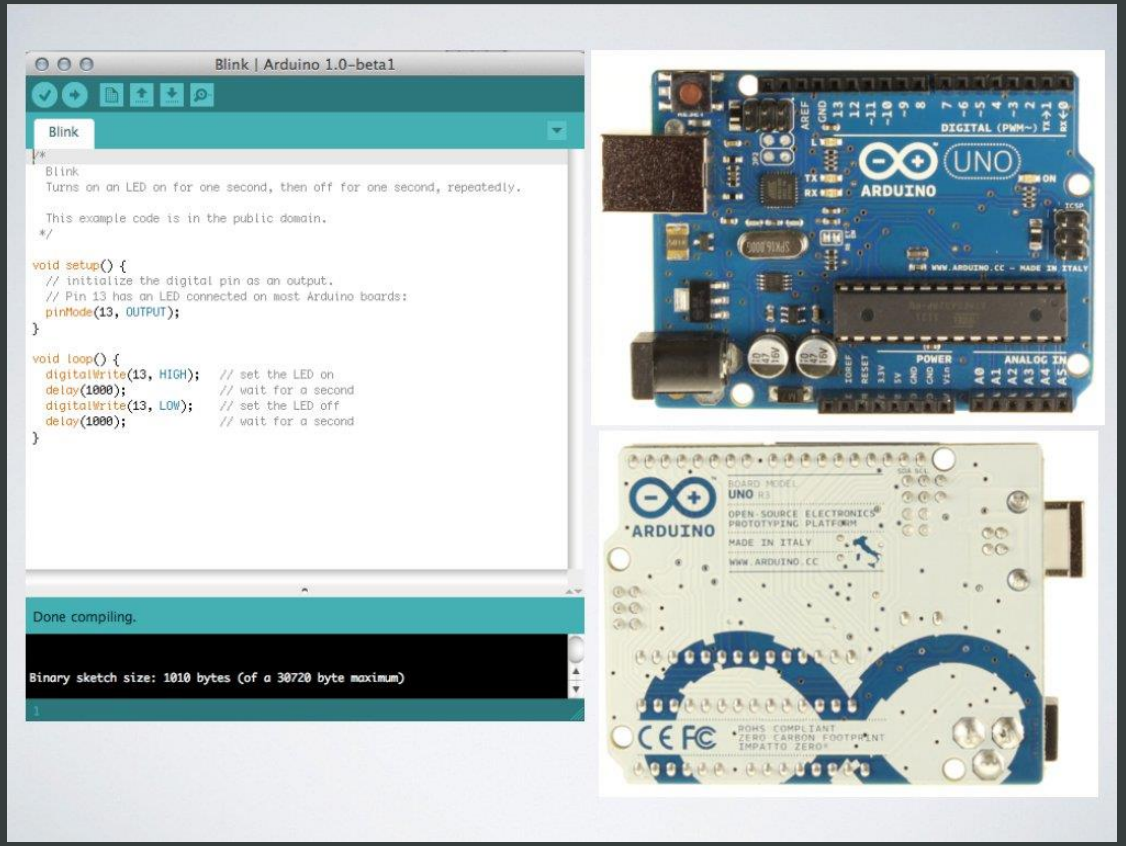


Arduino IDE

- Processing.org
- Wiring.org.co
- Arduino.cc

Arduino IDE

- Processing.org
- Wiring.org.co
- Arduino.cc



創立立於2004年

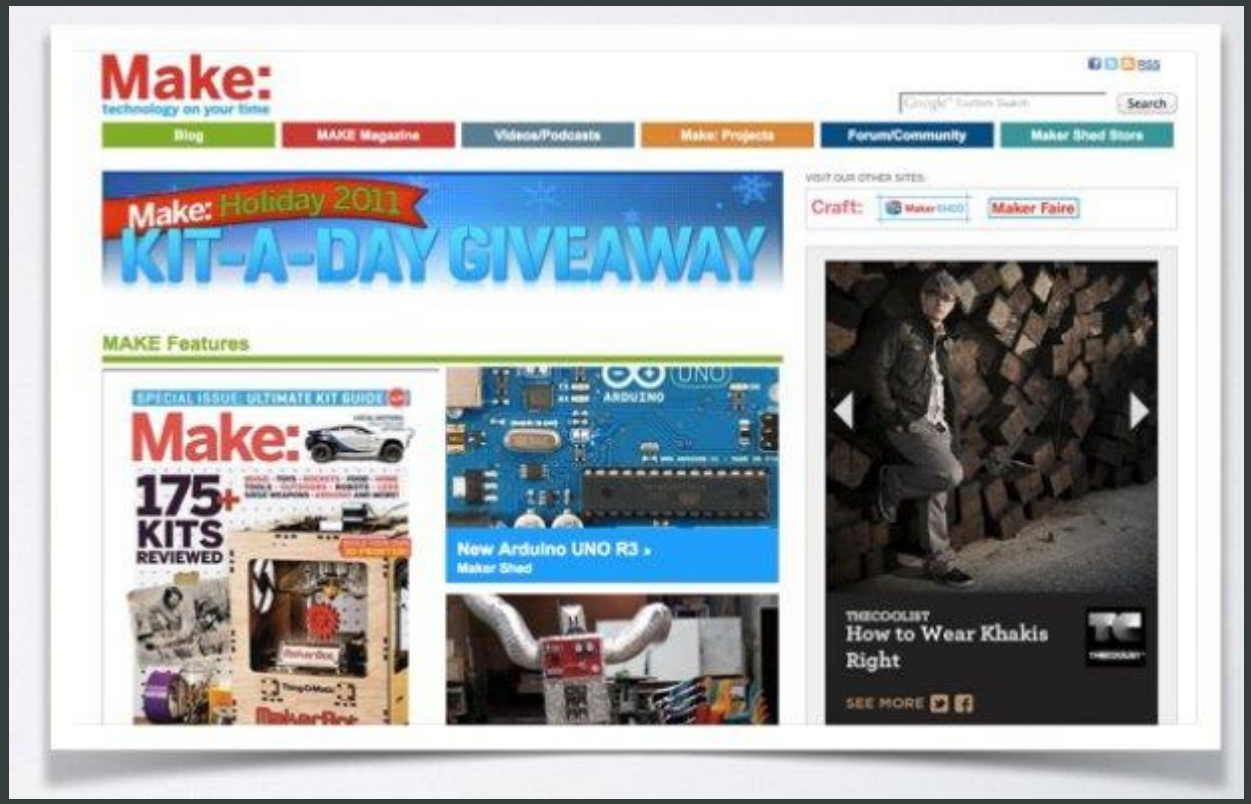
- 結合Processing與Wiring的優點，採用更便宜的架構
- 當時的主要對手為BasicStamp
- 採用用Atmel AVR的相關OpenSource軟體
- 簡單使用用、零件成本便宜、上手速度快、討論區完整、容易複製
- 眾多的應用用範例與Libraries

Arduino社群的形成

- AVR Freaks 大大量的DIY(Maker) 玩家討論AVR
- Avrdude , avrgcc, avr-g++, Java 開源軟體為基礎
- Arduino 軟硬體完全開放，不留任何機密
- MakeMagaize, Instructables.com DIY教學網站大量分享應用用範例，破解心得
- Sparkfun.com, seeedstudio.com, adafruit.com 提供完整的DIY套件、零件、電路板、模組、擴充板，甚至至提供範例程式碼

MAKE:MAGAIZE

- DIY月刊/線上討論/教學/影片片
- 每期都有電子DIY的專題報導
- 專案製作教學與材料販賣相連
- 提供高畫質教學影片
- 熱絡的DIY討論區
- 定期舉辦工工作坊
- 年度DIY創意競賽

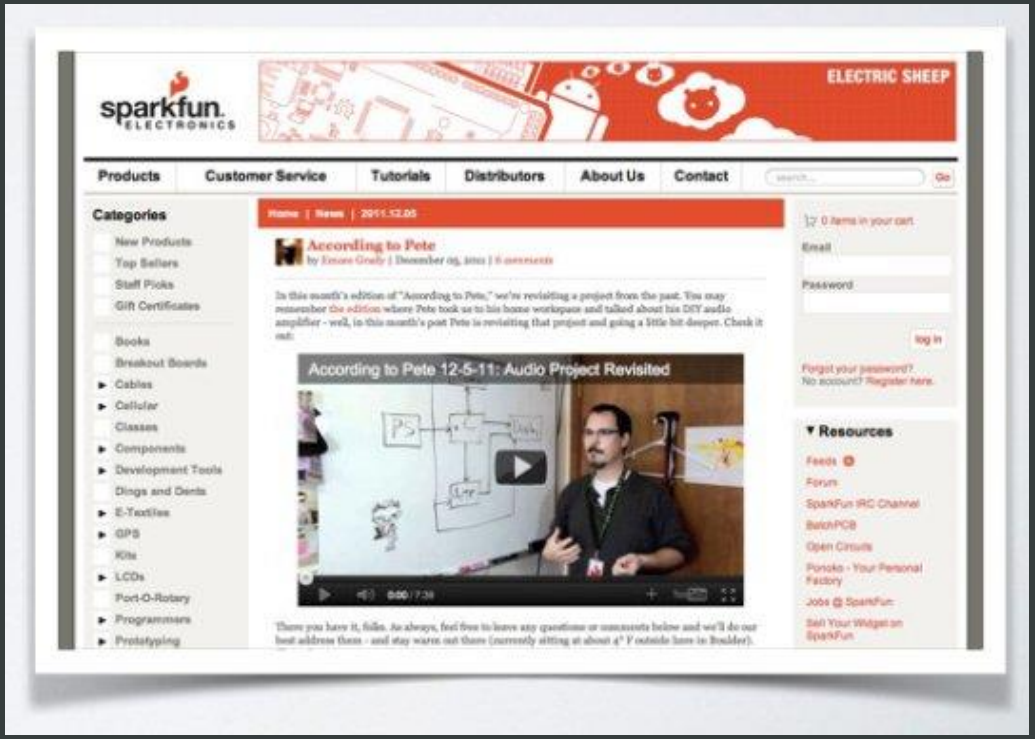


INSTRUCTABLES.COM

- DIY製作分享平台
- 會員可下載完整製作過程與材料清單[pdf]
- DIY範圍廣：美食、生活、戶外、科技、玩樂
- 適合樂於分享創意的作者
- 上萬篇DIY創意
- 分類清楚
- 入會兩年只需$40美元

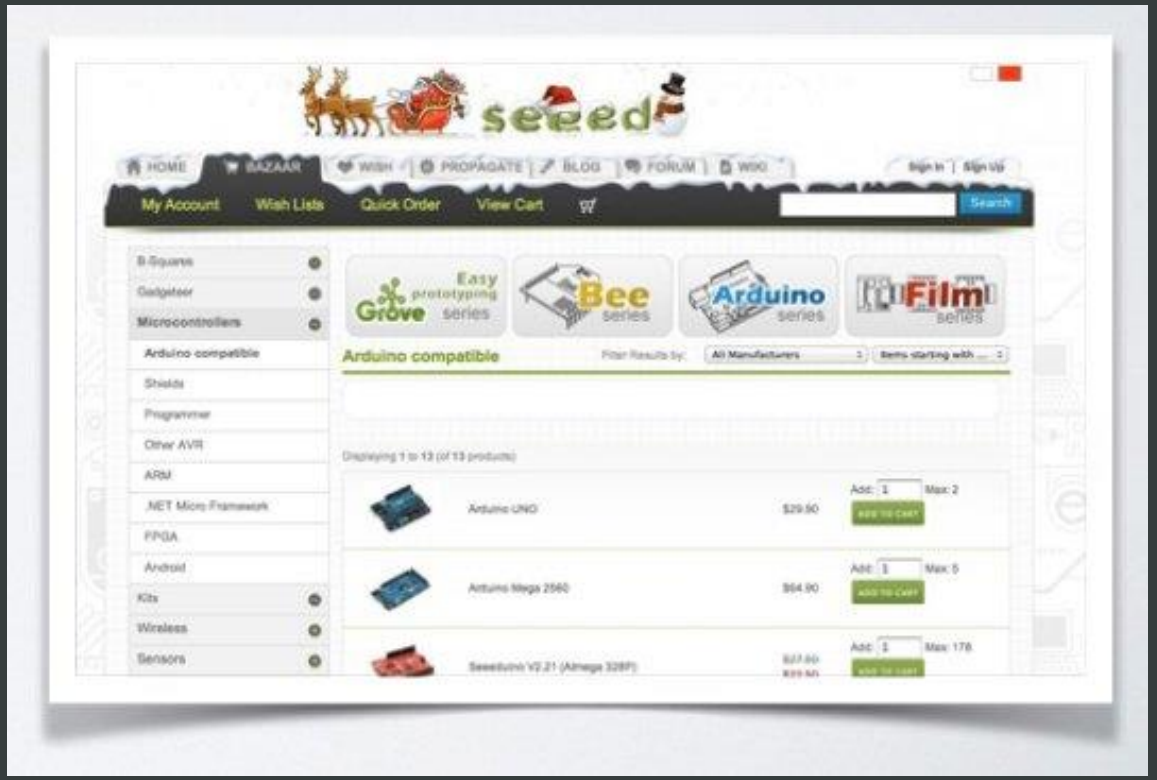
SPARKFUN.COM

- 最完整的DIY套件、模組、開發板、零件販賣商
- 套件、模組都有相關教學、範例程式碼
- 自行研發DIY模組與PCB板設計
- 每項產品都有豐富的討論串
- 發貨速度快，配合國際貨運最快2天到貨
- 即時上架最新的開發模組
- 可販賣自行研發的套件



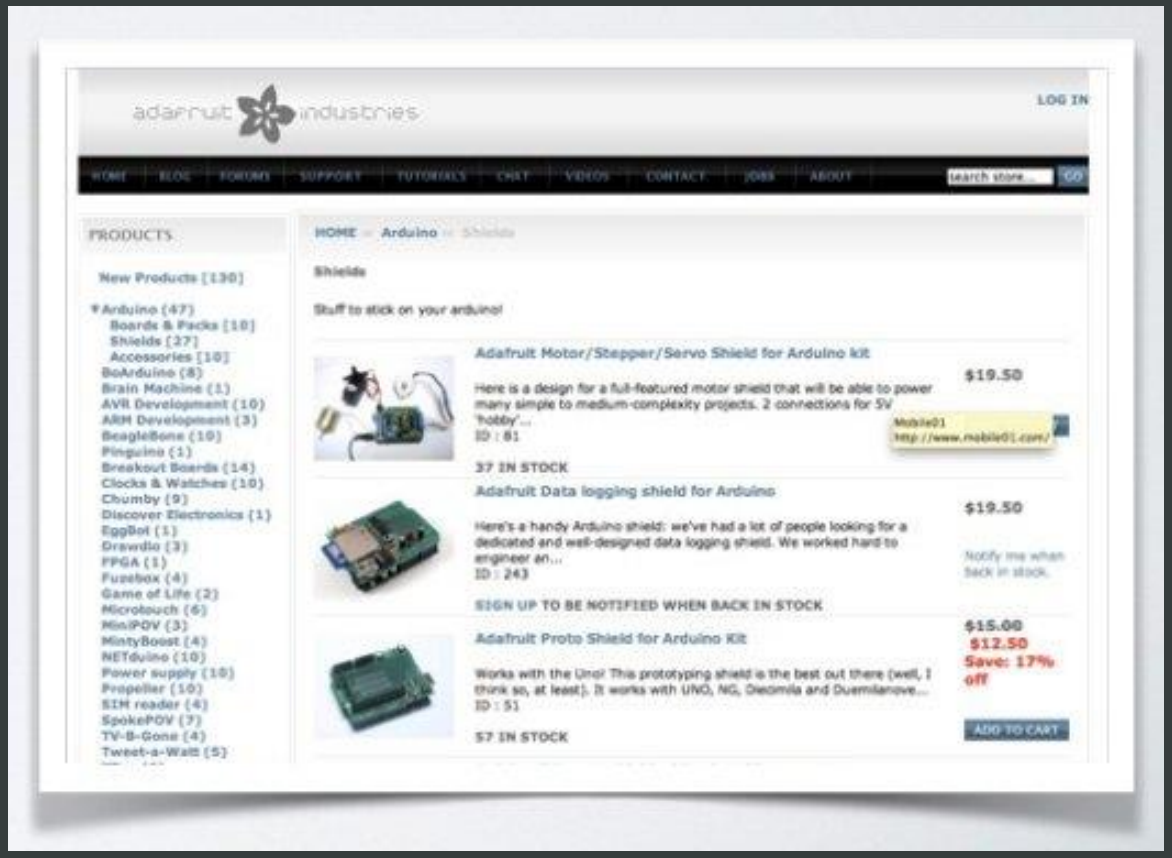
SEEEDSTUDIO.COM

- Hacker最愛，最多的特殊模組套件
- Arduino延伸版本匯集地
- 眾多的擴充板套件
- 自行研發的PCB板設計
- 機器人套件較多
- 可代為設計電路原型



ADAFRUIT.COM

- 以OpenSource Hardware為主的套件與元件銷售
- 自行設計Open Source Hardware擴充板
- 豐富的教學與影片
- 價格最為合理便宜
- 工作坊教學套件居多



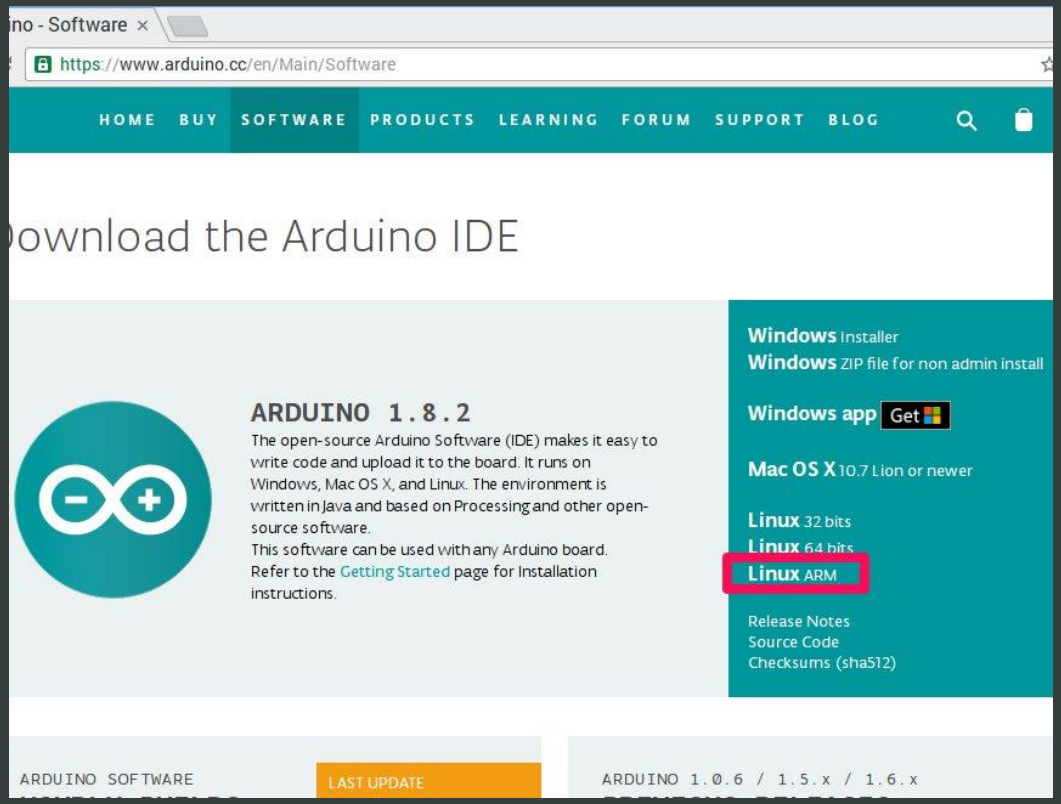
Taobao.com 淘宝

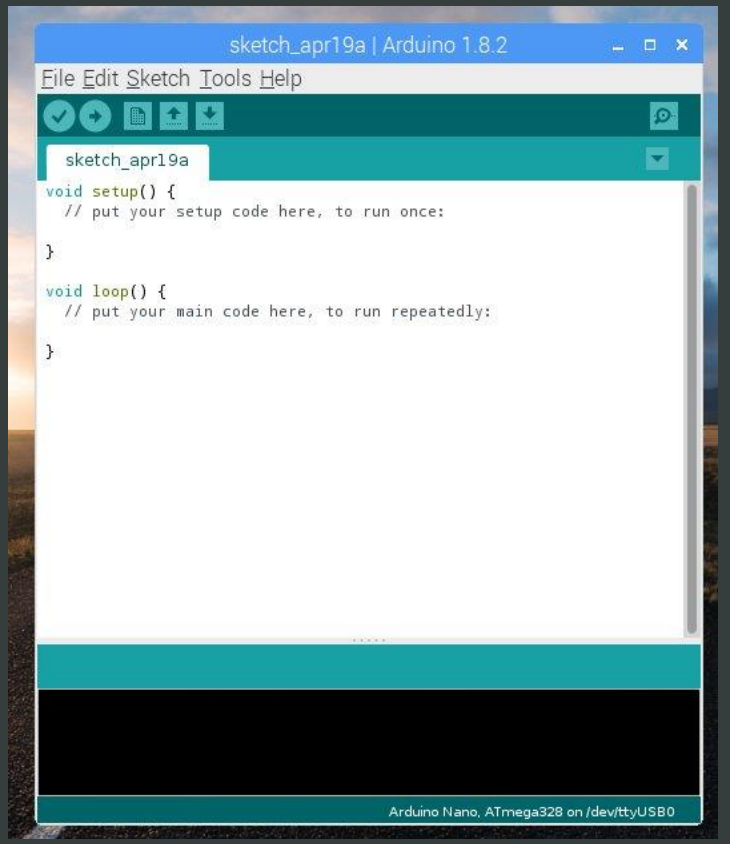
- 世界工廠
- 超低價格
- 最多的感測器款式與種類
- 資訊清楚透明
- 但有買到假貨的可能
- 可靠性不足

安裝1.8.x版Arduino IDE

Step1 – 下載Arduino

打開瀏覽器下載Arduino Linux ARM版本





解壓縮與安裝Arduino 1.8.x

Step2 – 解壓縮Arduino

打開終端機輸入指令

```
$ cd ~/Downloads
```

Step3 – 解壓縮Arduino

```
$ tar –xvf arduino-1.8.8-linuxarm.tar.xz
```

Step4 – 修改Arduino安裝指令檔

```
$ sudo vim ~/Downloads/arduino-1.8.8/install.sh
```

安裝執行腳本

Step5 – 安裝Arduino資料的快捷路徑

```
$ sudo ~/Downloads/arduino-1.8.8/install.sh
```

選擇Arduino的comport

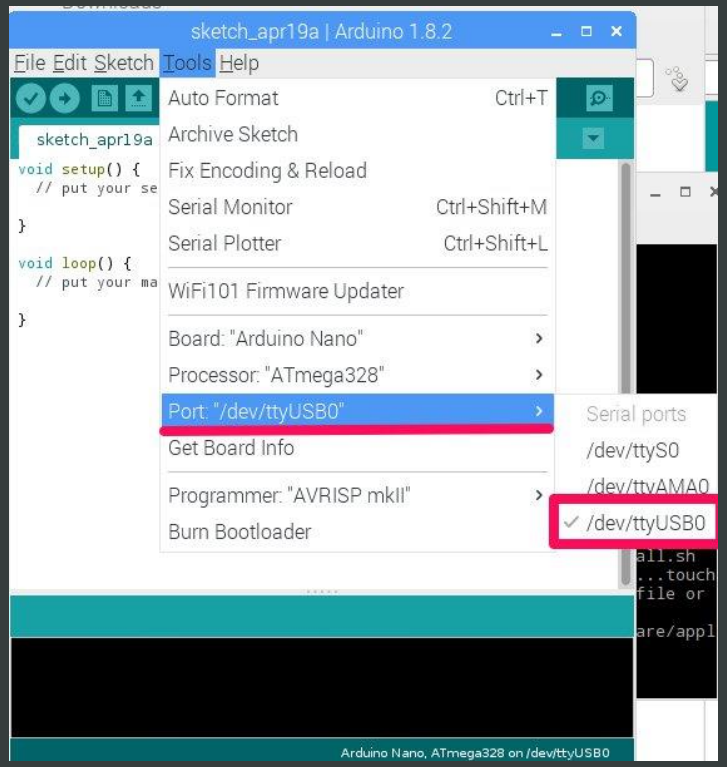
Port “/dev/ttyUSB0”

Port “/dev/ttyACM0”

/dev/ttyS0 是Pi內建的RXTX

/dev/ttyAMA0 是Pi內建的藍芽RXTX

/dev/ttyUSB0 才是USB連接Arduino的Port



Arduino各項資料夾

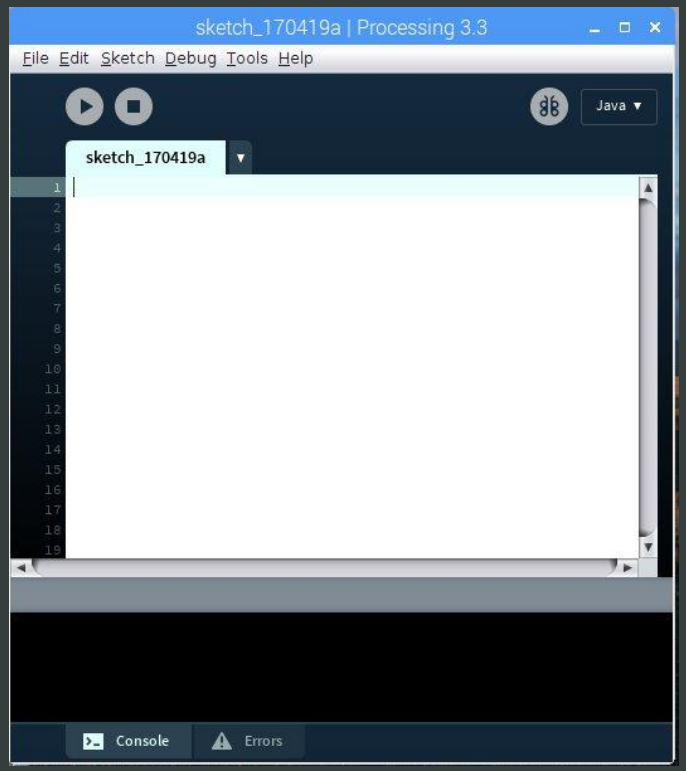
- Libraries資料夾
  - /home/pi/Arduino/libraries
- Sketch存放位置
  - /home/pi/Arduino
- Arduino-1.8.x資料夾位置
  - /home/pi/Downloads/arduino-1.8.x

安裝Processing 3.3版

Step1 – 下載並安裝Processing

```
$ curl https://processing.org/download/install-arm.sh | sudo sh
```

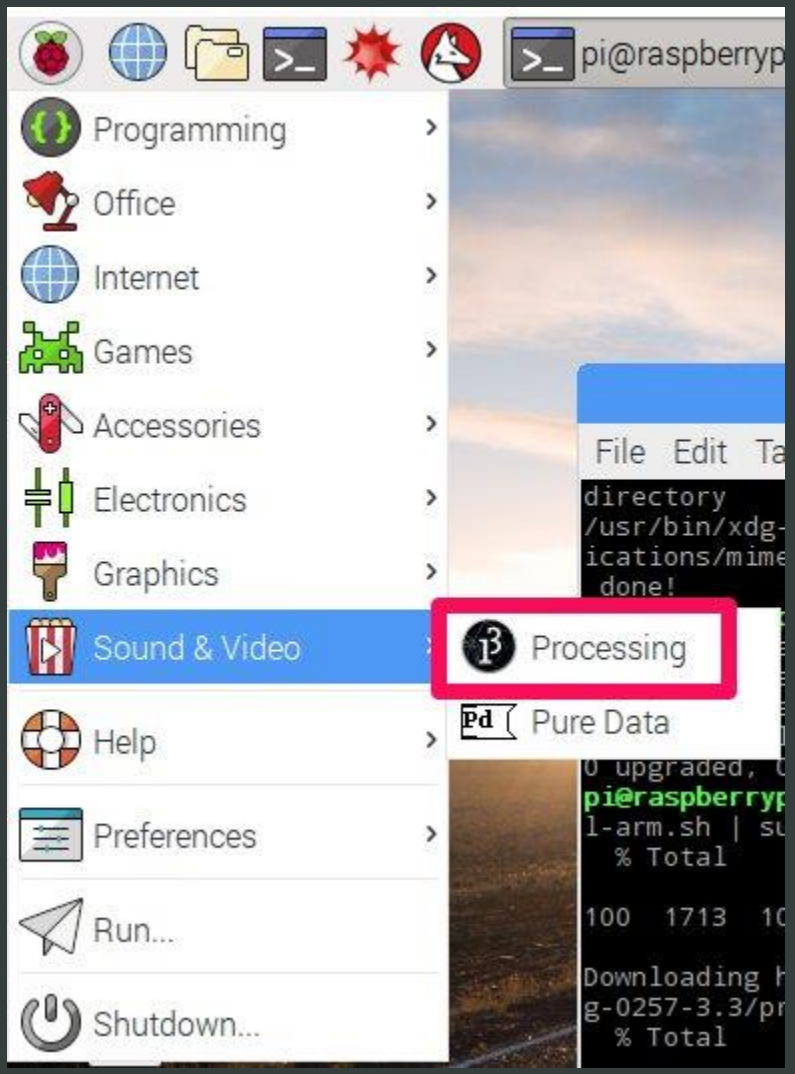
```
pi@raspberrypi_testbed:~/Downloads $ curl https://processing.org/
l-arm.sh | sudo sh
  % Total    % Received % Xferd  Average Speed   Time    Time
                                 Dload  Upload   Total   Spent
100  1713  100  1713    0     0    578      0  0:00:02  0:00:02 -

Downloading https://github.com/processing/processing/releases/dow
g-0257-3.3/processing-3.3-linux-armv6hf.tgz...
  % Total    % Received % Xferd  Average Speed   Time    Time
                                 Dload  Upload   Total   Spent
100   608    0   608    0     0    649      0 --:--:-- --:--:-- -
 58 90.6M   58 52.9M    0     0  1631k      0  0:00:56  0:00:33
```



開啟Processing 3.x版

Step2 – 開啟Processing

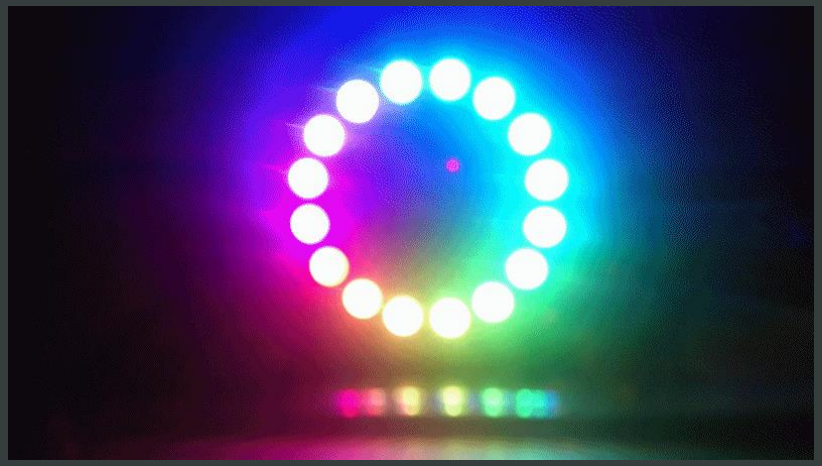
開始->Sound & Video->Processing

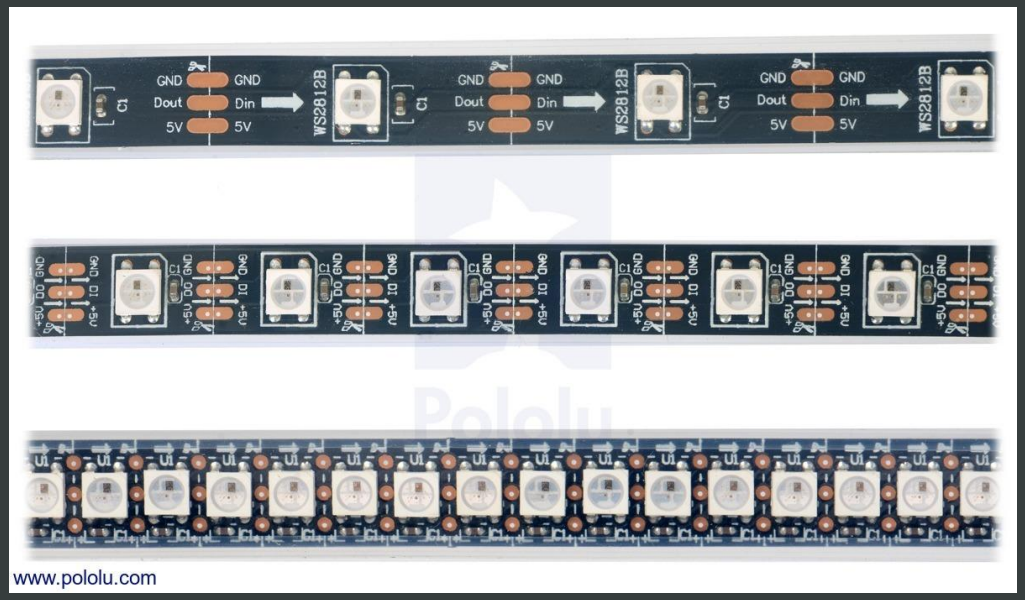


Processing各項資料夾

- Libraries資料夾
  - /home/pi/sketchbook/libraries
- Sketch存放位置
  - /home/pi/sketchbook
- Processing 3.3 執行檔資料夾位置
  - /usr/local/bin/
- Processing 3.3 程式庫資料夾位置
  - /usr/local/lib/processing-3.3

WS2812B 全彩LED環





用Arduino 驅動WS2812

Step1 – 下載FastSPI 程式庫

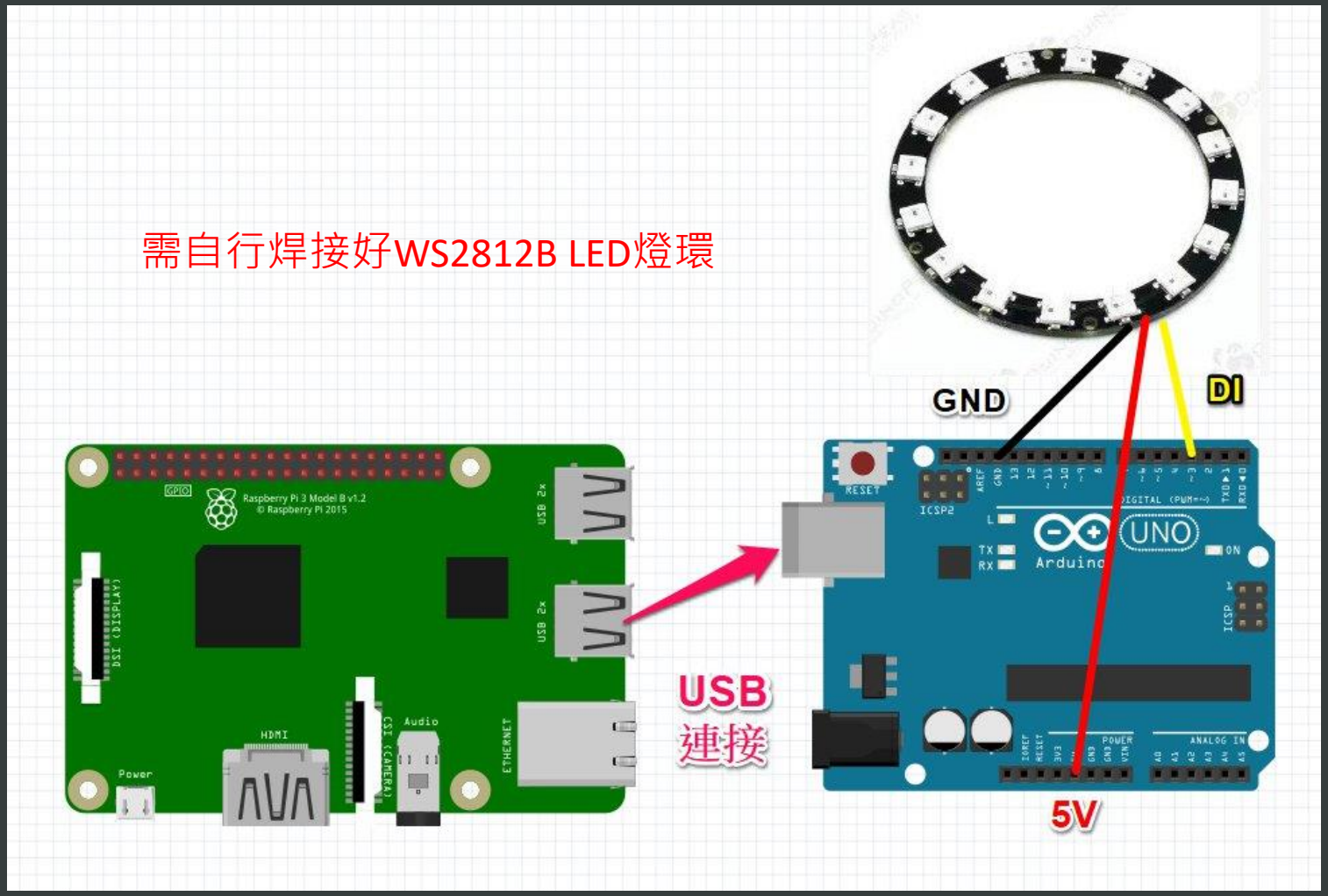
```
$ git clone https://github.com/FastLED/FaseLED.git
```

Step2 – 移動FastSPI 程式庫到Arduino Libraries

```
$ mv FastLED/ ~/Arduino/libraries
```

Step3 – 重新打開Arduino 1.8.8

接線圖


需自行焊接好WS2812B LED燈環
GND
DI
USB
連接
5V
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
GPIO
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
DIGITAL (PWM~)
ANALOG IN
POWER

範例程式

- 開啟Examples->FastLED->DemoReel100

FASTSPI程式庫使用方式

- #define DATA_PIN 3 //宣告要輸出DATA的Arduino腳位
- #define LED_TYPE WS2812B //設定LED的型號，原本預設是WS2811這邊需改成WS2812B
- #define NUM_LEDS 16 //LED的數量，預設64點，修改成16點

- 開啟範例：
  - Examples->FastLED->Blink
- 設置不同LED型號：
  - FastLED.addLeds<WS2812B, DATA_PIN, RGB>(leds, NUM_LEDS); //指定要驅動的LED型號
- 點亮LED燈點：
  - leds[0] = CRGB::Red;
  - FastLED.show();

課堂討論與問題解決1

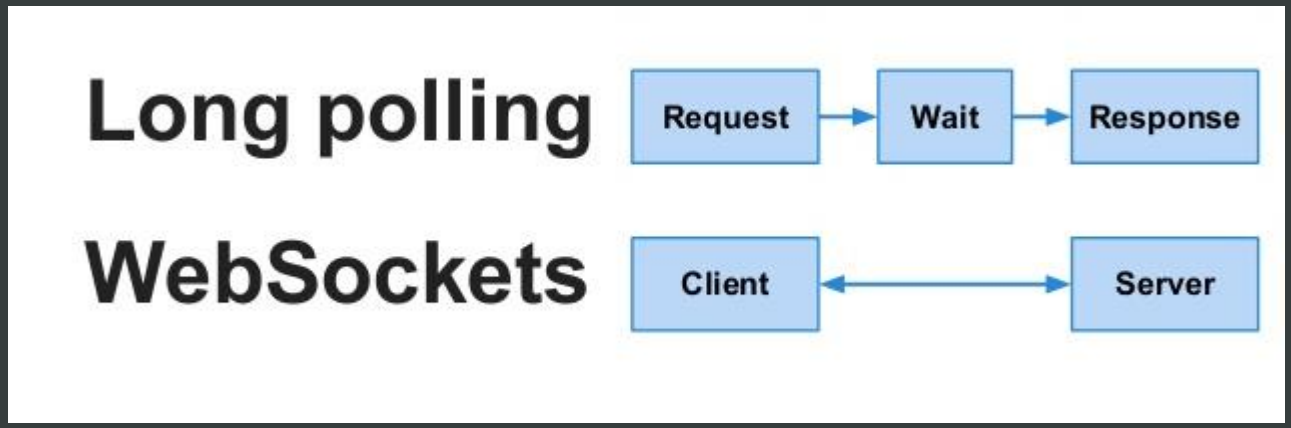
- 請同學試試看如何用Python語言方式，透過Raspberry Pi控制Arduino上面的WS2812B LED燈環??

WebSocket

WebSocket

- WebSocket = 網頁版的Socket Server，透過客戶端瀏覽器直接與Server進行通訊，比傳統的瀏覽網頁(Polling輪詢)方式更加即時。

Gmail / google Calender /google 文件 …等
http://demo.kaazing.com/forex/
https://thingspeak.com/
http://www.blynk.cc/



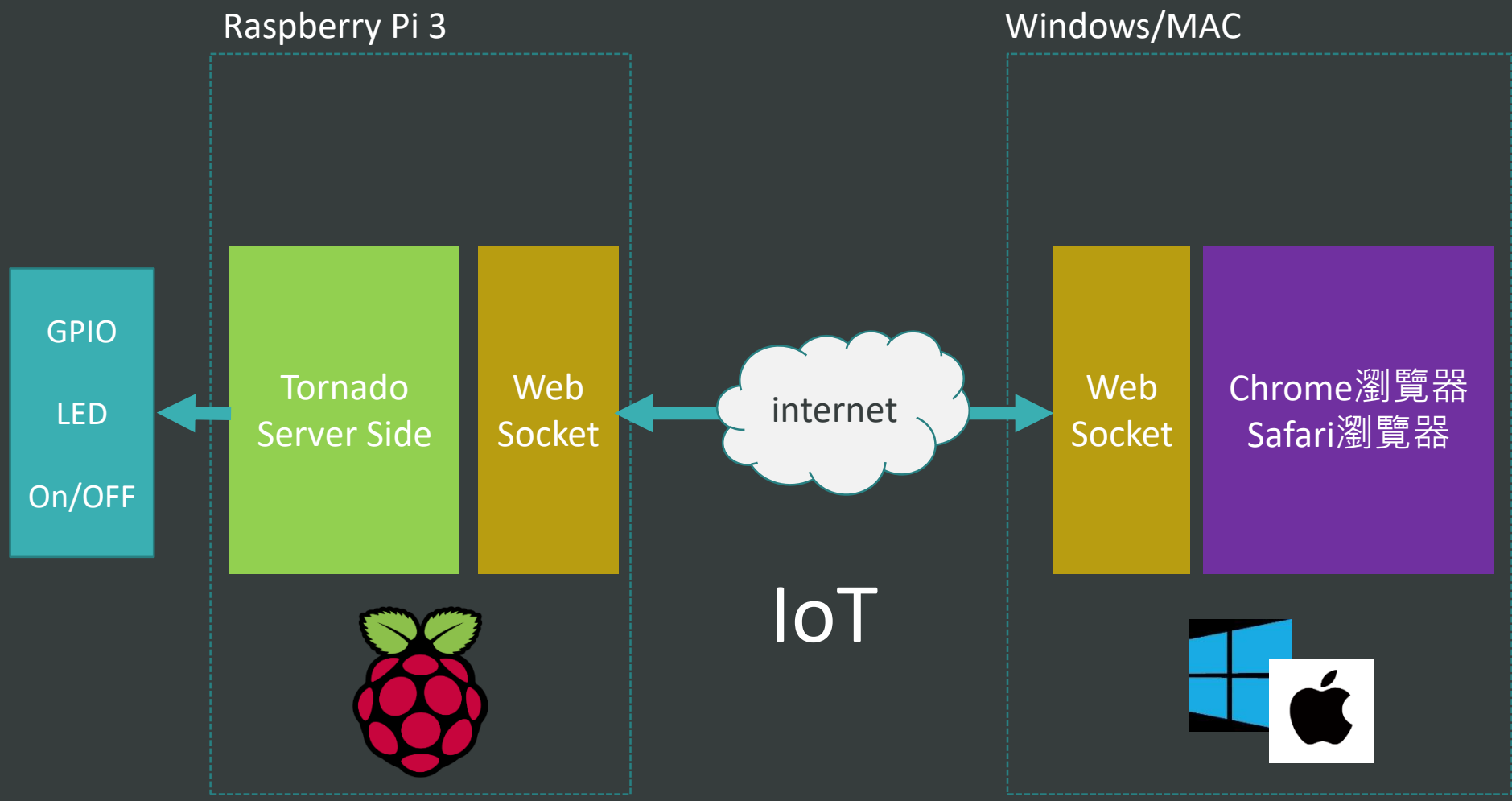
WebSocket Frameworks

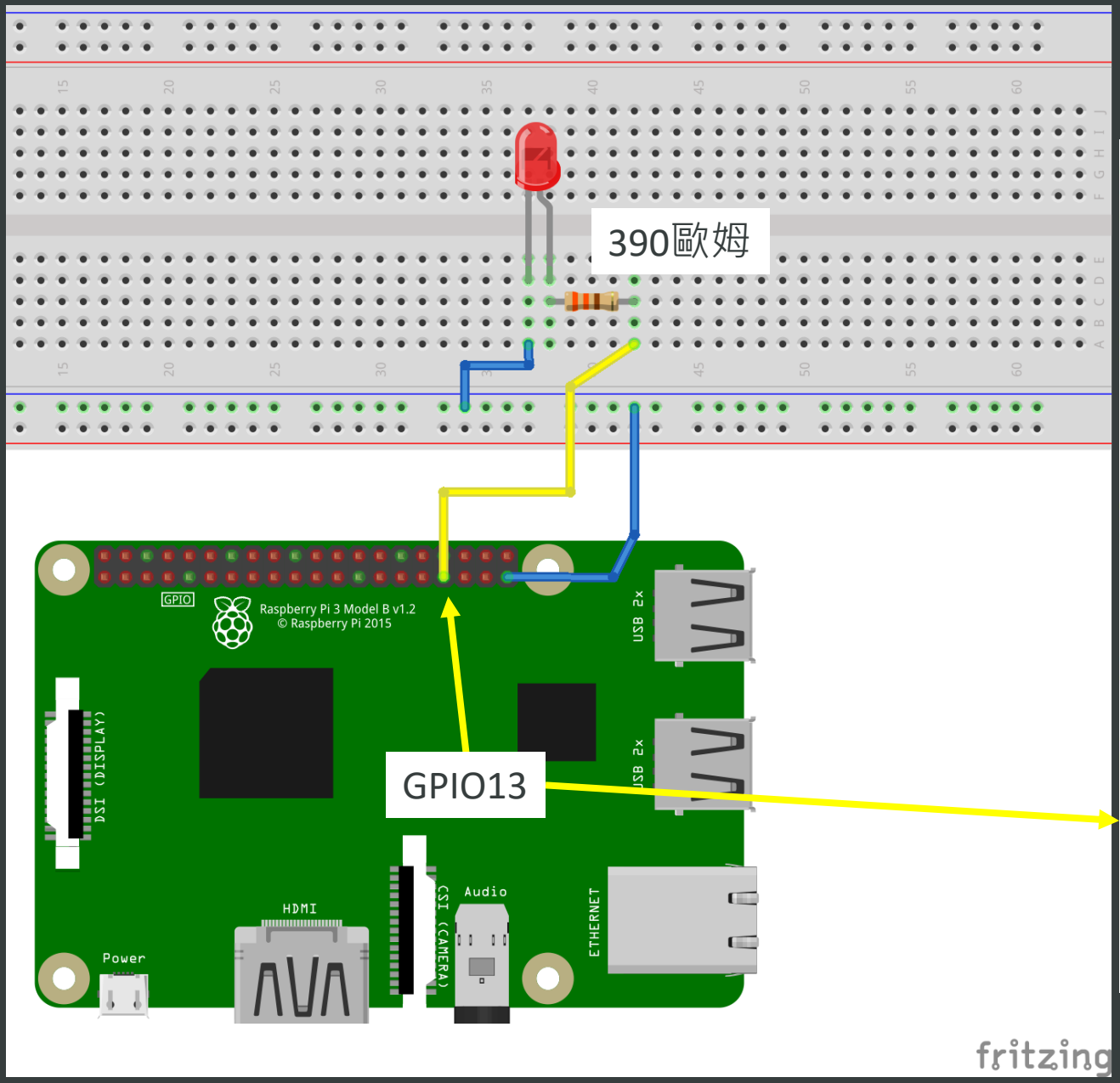
- Node.js
- Socket.IO
- WebSocket-Node
- Jetty
- Ruby
- Pywebsocket
- Tornado
- Libwebsockets
- SuperWebSocket
- WS
- Engine.io
- sockJS
- Primus
- Fleck
- Atmosphere
- Java Web Socket
- Mojolicious
- SignalR
- Plezy
- Deepstream.io
- Meteor
- Kuzzle.io

Tornado Web Frameworks

- 特色
  - 速度快
  - 支援非阻塞式伺服器
  - 支援WebSocket連接
  - 支援Web應用服務框架
  - 支援HTTP Server / AsyncHTTP Client
  - 支援IOLoop / IOStream 異步網路功能程式庫
  - 使用python程式語言
- 安裝方式
  - `$ sudo pip3 install tornado`

利用Tornado + WebSocket + LED GPIO控制





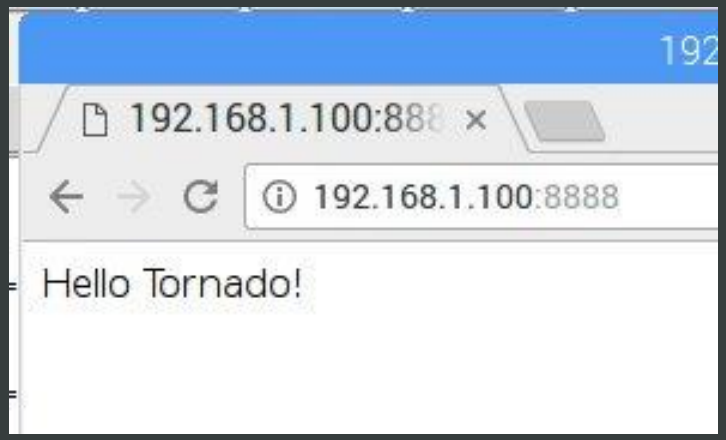
	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

建構Raspberry端的後台程式

```
import tornado.ioloop
import tornado.web

class MainApp (tornado.web.RequestHandler):
    def get(self):
        self.write("Hello Tornado!")

app = tornado.web.Application([
    (r"/", MainApp),
    ])

app.listen(8888)
tornado.ioloop.IOLoop.instance().start()
```

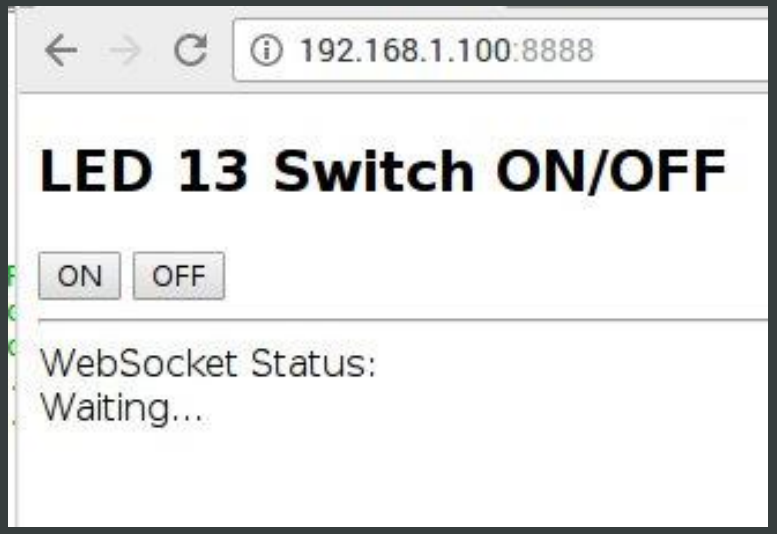
```
#匯入ioloop
#匯入web

#定義get函式處理http的GET請求

#建立APP物件,調用request根目錄下的
MainApp Class

#讓APP監聽在8888 port
#實例化
```

simpleWebServer.py



修改輸出的文本為html內容

```
import tornado.ioloop
import tornado.web

html_text = '''
<!DOCTYPE html>
<html>
 <body>
  <h2>LED 13 Switch ON/OFF</h2>
   <input type='button' id='led_on' value='ON'>
   <input type='button' id='led_off' value='OFF'>
   <script src="//192.168.3.4/jquery.min.js"></script>
   <script src="//192.168.3.4/ws.js"></script>
   <hr>
   WebSocket Status:
   <br>
   <div id="ws-status">Waiting...</div>
 </body>
</html>
'''

class MainApp (tornado.web.RequestHandler):
    def get(self):
        self.write(html_text)

app = tornado.web.Application([
    (r"/", MainApp),
    ])

app.listen(8888)
tornado.ioloop.IOLoop.instance().start()
```



simpleWebServer_html.py

*記得修改ip為自己的pi的ip http://192.168.X.X

ws.js

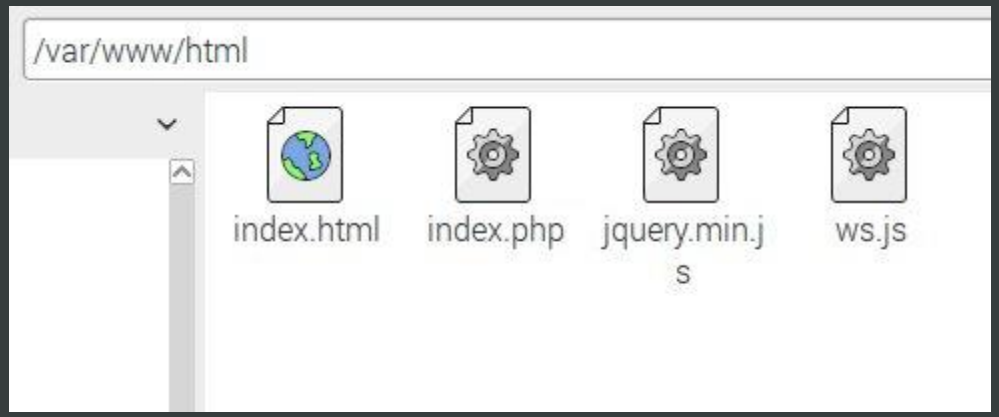
```
$(document).ready(function(){
    var WEBSOCKET_ROUTE = "/ws";
    if(window.location.protocol=="http:"){
        var ws = new WebSocket("ws://"+window.location.host+WEBSOCKET_ROUTE);
    }

    ws.onopen = function(evt){
        $("#ws-status").html("Connected");
    };

    ws.onclose = function(evt){
        $("#ws-status").html("Disconnected");
    };

    ws.onmessage = function(evt){};

    $("#led_on").click(function(){
        ws.send("on_13");
    });

    $("#led_off").click(function(){
        ws.send("off_13");
    });
});
```

補充資料js檔下載位置

- Jquery.min.js →在iLMS下載或是以下網址：

  https://jquery.com/download/

- 將下載好的Jquery.min.js放入以下目的資料夾：
- `$ sudo cp ./你的路徑/jquery.min.js /var/www/html/`
- 同樣把寫好的`ws.js`放到以下目的資料夾：
- `$ sudo cp ./你的路徑/ws.js /var/www/html/`



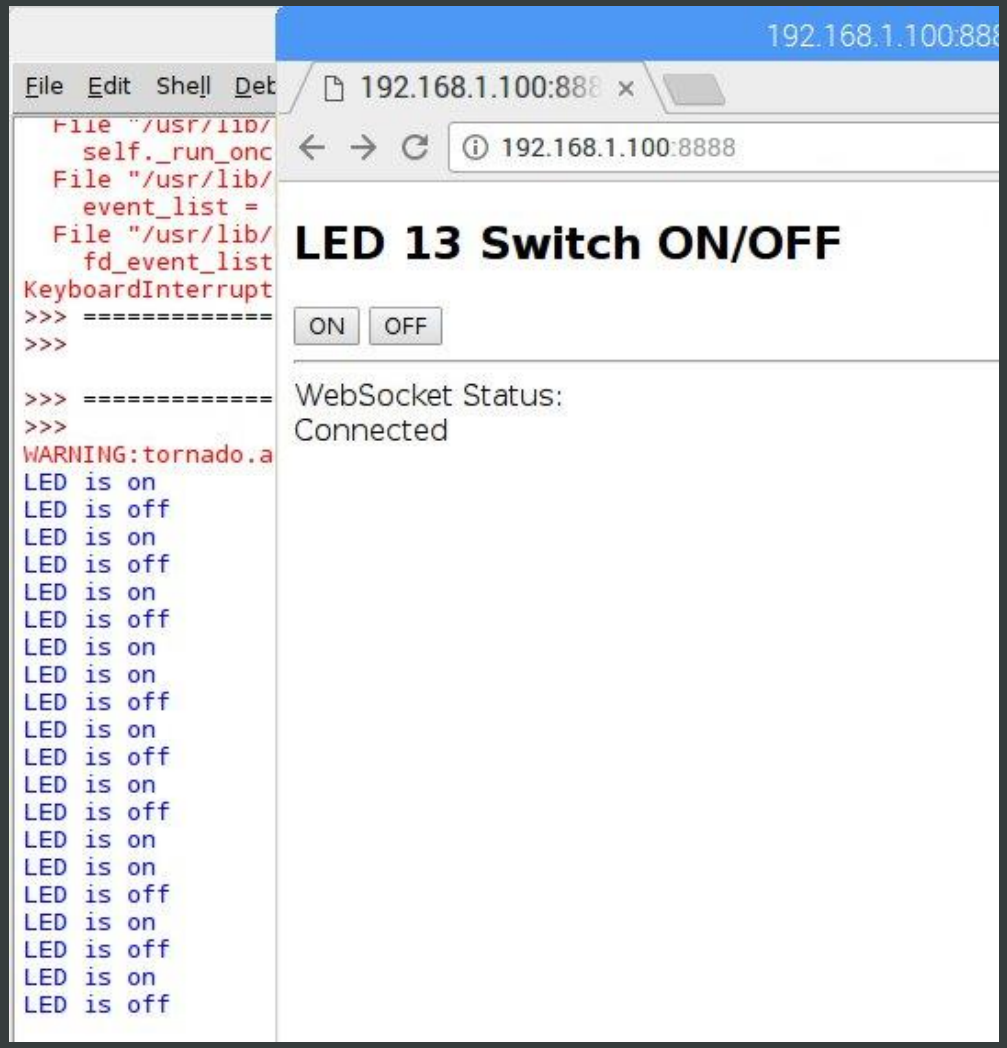
完整程式碼(1/2)

```
1  import tornado.ioloop
2  import tornado.web
3  import tornado.websocket
4  import RPi.GPIO as GPIO
5
6  led = 13
7
8  #Initialize
9  GPIO.setmode(GPIO.BCM)
10 GPIO.setup(led, GPIO.OUT)
11
12 html_text = '''
13 <!DOCTYPE html>
14 <html>
15  <body>
16   <h2>LED 13 Switch ON/OFF</h2>
17    <input type='button' id='led_on' value='ON'>
18    <input type='button' id='led_off' value='OFF'>
19    <script src="//192.168.3.4/jquery.min.js"></script>
20    <script src="//192.168.3.4/ws.js"></script>
21    <hr>
22    WebSocket Status:
23    <br>
24    <div id="ws-status">Waiting...</div>
25  </body>
26 </html>
27 '''
28
```

完整程式碼(2/2)

```
29  class MainApp (tornado.web.RequestHandler):
30      def get(self):
31          self.write(html_text)
32
33  class WSHandler (tornado.websocket.WebSocketHandler):
34      def on_message(self, message):
35          if message == "on_13":
36              GPIO.output(led, True)
37              print("LED is on")
38          if message == "off_13":
39              GPIO.output(led, False)
40              print("LED is off")
41
42  app = tornado.web.Application([
43      (r"/", MainApp),
44      (r'/ws', WSHandler),
45      ])
46
47  app.listen(8888)
48  tornado.ioloop.IOLoop.instance().start()
```

執行結果


File Edit Shell Deb
File "/usr/lib/
self._run_onc
File "/usr/lib/
event_list =
File "/usr/lib/
fd_event_list
KeyboardInterrupt
>>> ==========
>>>
>>> ==========
>>>
WARNING:tornado.a
LED is on
LED is off
LED is on
LED is off
LED is on
LED is off
LED is on
LED is on
LED is off
LED is on
LED is off
LED is on
LED is off
LED is on
LED is on
LED is off
LED is on
LED is off
LED is on
LED is off
192.168.1.100:888
192.168.1.100:888
192.168.1.100:8888
LED 13 Switch ON/OFF
ON
OFF
WebSocket Status:
Connected

雲端服務範例
DropBox

安裝dropbox雲端上傳client端

Step1 – 建立dropbox資料夾

```
$ mkdir ~/dropbox
$ cd ~/dropbox
```

Step2 – 下載Dropbox-Uploader資料夾

```
$ git clone https://github.com/andreafabrizi/Dropbox-Uploader.git
```

Step3 – 進入Dropbox-Uploader資料夾

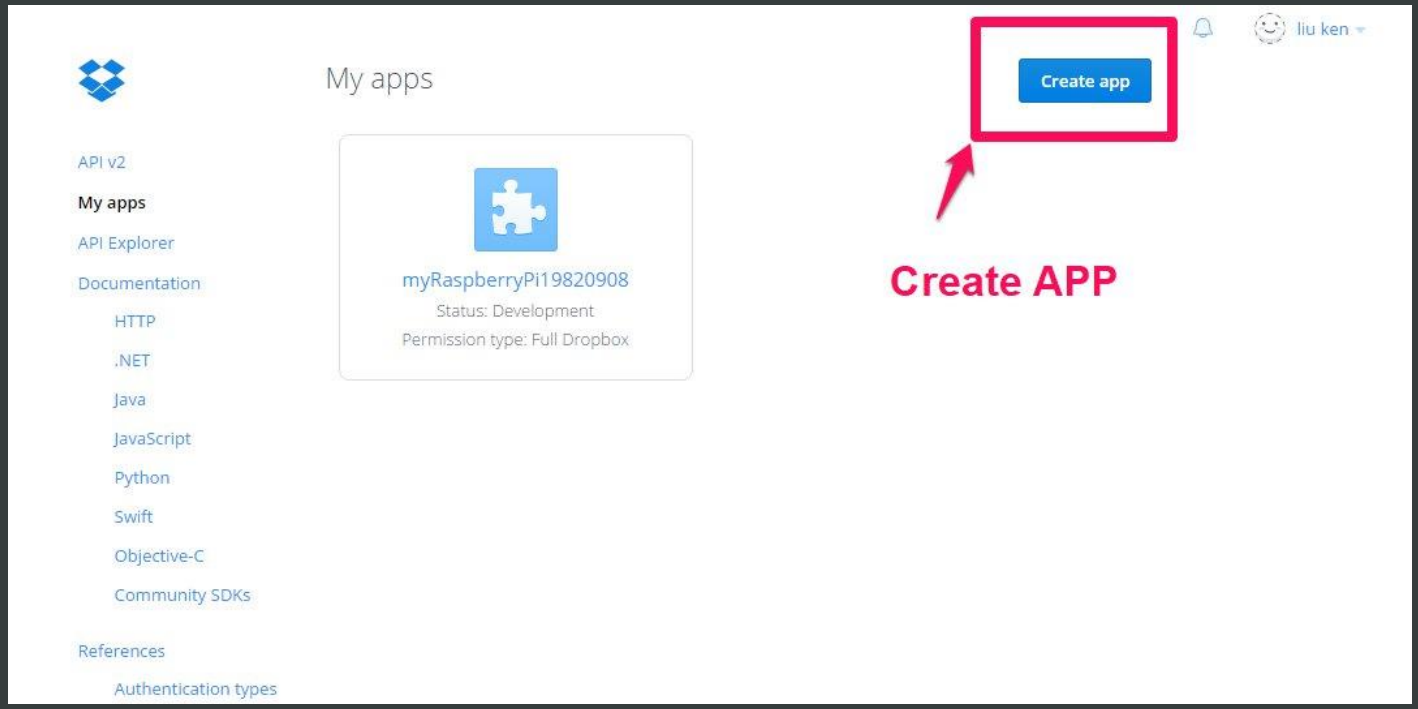
```
$ cd Dropbox-Uploader
```

```
pi@raspberrypi_testbed:~/dropbox $ cd Dropbox-Uploader/
pi@raspberrypi_testbed:~/dropbox/Dropbox-Uploader $ ls
CHANGELOG.md  dropbox_uploader.sh  LICENSE     testUnit.sh
Dockerfile    dropShell.sh         README.md
pi@raspberrypi_testbed:~/dropbox/Dropbox-Uploader $
```

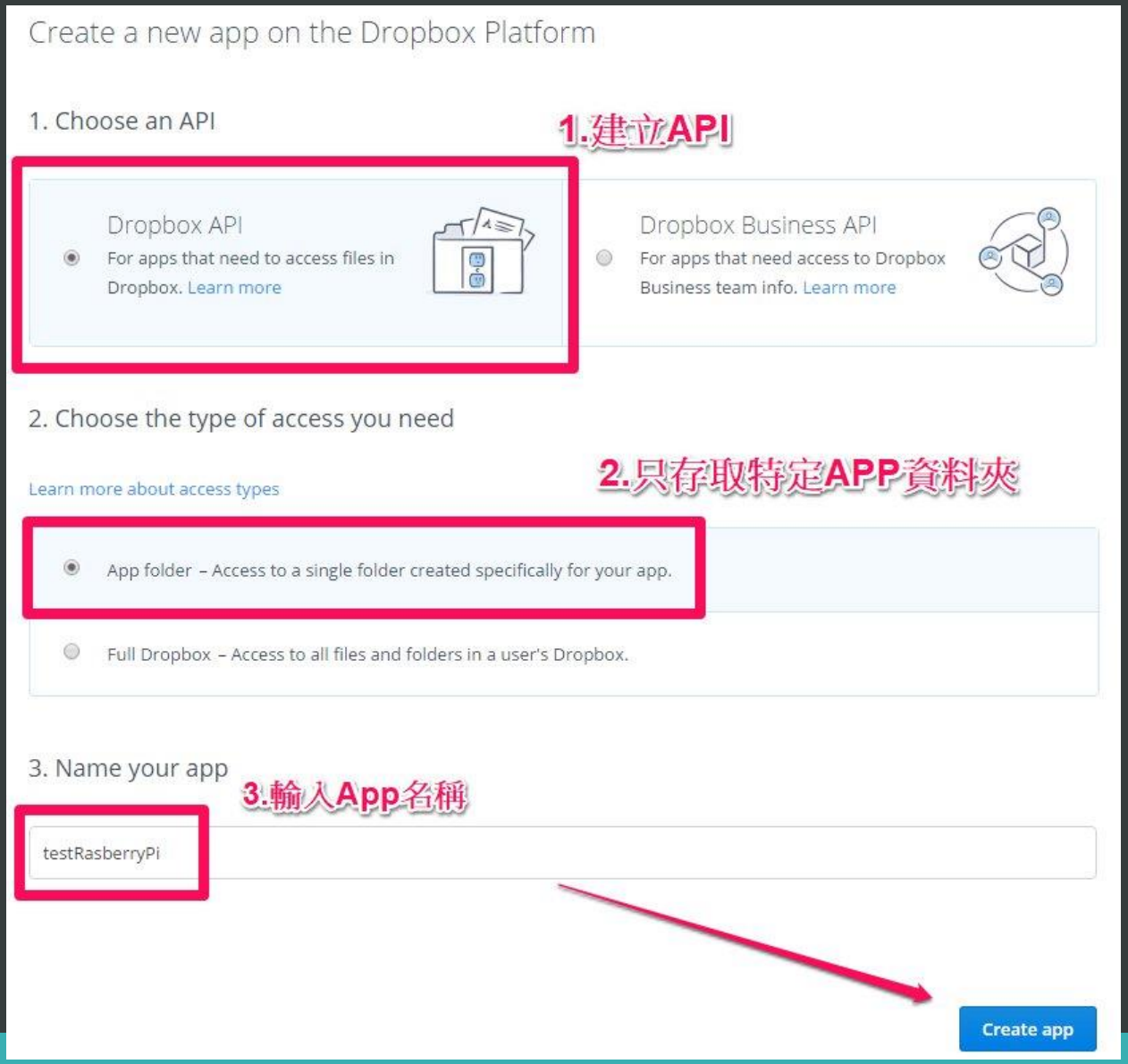
取得dropbox API Key

Step4 – 到dropbox網站取得API Key

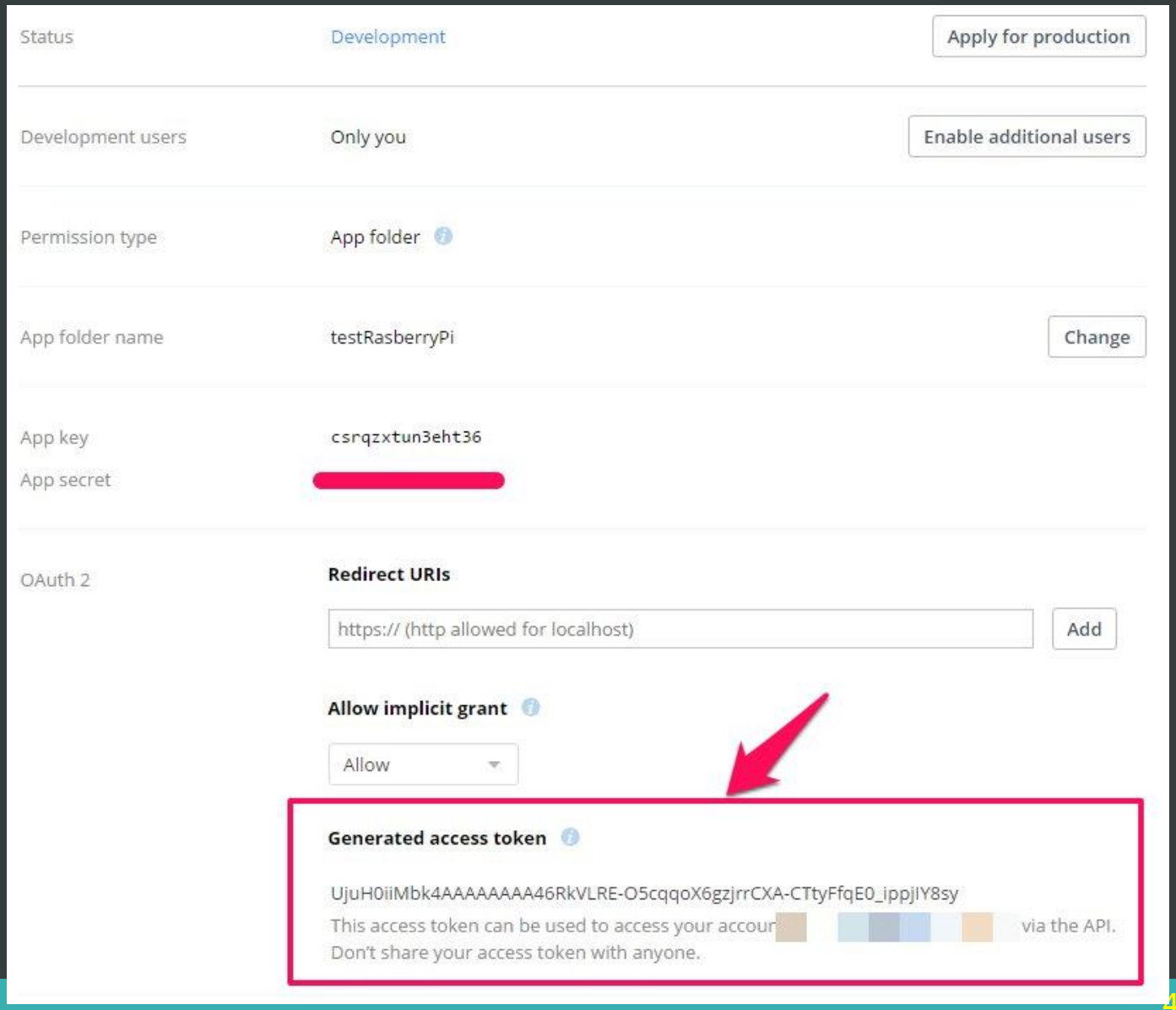
https://www.dropbox.com/developers/apps



填寫新的APP設定


Create a new app on the Dropbox Platform
1. Choose an API
1.建立API
Dropbox API
For apps that need to access files in Dropbox. Learn more
Dropbox Business API
For apps that need access to Dropbox Business team info. Learn more
2. Choose the type of access you need
2.只存取特定APP資料夾
Learn more about access types
App folder – Access to a single folder created specifically for your app.
Full Dropbox – Access to all files and folders in a user's Dropbox.
3. Name your app
3.輸入App名稱
testRasberryPi
Create app

Generated access token



設定token

Step5 – 執行dropbox-uploader

```
$ ./dropbox_uploader.sh
```

```
This is the first time you run this script, please follow the instructions:

 1) Open the following URL in your Browser, and log in using your account: https://www.dropbox.com/d
evelopers/apps
 2) Click on "Create App", then select "Dropbox API app"
 3) Now go on with the configuration, choosing the app permissions and access restrictions to your D
ropBox folder
 4) Enter the "App Name" that you prefer (e.g. MyUploader16480759813706)

Now, click on the "Create App" button.

When your new App is successfully created, please click on the Generate button
under the 'Generated access token' section, then copy and paste the new access token here:

# Access token: UjuH0iiMbk4AAAAAAAA46RkVLRE-O5cqqoX6gzjrrCXA-CTtyFfqE0_ippjIY8sy
```

Step6 – 輸入access token

```
# Access token:你的key
```

***若輸入錯誤, 用 $ ./dropbox_uploader.sh unlink 解開連接

確認key值

Step7 – 確認Key

輸入 y

```
# Access token: UjuH0iiMbk4AAAAAAAA46RkVLRE-O5cqqoX6gzjrrCXA-CTtyFfqE0_ippjIY8sy

> The access token is UjuH0iiMbk4AAAAAAAA46RkVLRE-O5cqqoX6gzjrrCXA-CTtyFfqE0_ippjIY8sy. Looks ok? [
y/N]: y
```

Step8 – 測試檔案上傳到Dropbox資料夾

```
$ cd ~/dropbox/Dropbox-Uploader/
$ ./dropbox_uploader.sh upload ~/img.jpg /001.jpg
```

```
pi@raspberrypi_testbed:~/dropbox/Dropbox-Uploader $ ./dropbox_uploader.sh upload ~/img.jpg /001.jpg
 > Uploading "/home/pi/img.jpg" to "/001.jpg"... DONE
```

***上傳的圖片會放在dropbox\應用程式\應用程式名稱\

Dropbox-Uploader指令

upload

```
$ ./dropbox_uploader.sh upload /home/pi/來源圖檔.jpg /目的圖檔.jpg
```

download

```
$ ./dropbox_uploader.sh download /下載圖檔.jpg /home/pi/存檔圖檔.jpg
```

mkdir

```
$ ./dropbox_uploader.sh mkdir <資料夾名稱>
```

help

```
$ ./dropbox_uploader.sh help
```

ITRI TTS Web服務

- http://tts.itri.org.tw/development/javascript_api.php

回家作業

- 請解決透過WebSocket方式能控制WS2812B LED燈環，Web介面上可以選擇三種不同顏色，並透過即時方式更改WS2812B上的全部燈點或部分燈點效果，若能控制更多顏色、更多效果分數越高。