Raspberry Pi Camera and Python

台灣樹莓派 <sosorry@raspberrypi.com.tw>

2018/01/16 @ITRI

社群 x 活動

- 專注於 Raspberry Pi 應用與推廣
- 舉辦社群聚會 / 工作坊 / 讀書會 / 黑客松
- Website:
  - https://www.raspberrypi.com.tw/

- Facebook:
  - 搜尋 RaspberryPi.Taiwan
  - https://www.facebook.com/RaspberryPi.Taiwan

分享 x 教學

- COSCUP,MakerConf,PyCon 講者
- 投影片
  - http://www.slideshare.net/raspberrypi-tw/presentations
- 程式碼
  - https://github.com/raspberrypi-tw

大綱

- 相機原理與應用
- 控制 Raspberry Pi Camera
  - 使用指令列
  - 使用 Python
  - 串接 imagga 網路服務
- Camera 和 Webcam
- 影像串流
  - 使用 RTSP + H.264
  - 使用 HTTP + MJPG

今日環境

- 硬體：Raspberry Pi 3
- 作業系統：2017-11-29-raspbian-stretch.img

- 為了可以使用USB轉TTL傳輸線
  - 修改/boot/config.txt，新增三行
    - dtoverlay=pi3-miniuart-bt
    - core_freq=250
    - enable_uart=1

```
55 # Enable audio (loads snd_bcm
56 dtparam=audio=on
57 dtoverlay=pi3-miniuart-bt
58 core_freq=250
59 enable_uart=1
```

新增三行

  - 修改/boot/cmdline.txt，將quiet splash的quiet移除

```
1 dwc_otg.lpm_enable=0 console=serial0,115200
  console=tty1 root=/dev/mmcblk0p2 rootfstype=
  ext4 elevator=deadline fsck.repair=yes rootw
  ait quiet splash plymouth.ignore-serial-cons
  oles quiet init=/usr/lib/raspi-config/init_r
  esize.sh
```

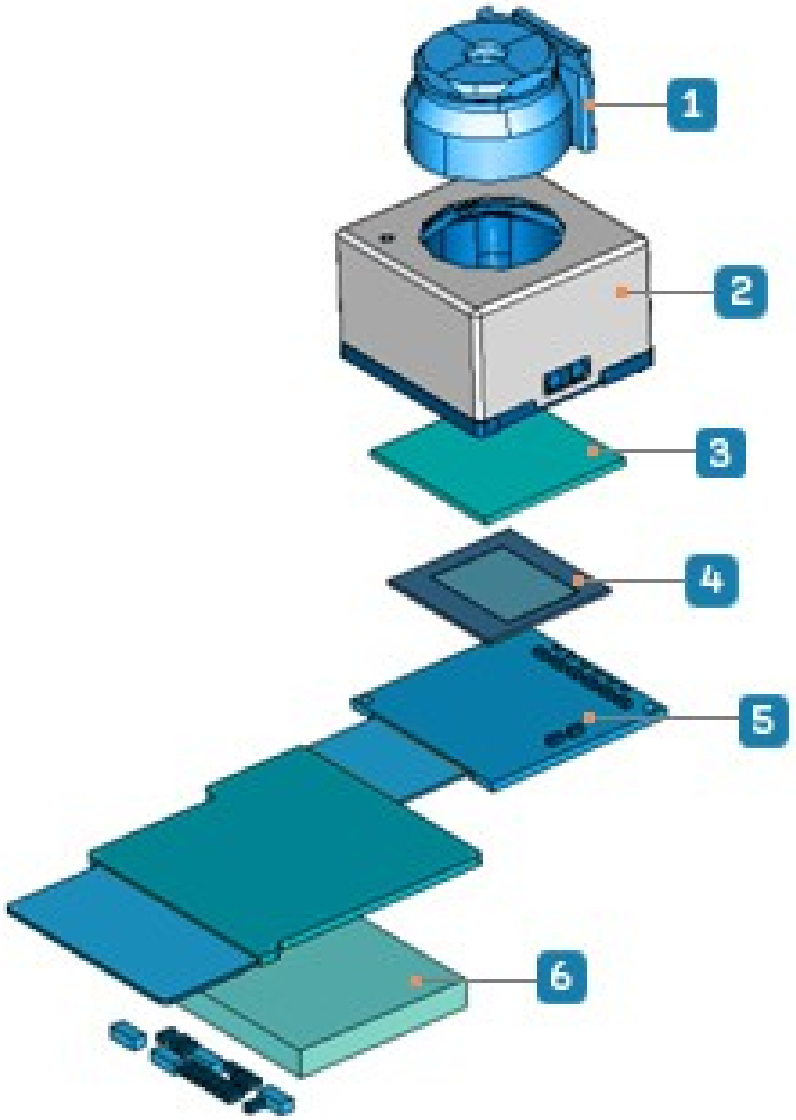
刪除 quiet

安裝今日所需軟體(已安裝)

- $ sudo apt-get update
- $ sudo apt-get install -y festival python-dev python-opencv python-pip x11vnc liblivemedia-dev libv4l-dev cmake libpthread-stubs0-dev vlc python-matplotlib
- $ sudo pip2 install request flask numpy

Raspberry Pi Camera 簡介

從手機相機模組講起



1. Lens(透鏡)
2. VCM(音圈馬達)
3. IR-Cut(紅外光濾片)
4. Sensor(感光元件)
5. PCB(印刷電路板)
6. ISP(影像訊號處理器)

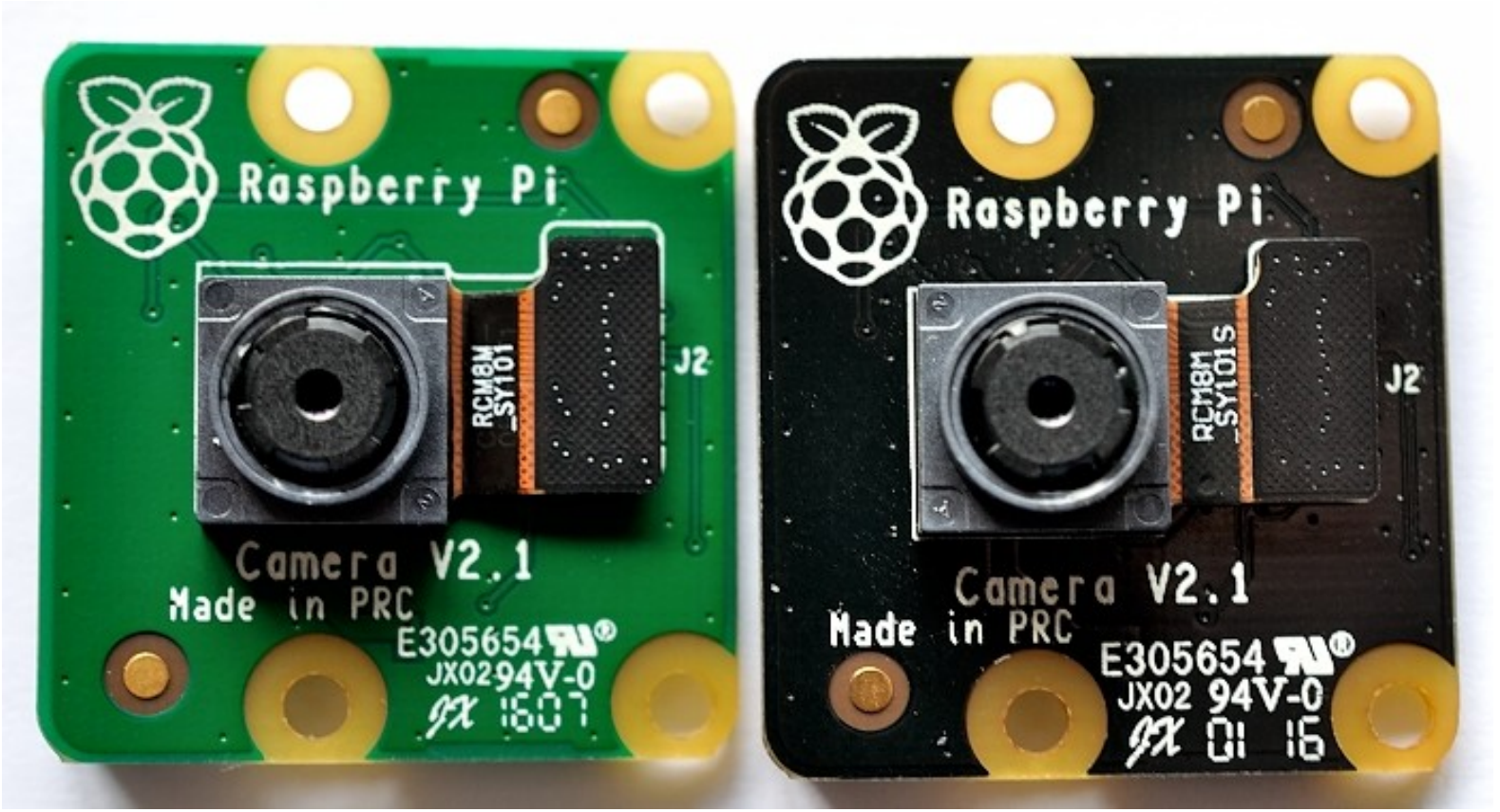
http://www.phonearena.com/news/13MP-camera-tipped-for-Samsung-Galaxy-S-IV_id35168

技術規格 (v2)

- Sensor:Sony IMX219 (8MP)
- 靜態拍照最高解析度:3280 x 2464 pixel
- Pixel Size:1.4 x 1.4 µm
- Lens:f=3.6 mm, f/2.9
- Fixed Focus:1m to infinity
- 動態攝影最高解析度:1080p@30 FPS with H.264/AVC

https://www.raspberrypi.org/documentation/hardware/camera/

Type of Raspberry Pi Camera(v2)

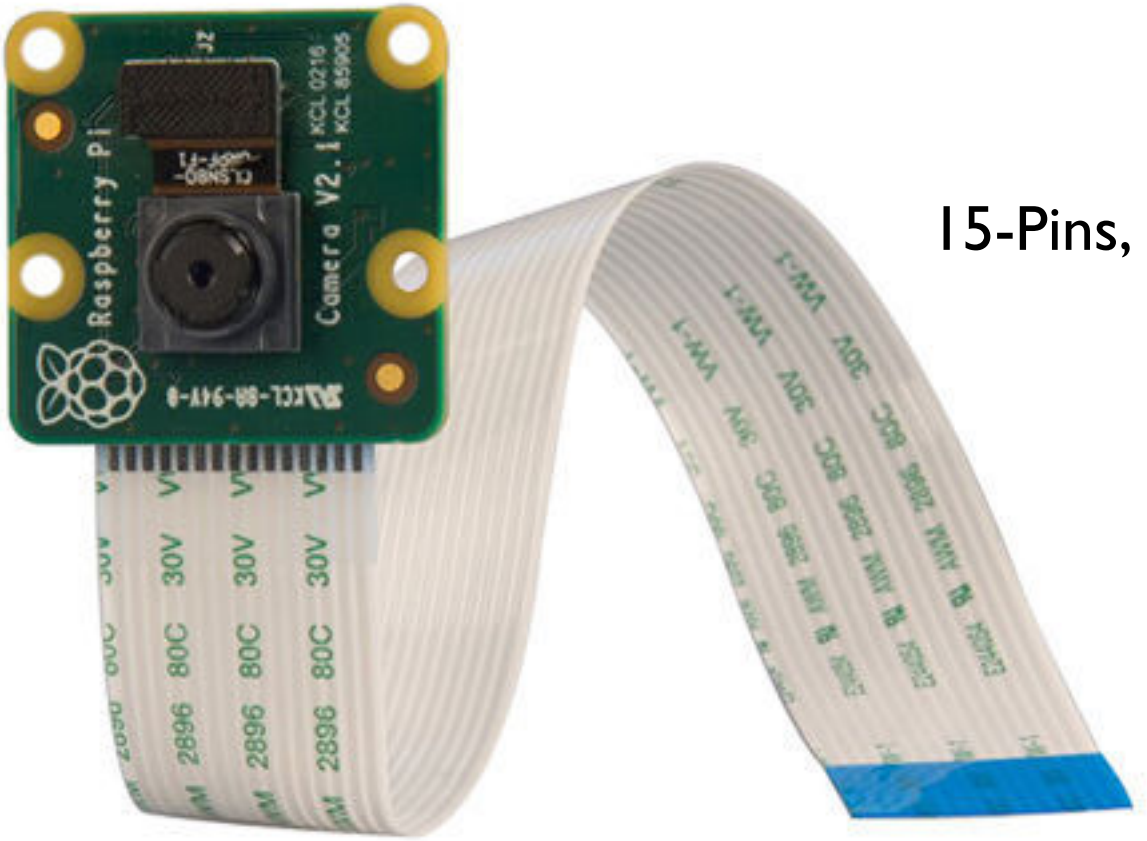


Raspberry Pi Camera Module NoIR Camera Module

http://raspi.tv/2016/new-8-megapixel-raspberry-pi-camera-2-1-launches

Raspberry Pi Camera Module

綠色 PCB 板

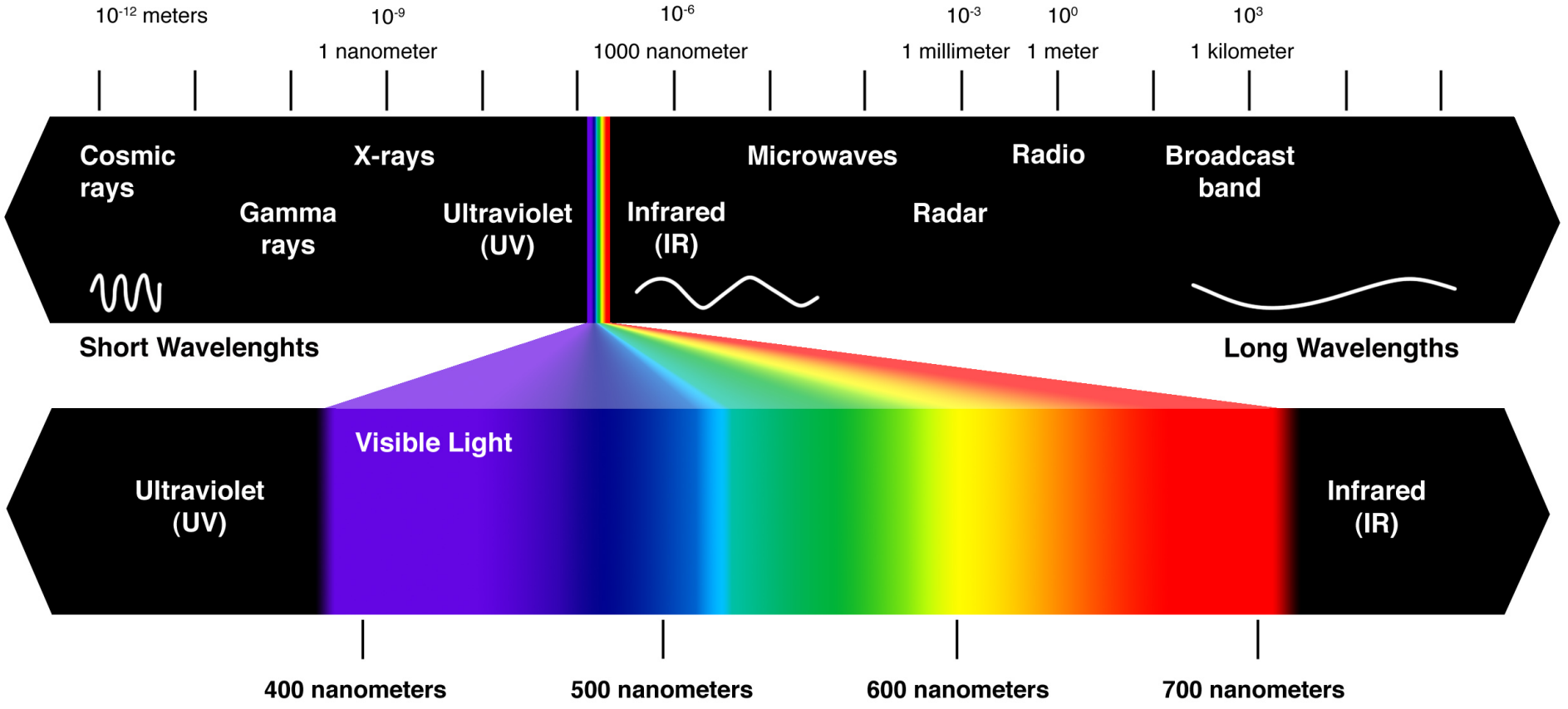


15-Pins, CSI 介面

http://www.raspberrypi-spy.co.uk/2016/04/8-megapixel-raspberry-pi-camera-module-v2/

基礎光學原理

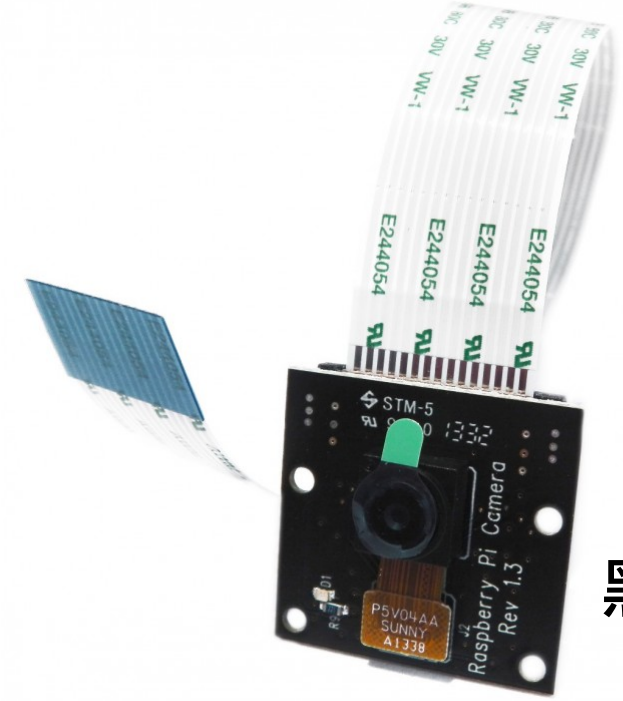
- 問：樹葉為什麼看起來是綠色的？
- 答：因為樹葉吸收了大部分可見光，只反射綠色光



https://www.raspberrypi.org/learning/infrared-bird-box/worksheet/

No IR Camera

- No IR = No 'IR cut filter' installed
- 因此 CMOS 可吸收到不可見光 (Infrared)
- No IR 相機 + 紅外線發光源 = 夜視相機



黑色 PCB 板

外接式紅外光

可控式紅外光

https://www.buyapi.ca/product/raspberry-pi-noir-camera-module-v2-8mp/

兩種相機效果比較

1. 非 NoIR 相機

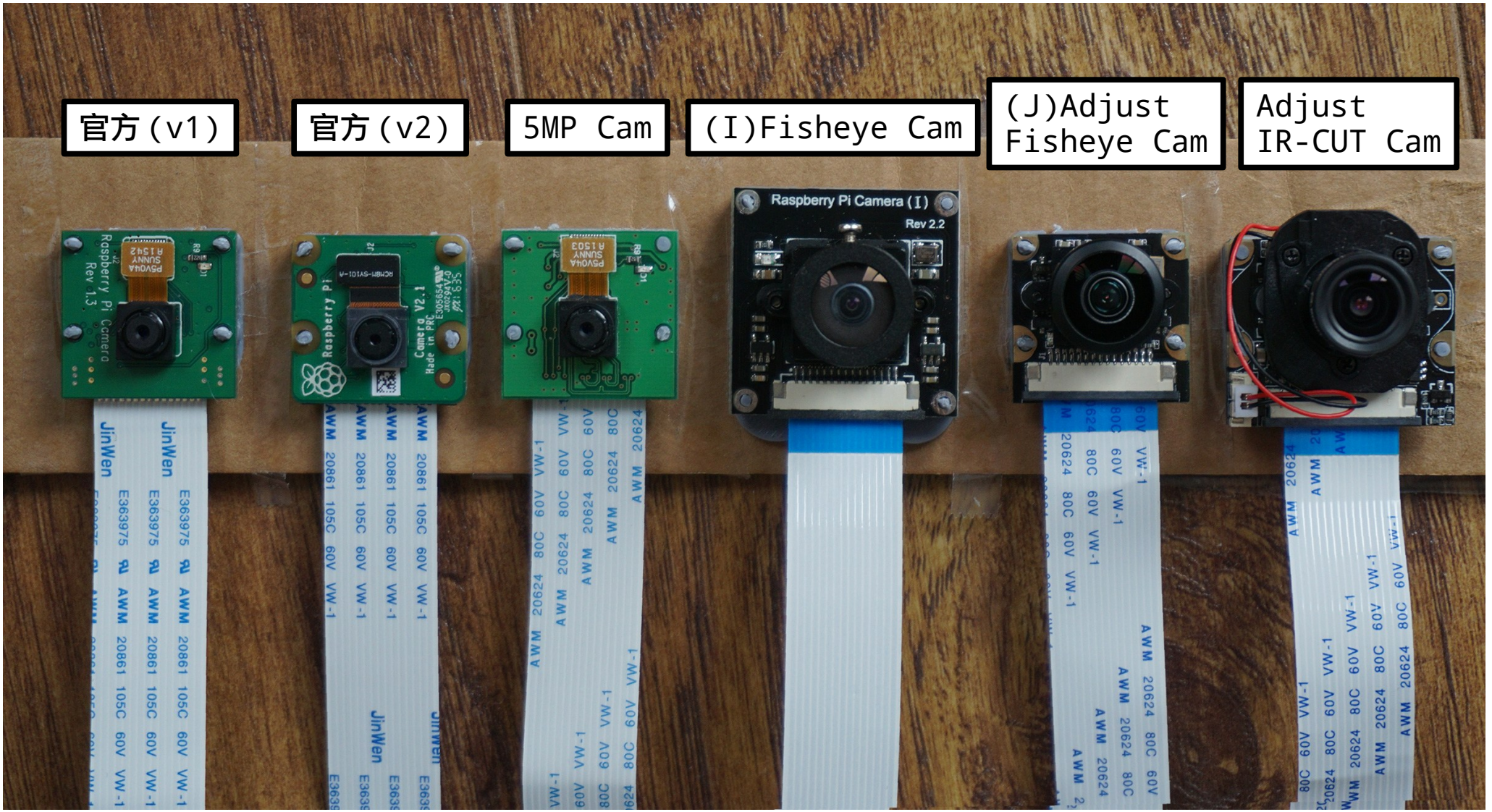
2. NoIR 相機

3. NoIR 相機

4. NoIR 相機 + 藍色濾光片

https://www.raspberrypi.org/magpi/issues/18/

更多 Camera



http://www.semifluid.com/2017/01/23/raspberry-pi-camera-comparison/

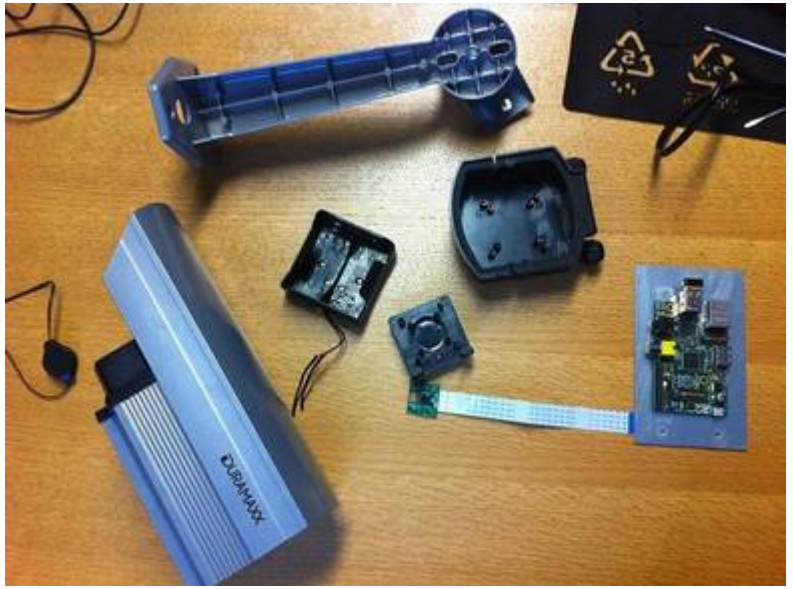
Raspberry Pi Camera Comparison

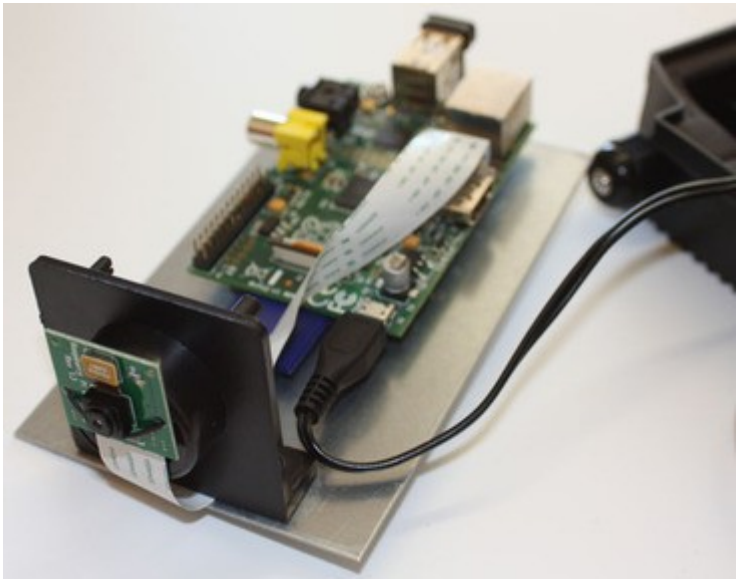
semifluid.com

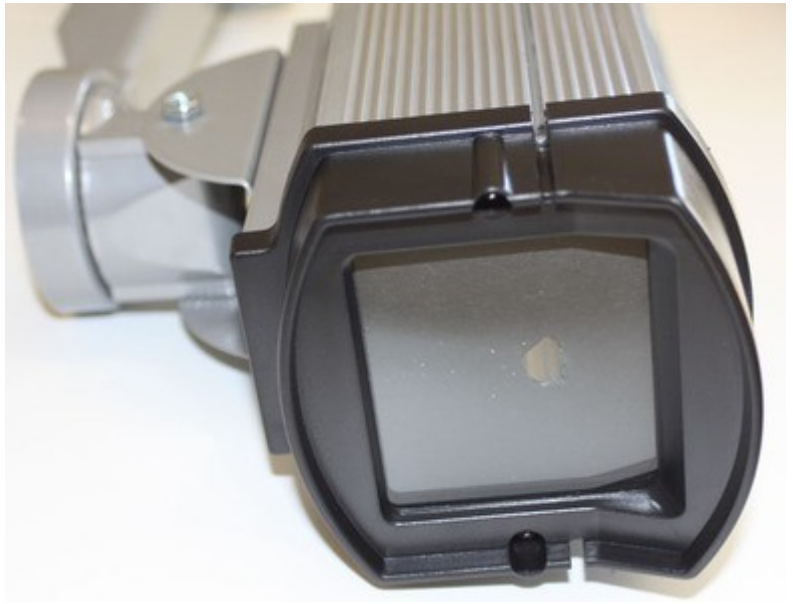
http://www.semifluid.com/2017/01/23/raspberry-pi-camera-comparison/

Raspberry Pi Camera 應用介紹

IP Camera



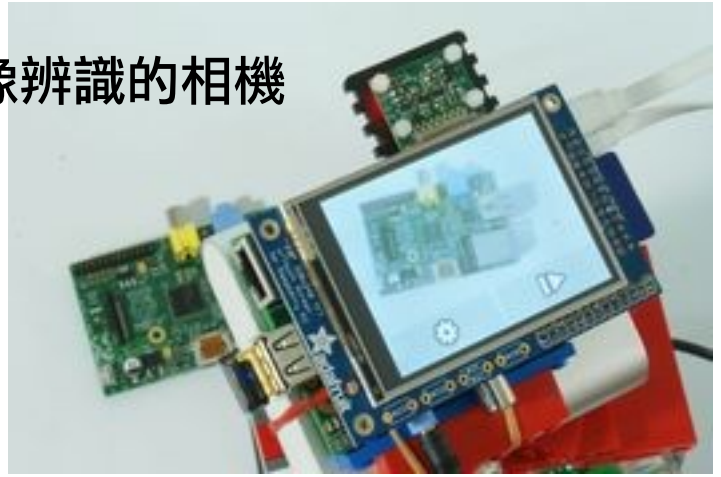


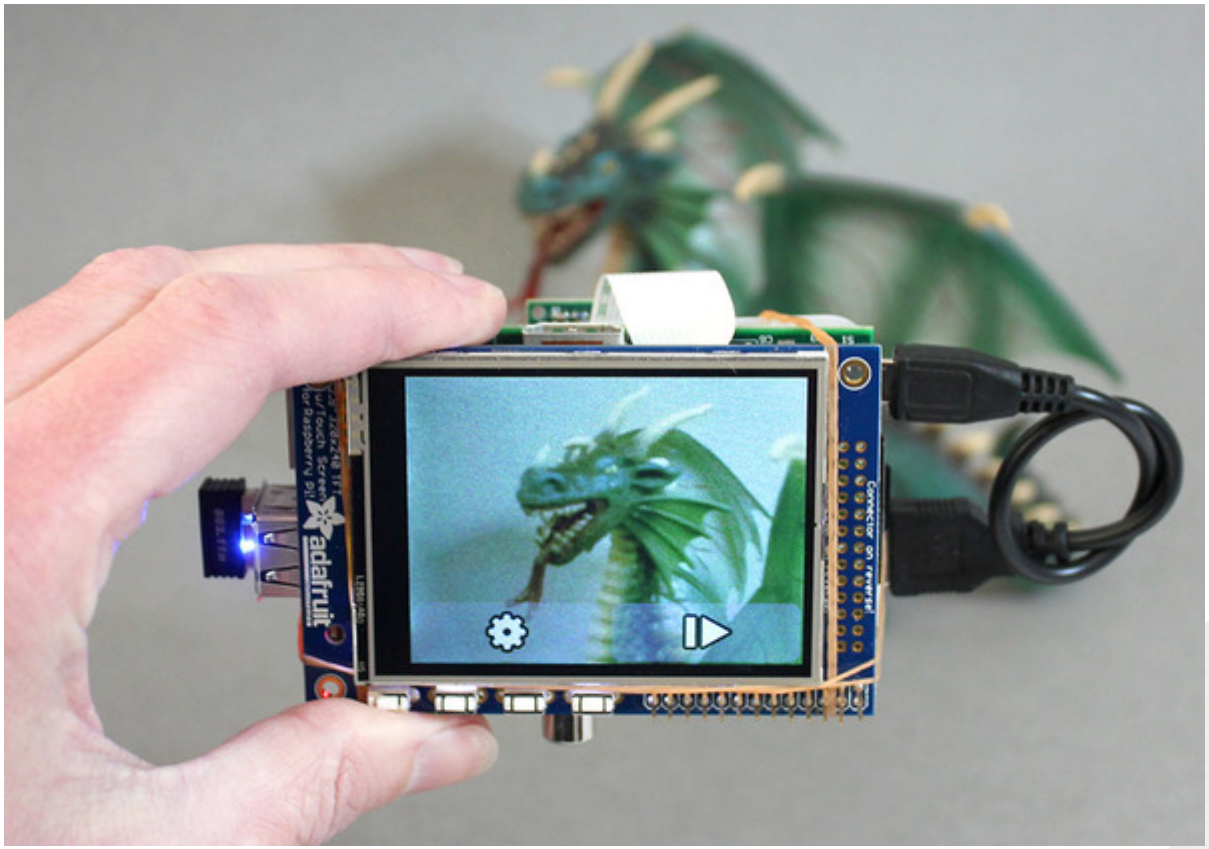


http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera

雲端相機

可做影像辨識的相機





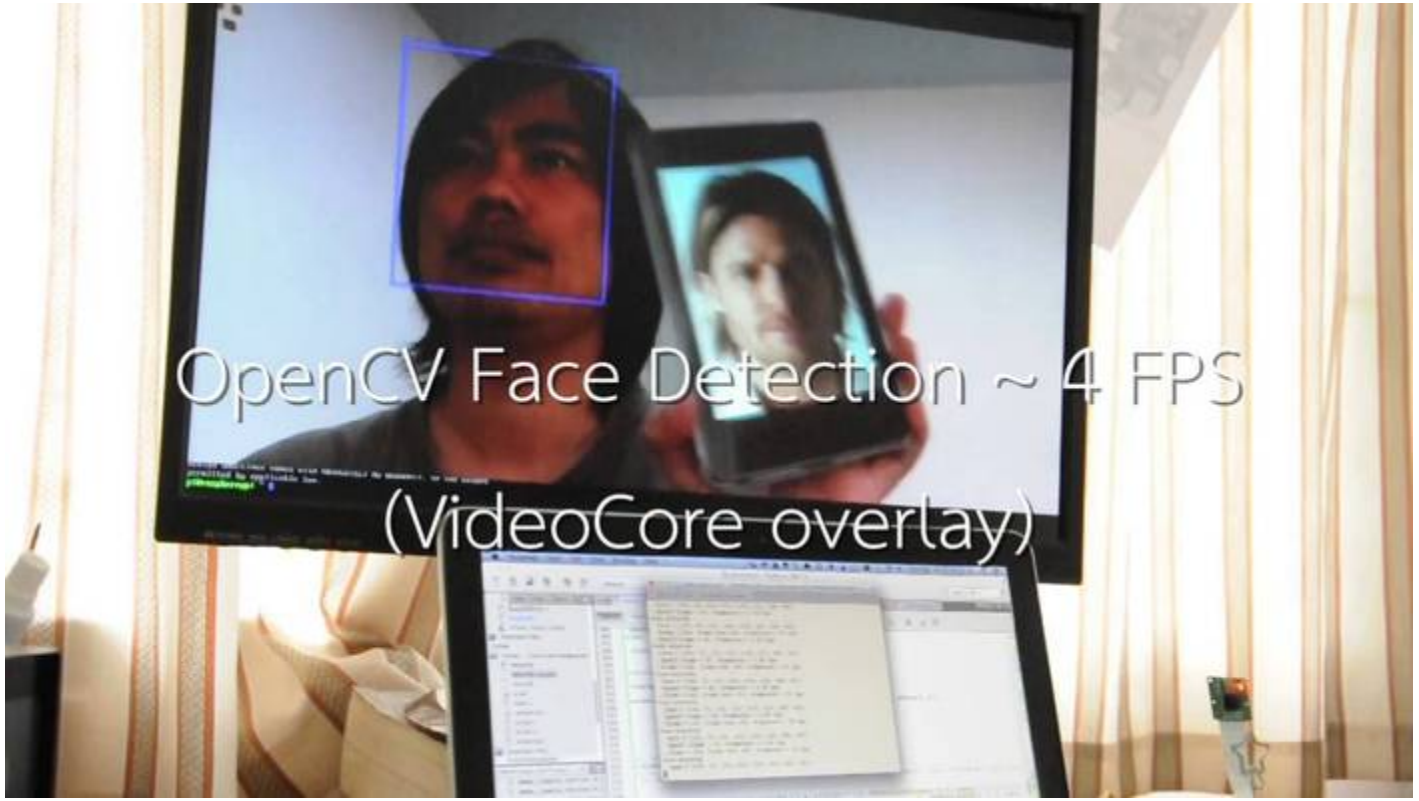
imagga

can you imaggine!

http://learn.adafruit.com/diy-wifi-raspberry-pi-touch-cam

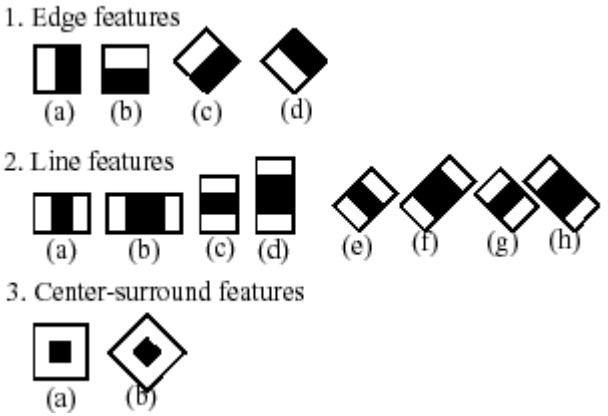
Pi 立得

人臉偵測與追蹤



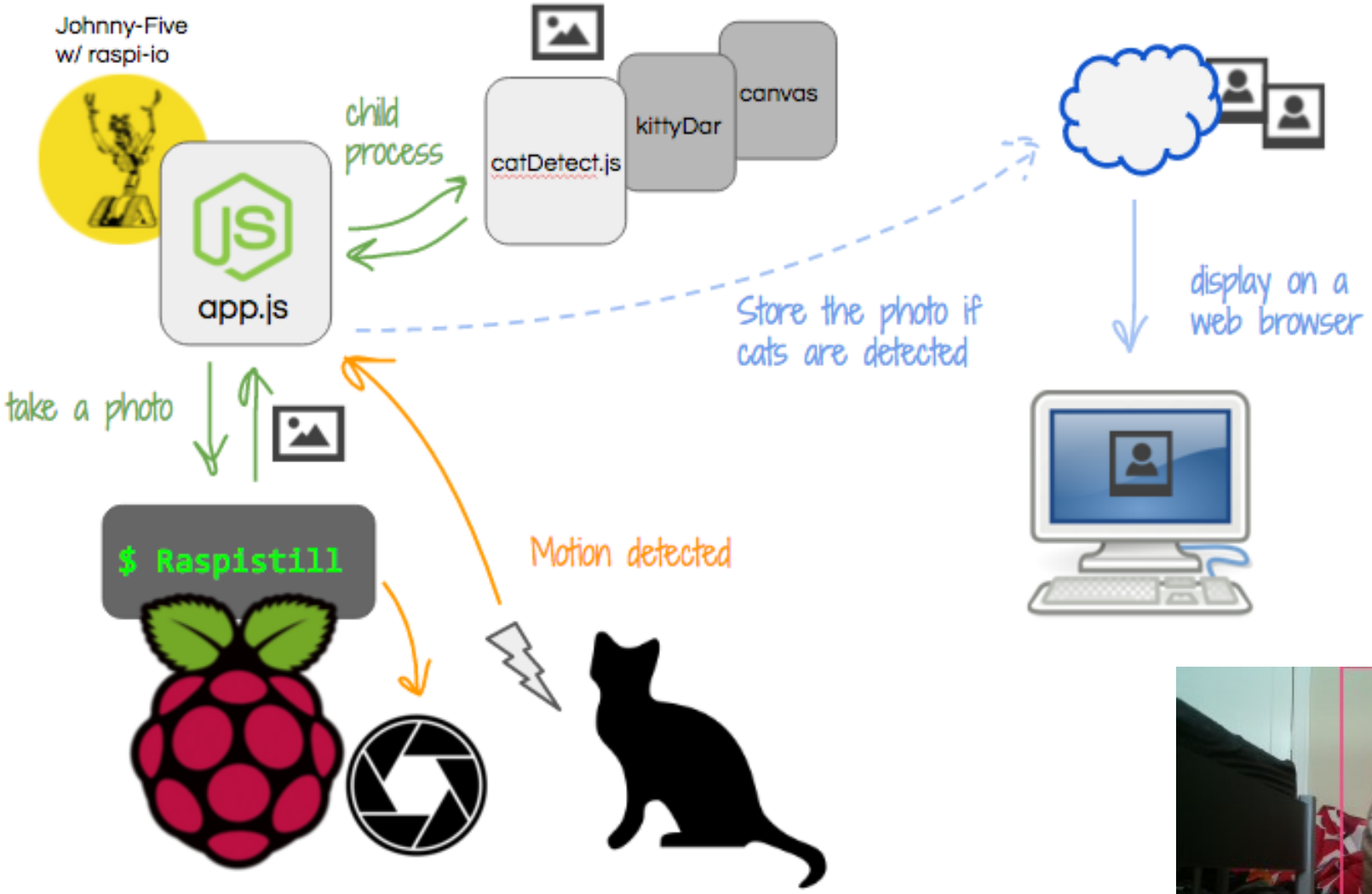
OpenCV

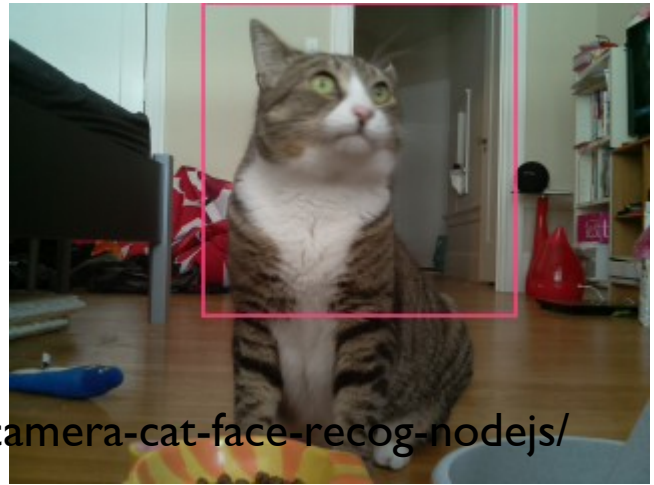
Haar	Daubechies-4 (D4)
H = {1 / sqrt(2), 1 / sqrt(2) } G = {1 / sqrt(2), -1 / sqrt(2) }	H = { (1 + sqrt(3)) / (4 * sqrt(2)), (3 + sqrt(3)) / (4 * sqrt(2)), (3 - sqrt(3)) / (4 * sqrt(2)), (1 - sqrt(3)) / (4 * sqrt(2)) } G[0] = H[3], G[1] = -H[2], G[2] = H[1], G[3] = -H[0]



https://github.com/tasanakorn/rpi-mmal-demo

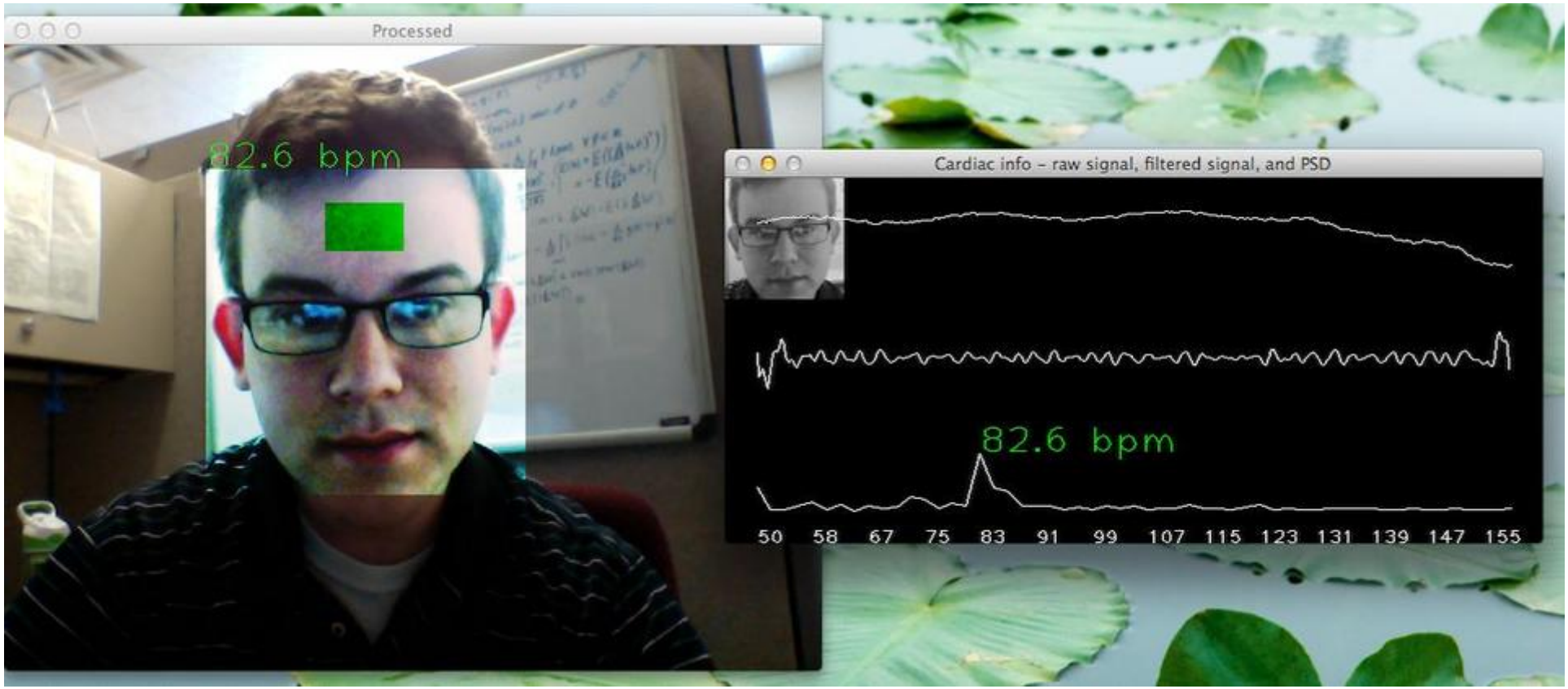
貓臉偵測





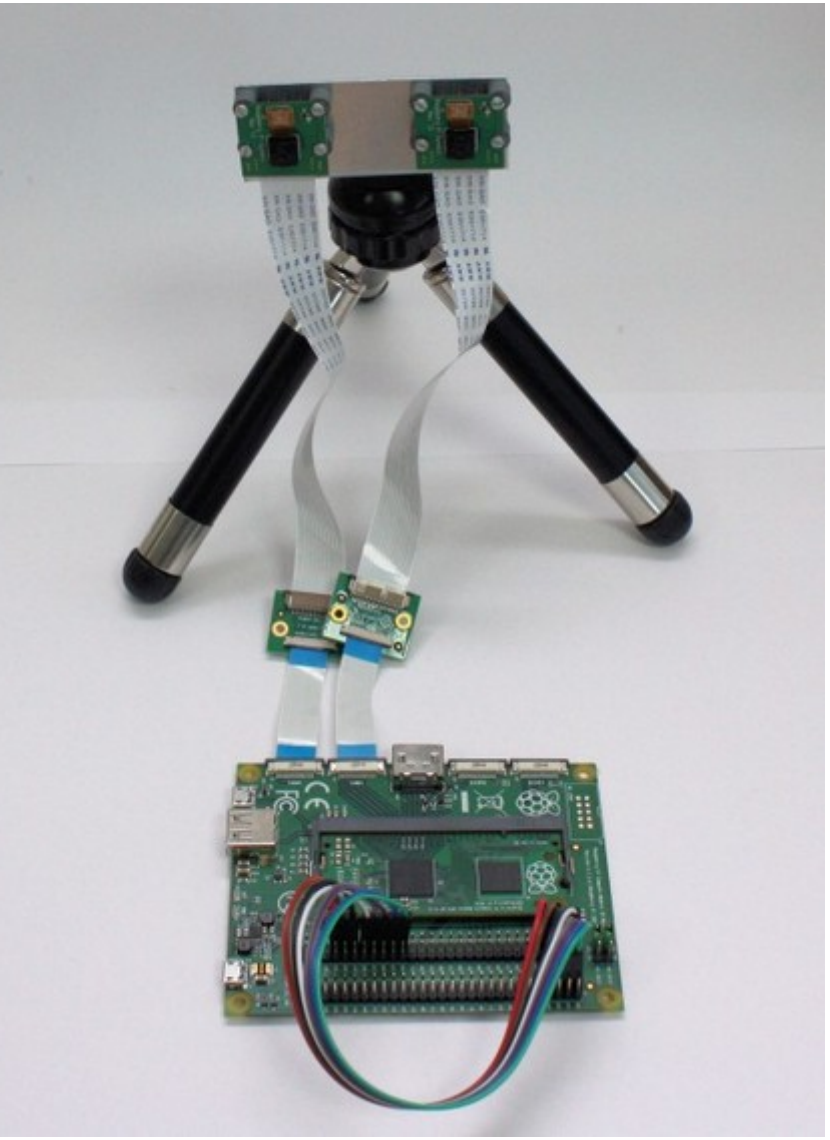
http://www.girliemac.com/blog/2015/12/25/kittycam-raspberrypi-camera-cat-face-recog-nodejs/

脈博辨識



https://github.com/thearn/webcam-pulse-detector

3D 建模計算深度 /2 Cameras



http://www.raspberrypi.org/real-time-depth-perception-with-the-compute-module/

3D 掃描 /50 Cameras

http://www.pi3dscan.com/

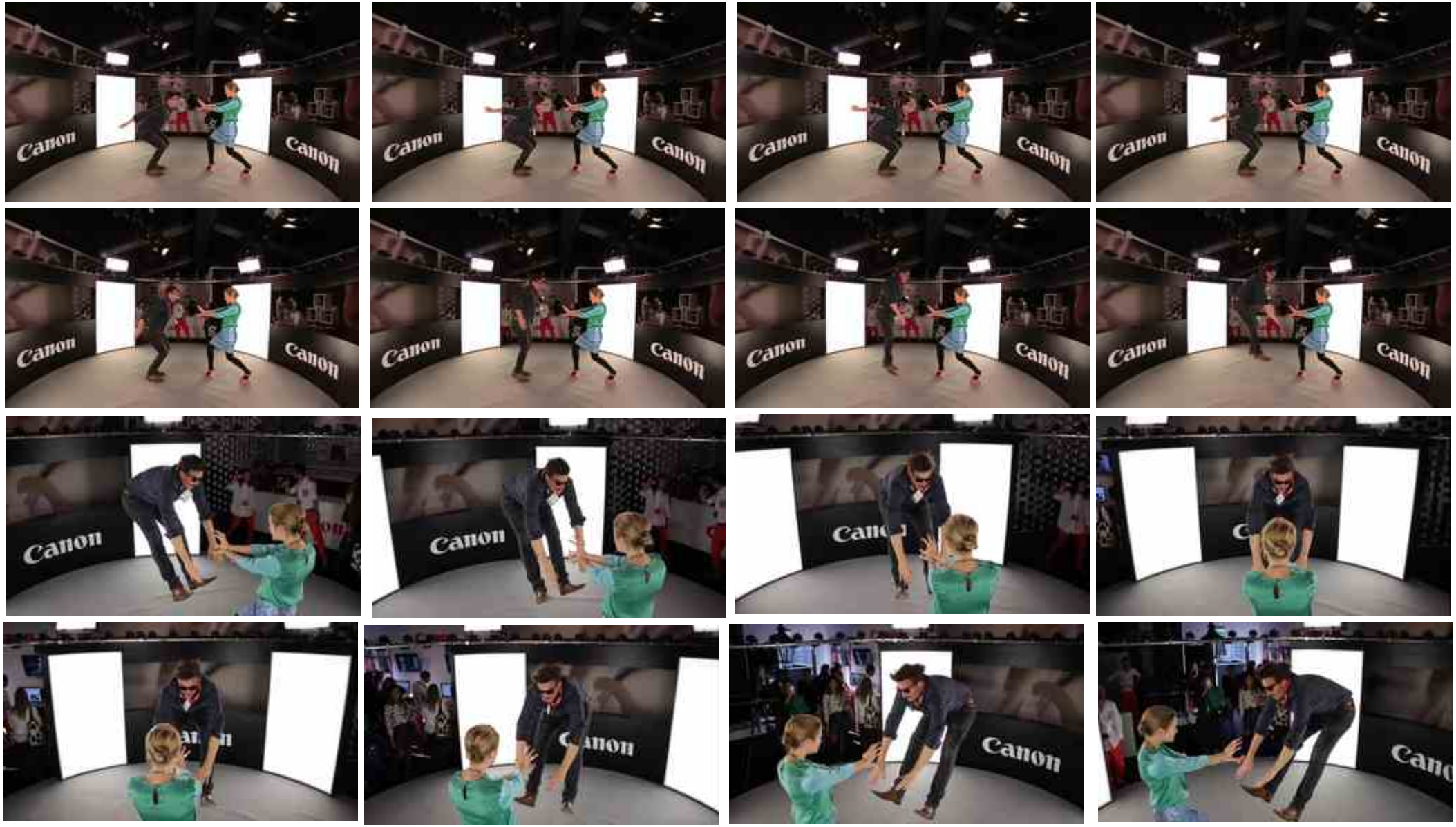
效果 + Autodesk Recap

http://www.pi3dscan.com/

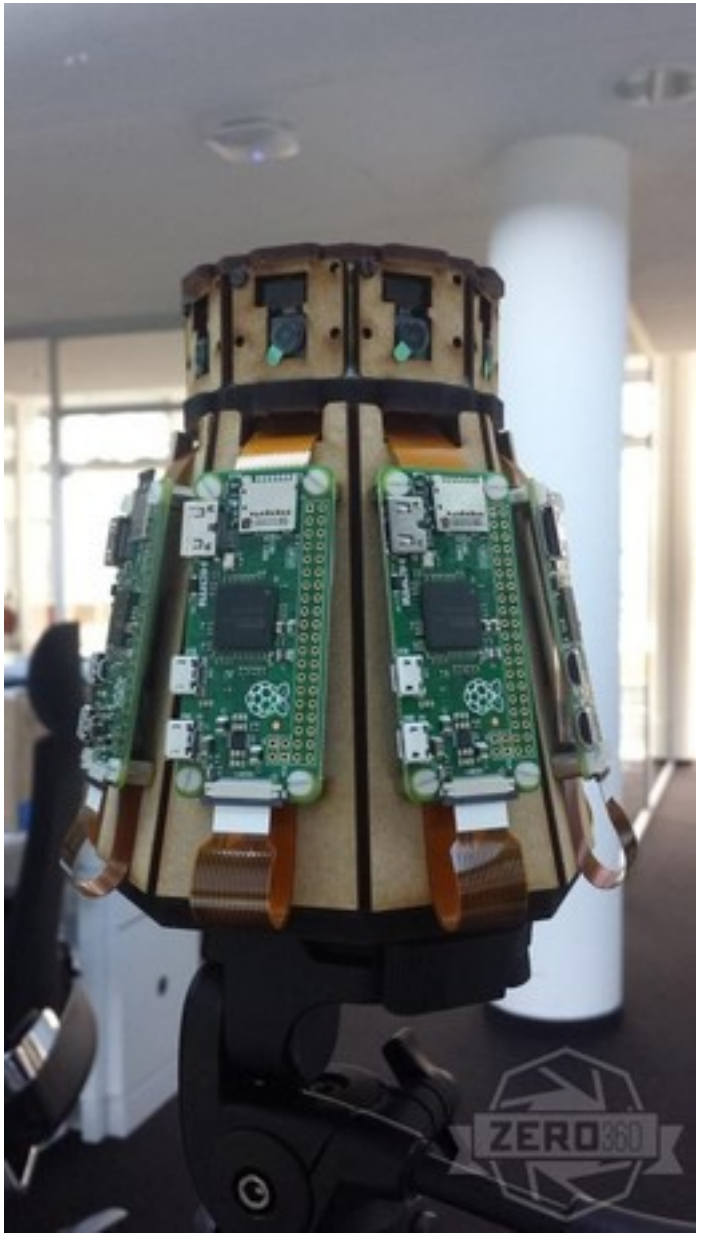
360 度照片

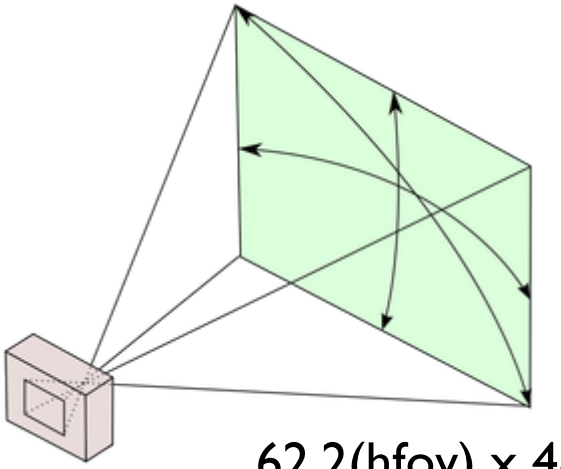
https://vimeo.com/77218985

效果

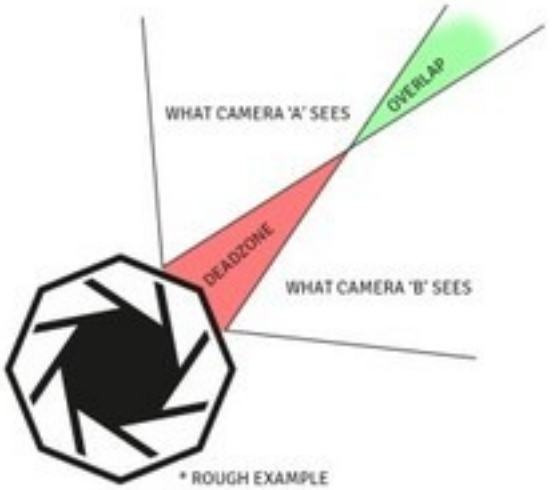


ZERO360





62.2(hfov) x 48.8(vfov)



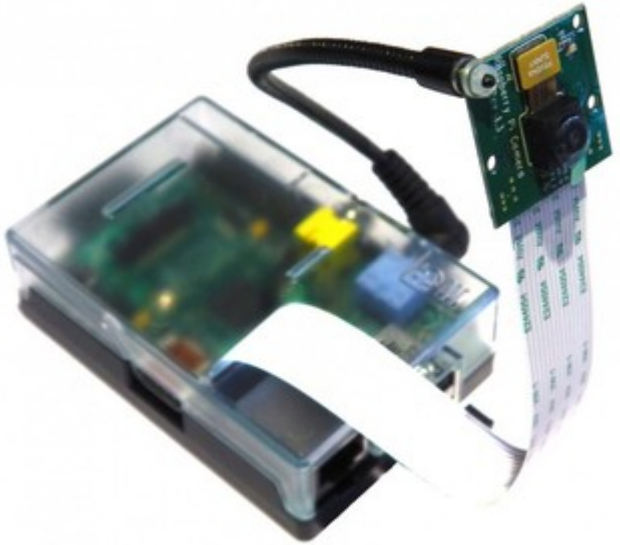
http://raspberryjamberlin.de/introducing-zero360/

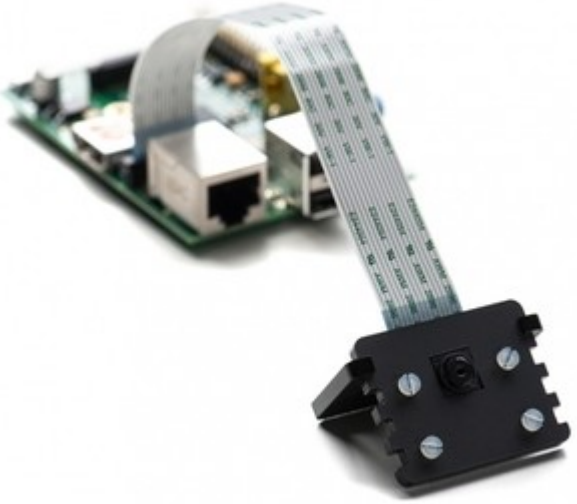
八台相機拍出另一種全景照片

http://raspberryjamberlin.de/introducing-zero360/

Camera 改裝套件

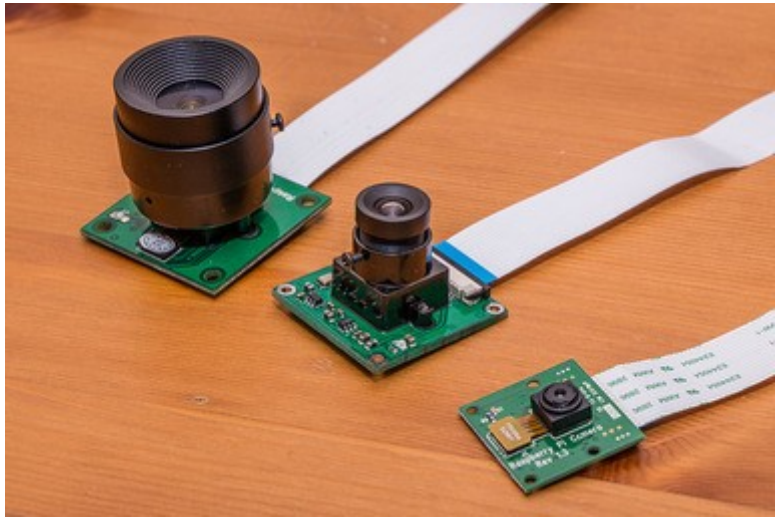
固定的機構





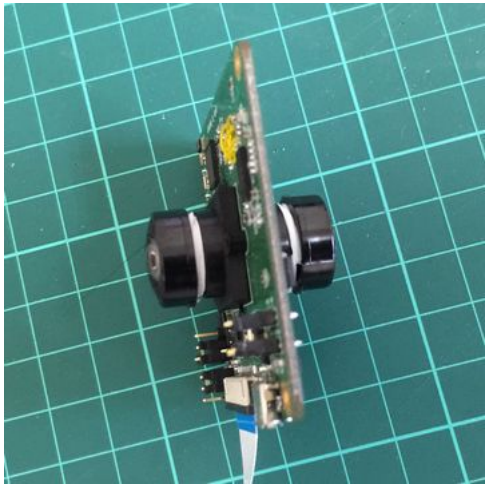
http://www.modmypi.com

相機模組改裝

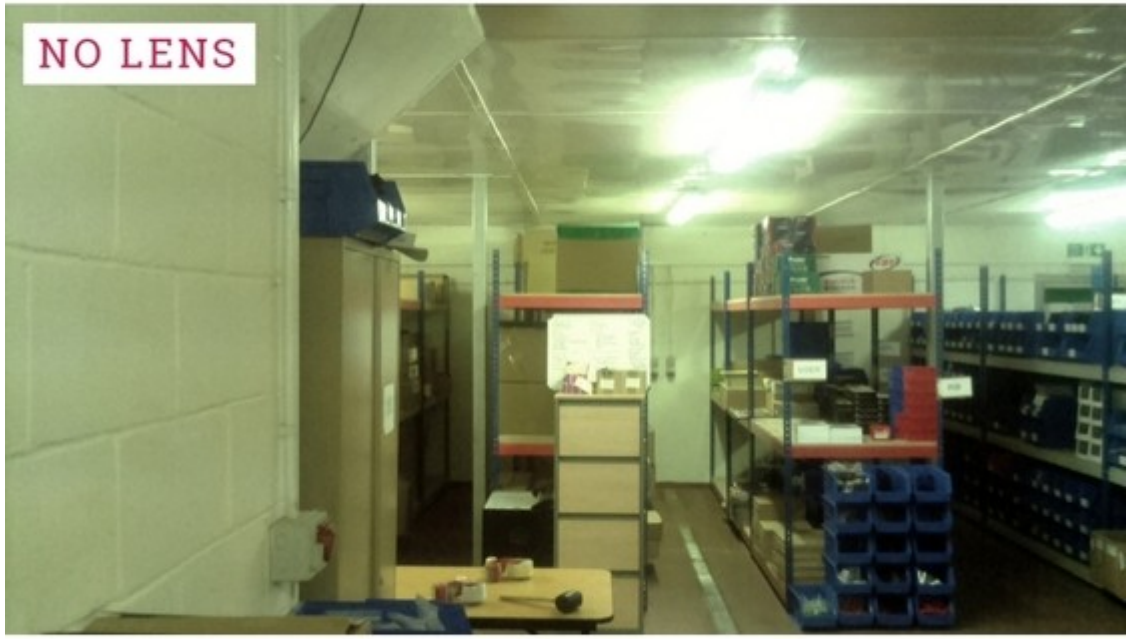


https://www.adafruit.com/products/1937
https://www.flickr.com/people/100320847@N06/

360 相機

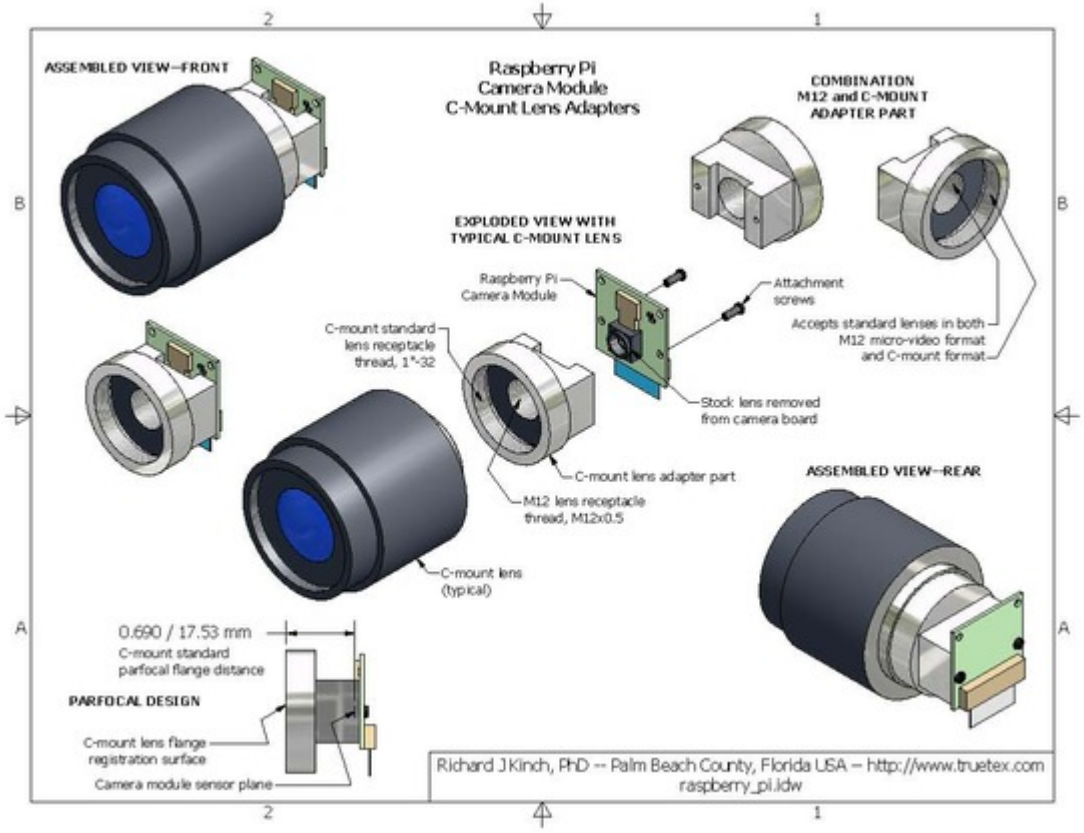


鏡頭改裝

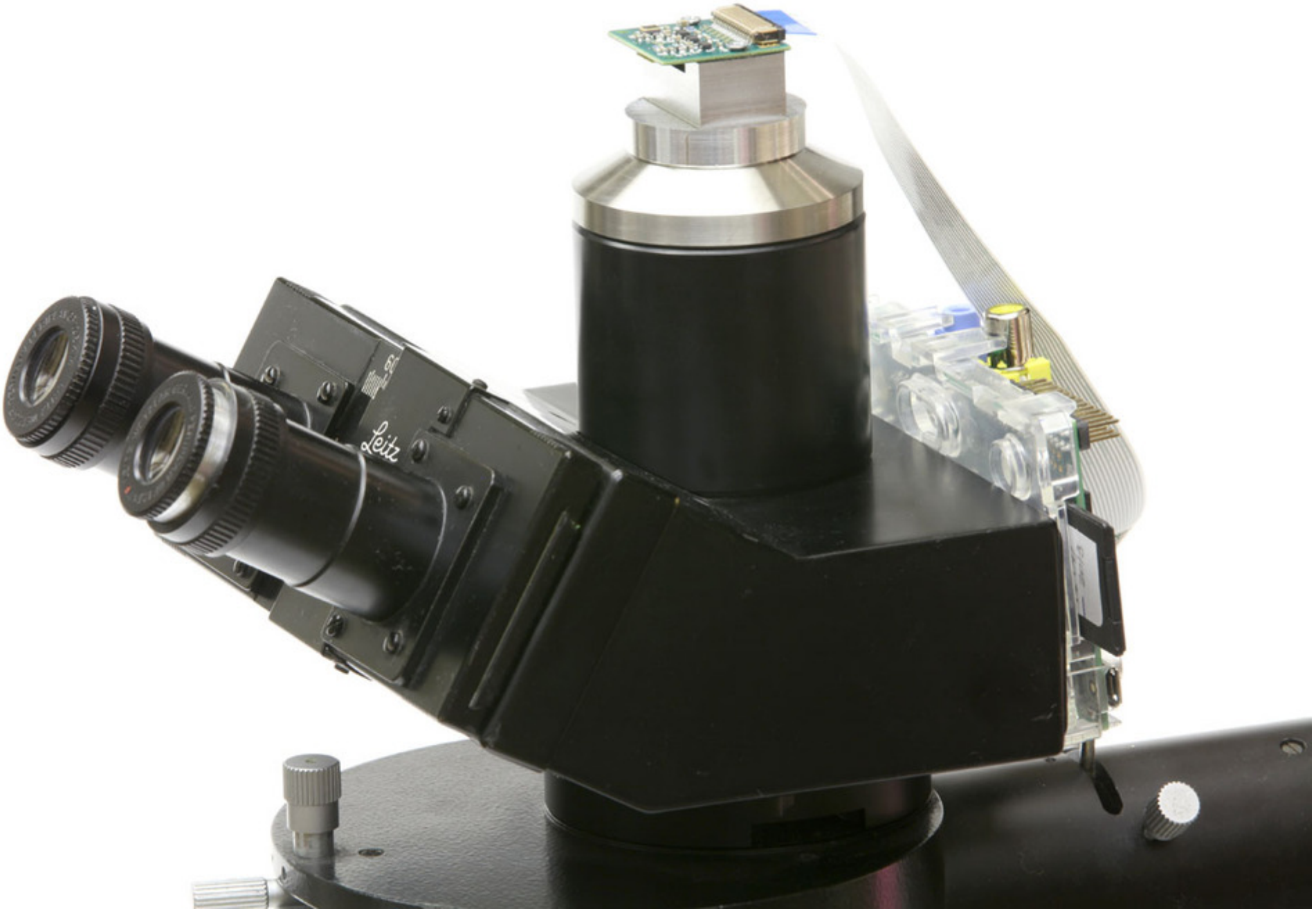


http://www.modmypi.com/

惡改鏡頭

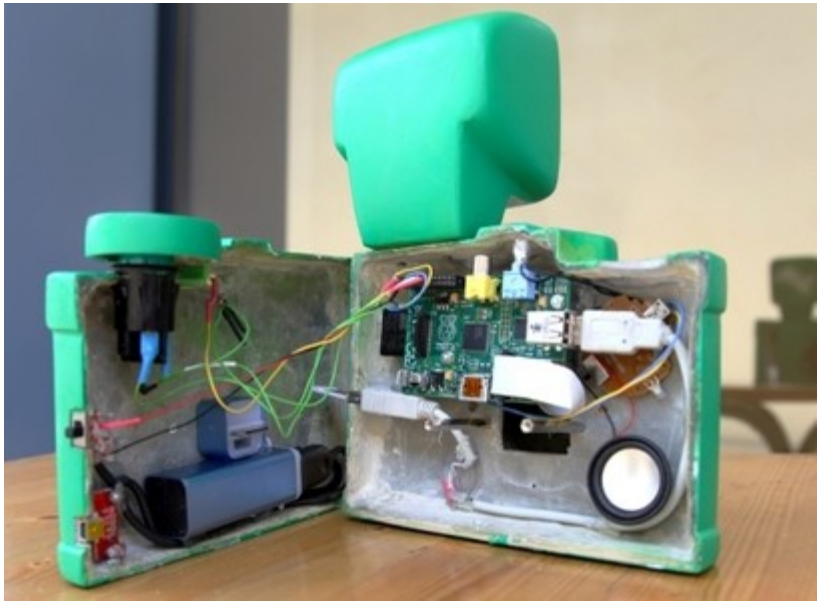


http://www.truetex.com/raspberrypi



http://www.truetex.com/raspberrypi

外殼改裝

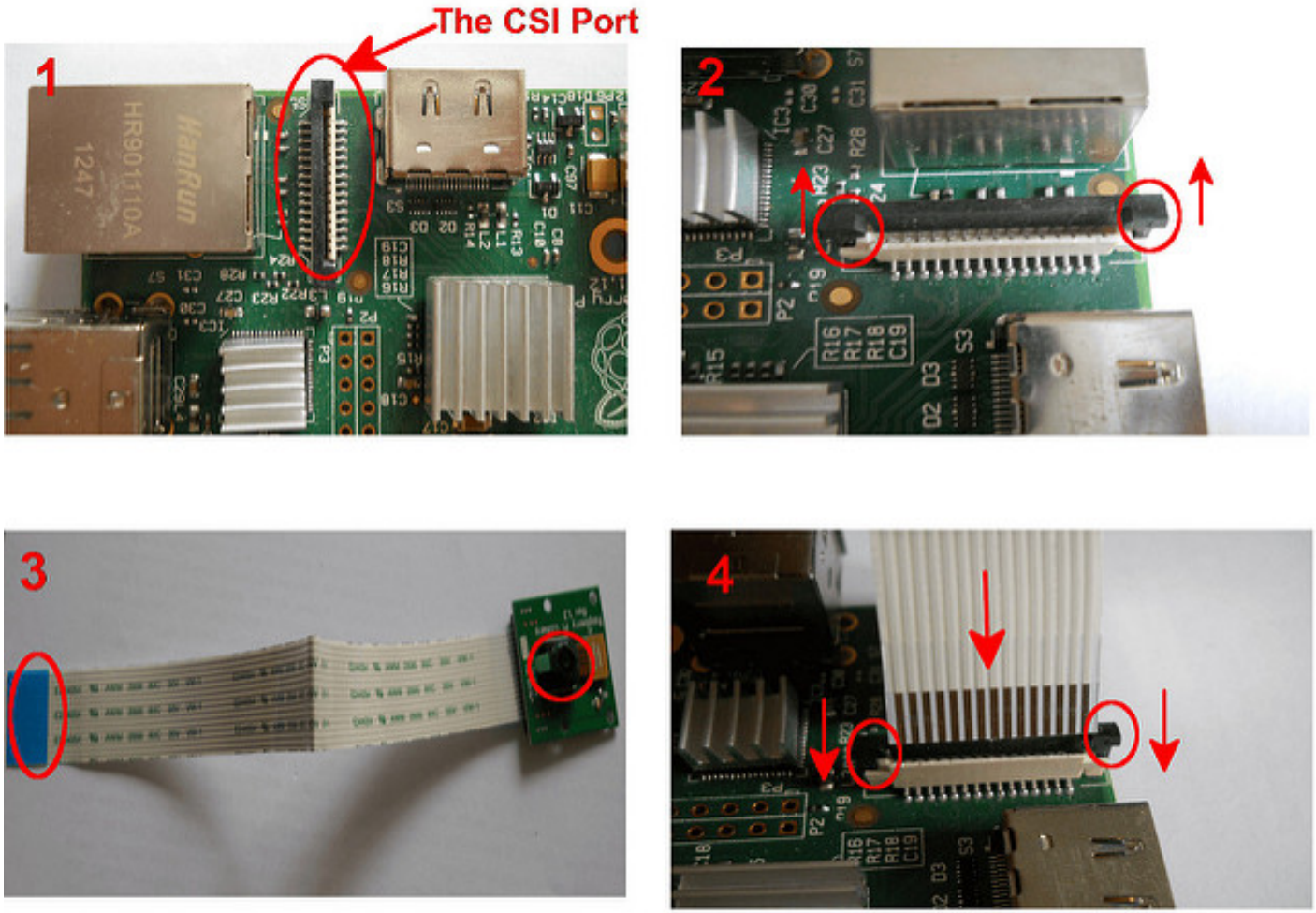


http://www.modmypi.com/
http://blog.pi3g.com/2013/11/coming-soon-raspberry-with-case-mounted-camera/

Camera 安装

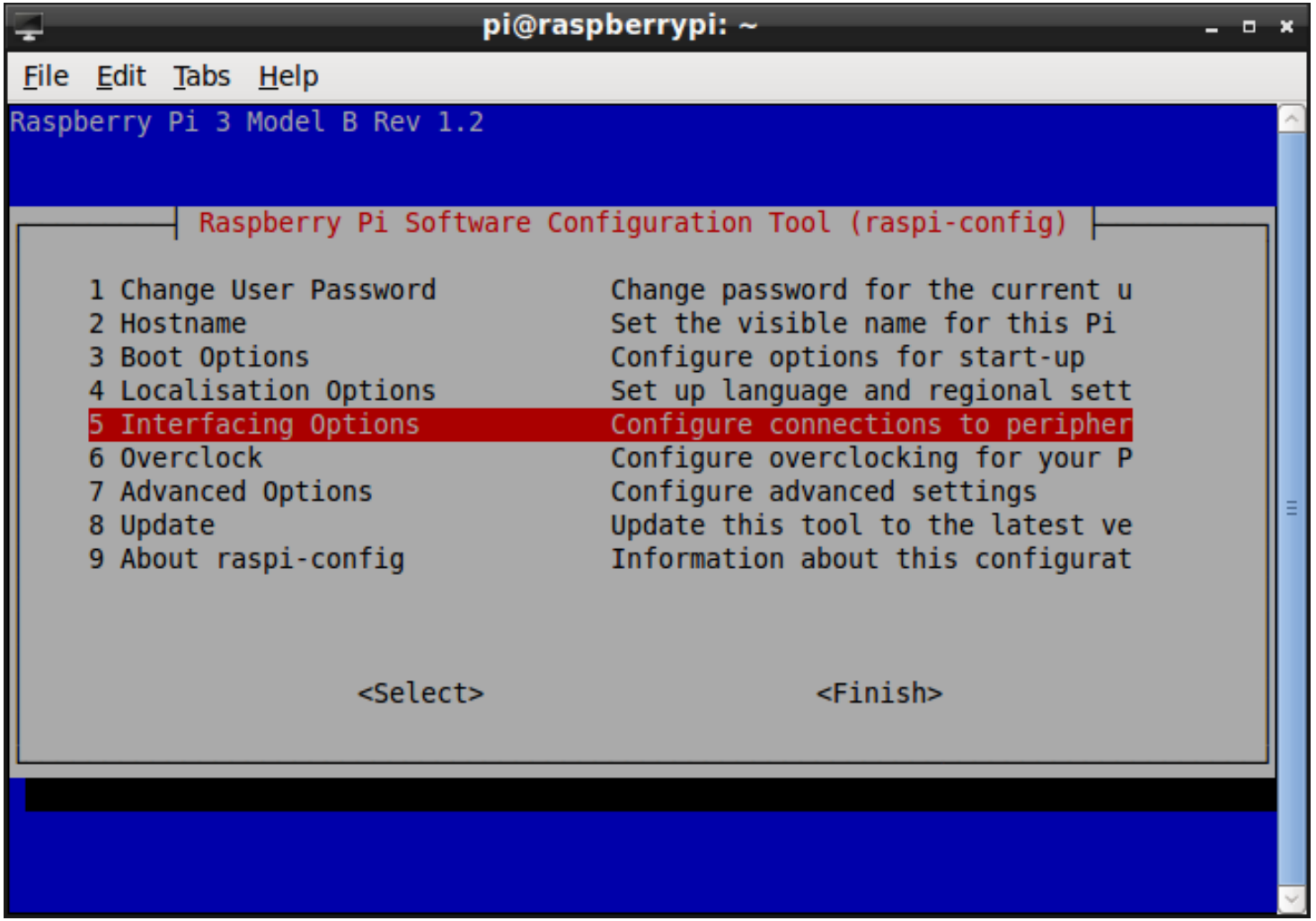
在關機的狀態下安裝 Camera

關機指令： $ sudo poweroff

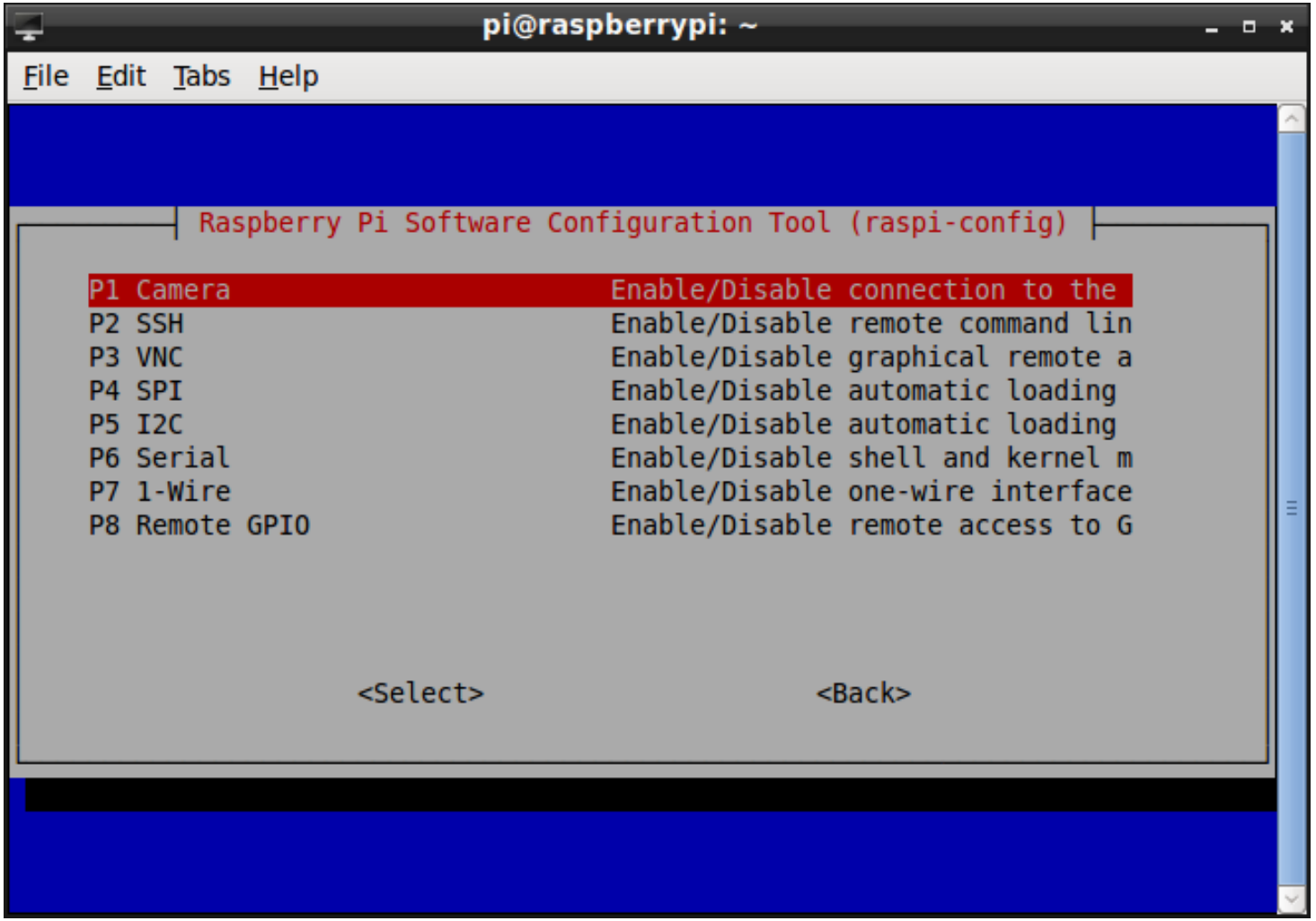


http://xmodulo.com/install-raspberry-pi-camera-board.html

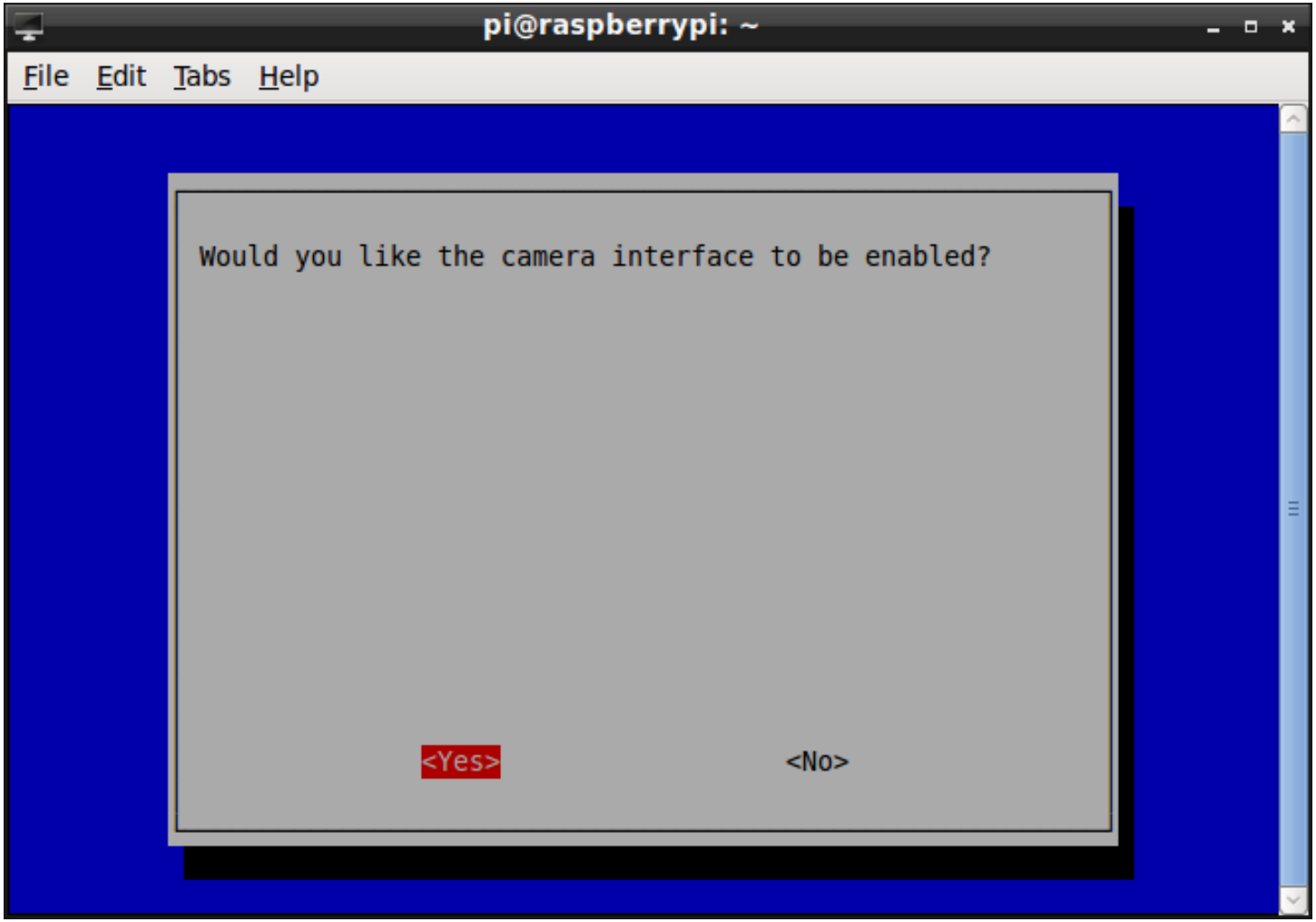
$ sudo raspi-config



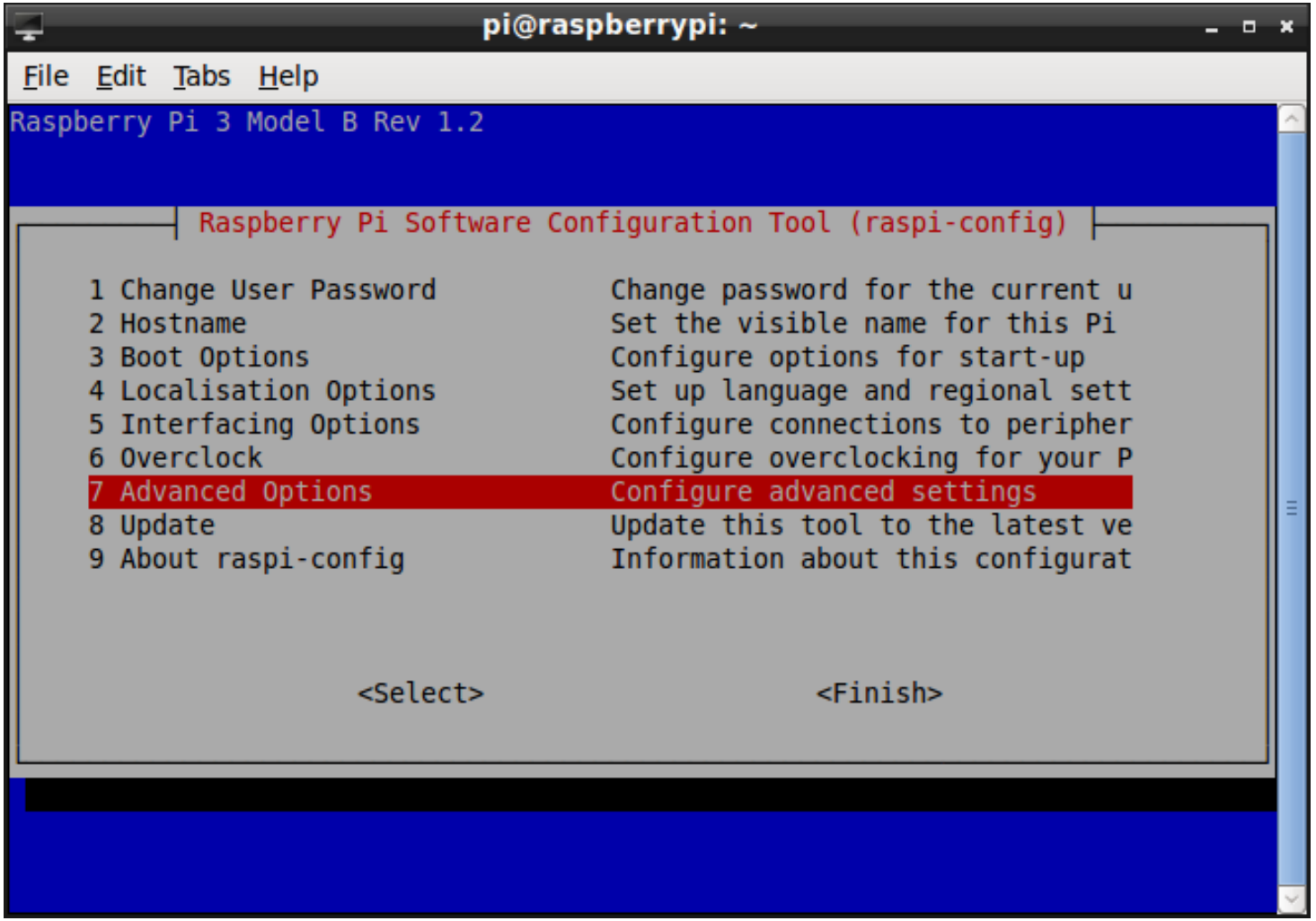
選擇 Camera


pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera Enable/Disable connection to the
P2 SSH Enable/Disable remote command lin
P3 VNC Enable/Disable graphical remote a
P4 SPI Enable/Disable automatic loading
P5 I2C Enable/Disable automatic loading
P6 Serial Enable/Disable shell and kernel m
P7 1-Wire Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to G
<Select>
<Back>

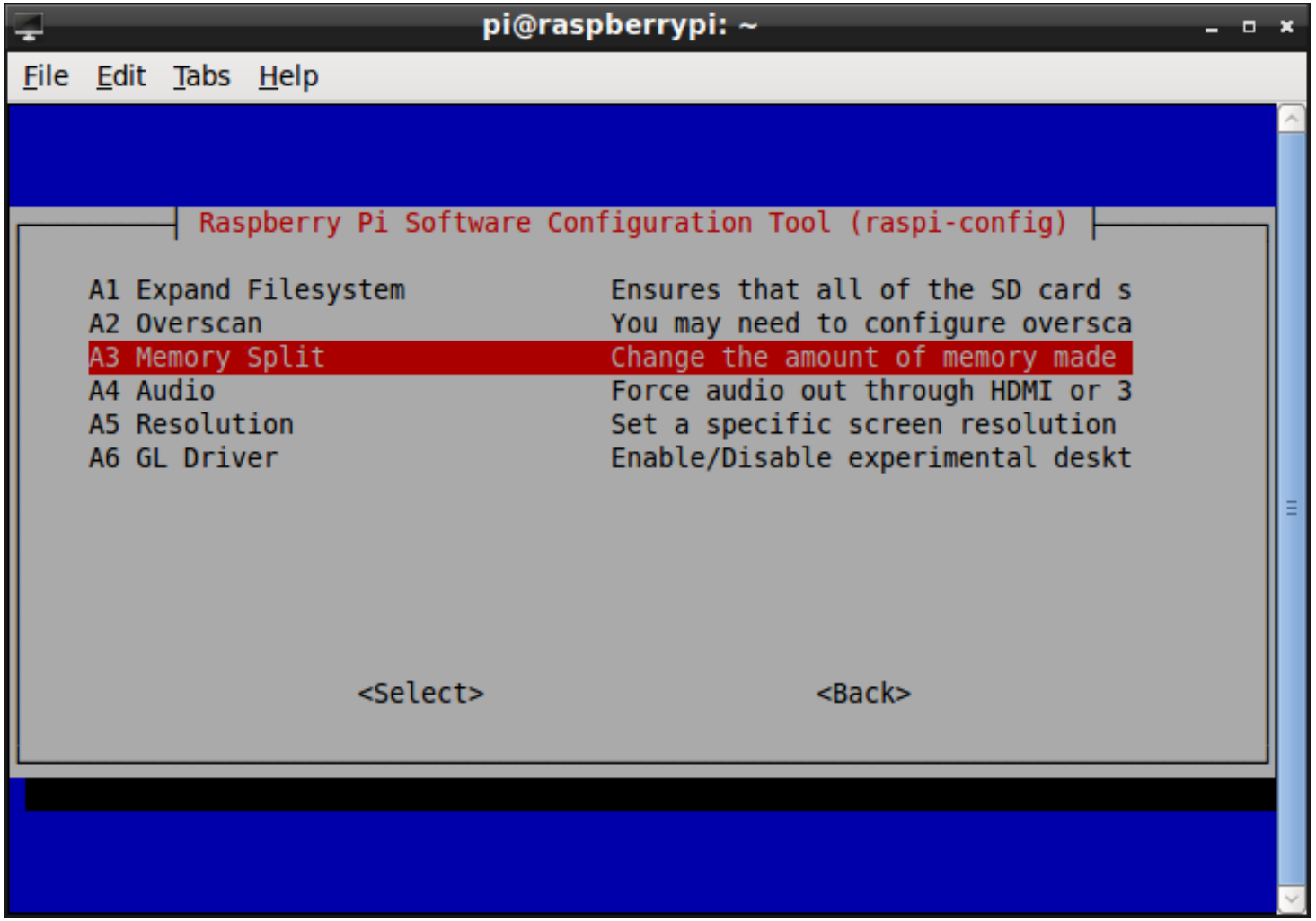
啟用 Camera


pi@raspberrypi: ~
File Edit Tabs Help
Would you like the camera interface to be enabled?
<Yes>
<No>

進階選項


pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi 3 Model B Rev 1.2
Raspberry Pi Software Configuration Tool (raspi-config)
1 Change User Password Change password for the current u
2 Hostname Set the visible name for this Pi
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional sett
5 Interfacing Options Configure connections to peripher
6 Overclock Configure overclocking for your P
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest ve
9 About raspi-config Information about this configurat
<Select>
<Finish>

設定記憶體分配


pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi Software Configuration Tool (raspi-config)
A1 Expand Filesystem
Ensures that all of the SD card s
A2 Overscan
You may need to configure oversca
A3 Memory Split
Change the amount of memory made
A4 Audio
Force audio out through HDMI or 3
A5 Resolution
Set a specific screen resolution
A6 GL Driver
Enable/Disable experimental deskt
<Select>
<Back>

要大於等於 128M


pi@raspberrypi: ~
File Edit Tabs Help
How much memory (MB) should the GPU have? e.g. 16/32/64/128/256
128
設定完後重開機生效
<Ok>
<Cancel>

實戰 Camera 使用

使用 Camera 前先消除靜電吧

http://www.wikihow.com/Remove-Static-Electricity

實驗 1:Hello Camera

目的：練習照相和攝影的指令

拍照指令 RaspiStill

- 只預覽 2 秒 (-t), 不存檔
  - `$ raspistill -t 2000`

- 5 秒後拍照 (預設), 檔案 test.jpg(-o)
  - `$ raspistill -o test.jpg`

- 3 秒後拍照，並編碼成 png 格式 (-e), 長 640x 寬 480
  - `$ raspistill -t 3000 -o test.png -e png -w 640 -h 480`

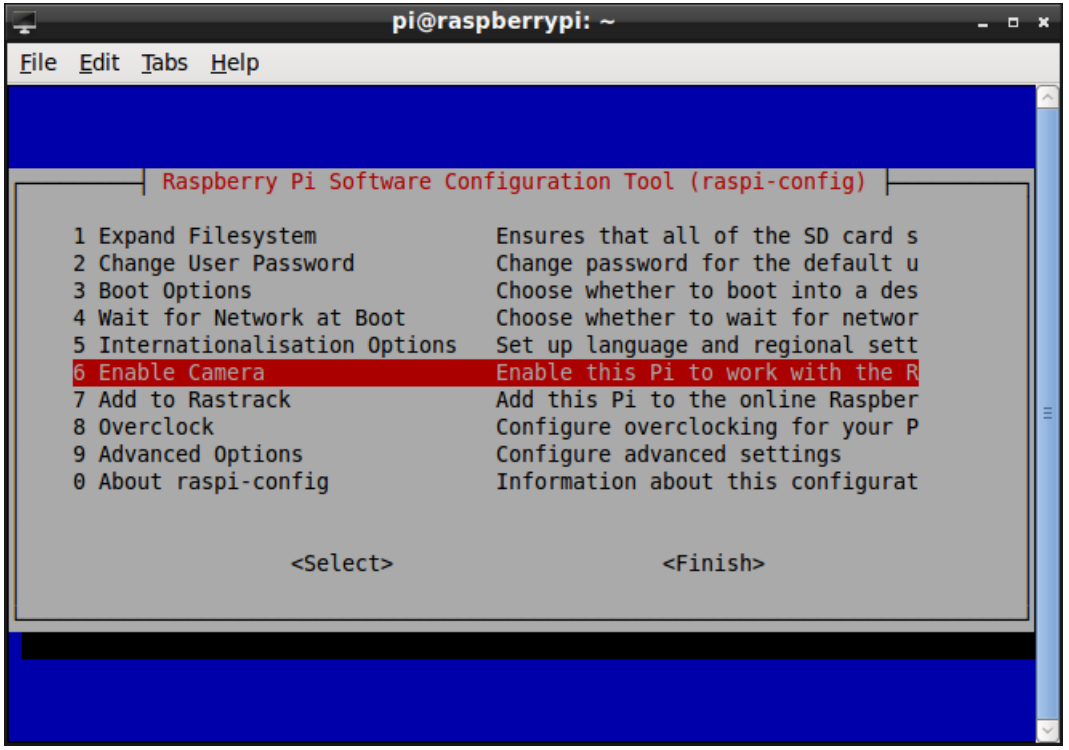
https://www.raspberrypi.org/documentation/raspbian/applications/camera.md

常見 Camera 問題？

- 訊息：Camera is not enabled in this build

```
mmal: mmal_vc_component_create: failed to create component 'vc.ril.camera' (1:ENOMEM)
mmal: mmal_component_create_core: could not create component 'vc.ril.camera' (1)
mmal: Failed to create camera component
mmal: main: Failed to create camera component
mmal: Camera is not enabled in this build. Try running "sudo raspi-config" and ensure that "camera" has been enabled
```

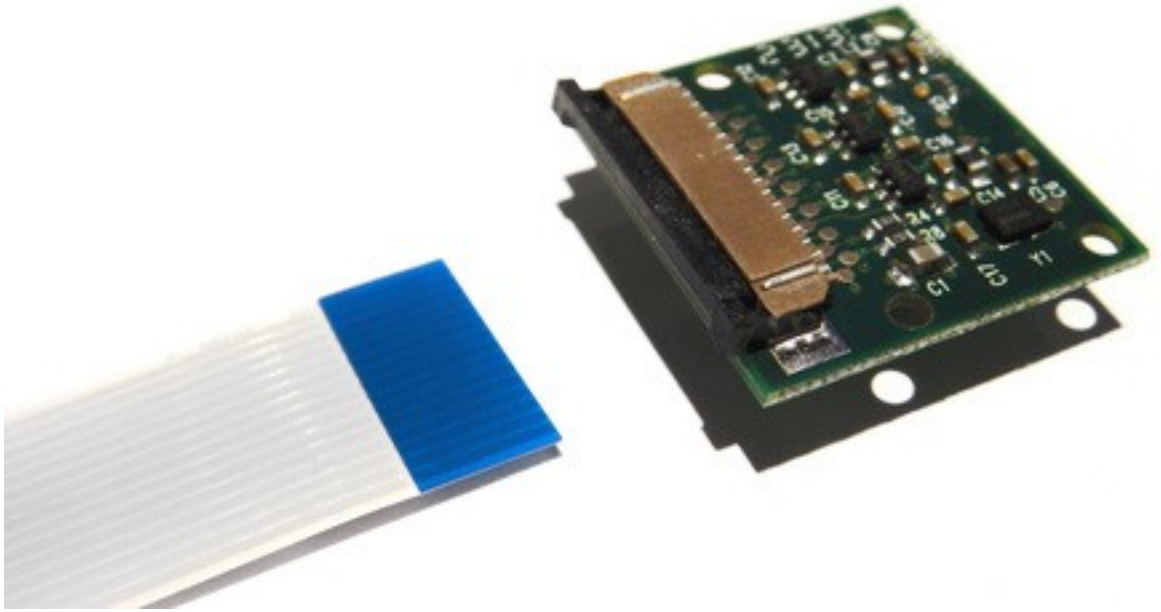
- 解法：進 raspi-config 重新 enable camera
  - $ sudo raspi-config



- 訊息：Camera is not detected

```
mmal: mmal_vc_component_create: failed to create component 'vc.ril.camera' (1:ENOMEM)
mmal: mmal_component_create_core: could not create component 'vc.ril.camera' (1)
mmal: Failed to create camera component
mmal: main: Failed to create camera component
mmal: Camera is not detected. Please check carefully the camera module is installed correctly
```

- 解法：重新安裝 camera，或是更換排線
  或是檢查 camera module 是否鬆脫



是否有鬆脫？

https://www.modmypi.com/blog/how-to-replace-the-raspberry-pi-camera-cable

錄影指令 RaspiVid

- 錄 5 秒 (-t) 1080p30 影片 (預設 w/h = 1920/1080)
  - $ raspivid -t 5000 -o video.h264

- 錄 5 秒的 1080p30 影片， 長 640x 寬 480
  - $ raspivid -t 5000 -w 640 -h 480 -o video.h264

https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspivid.md

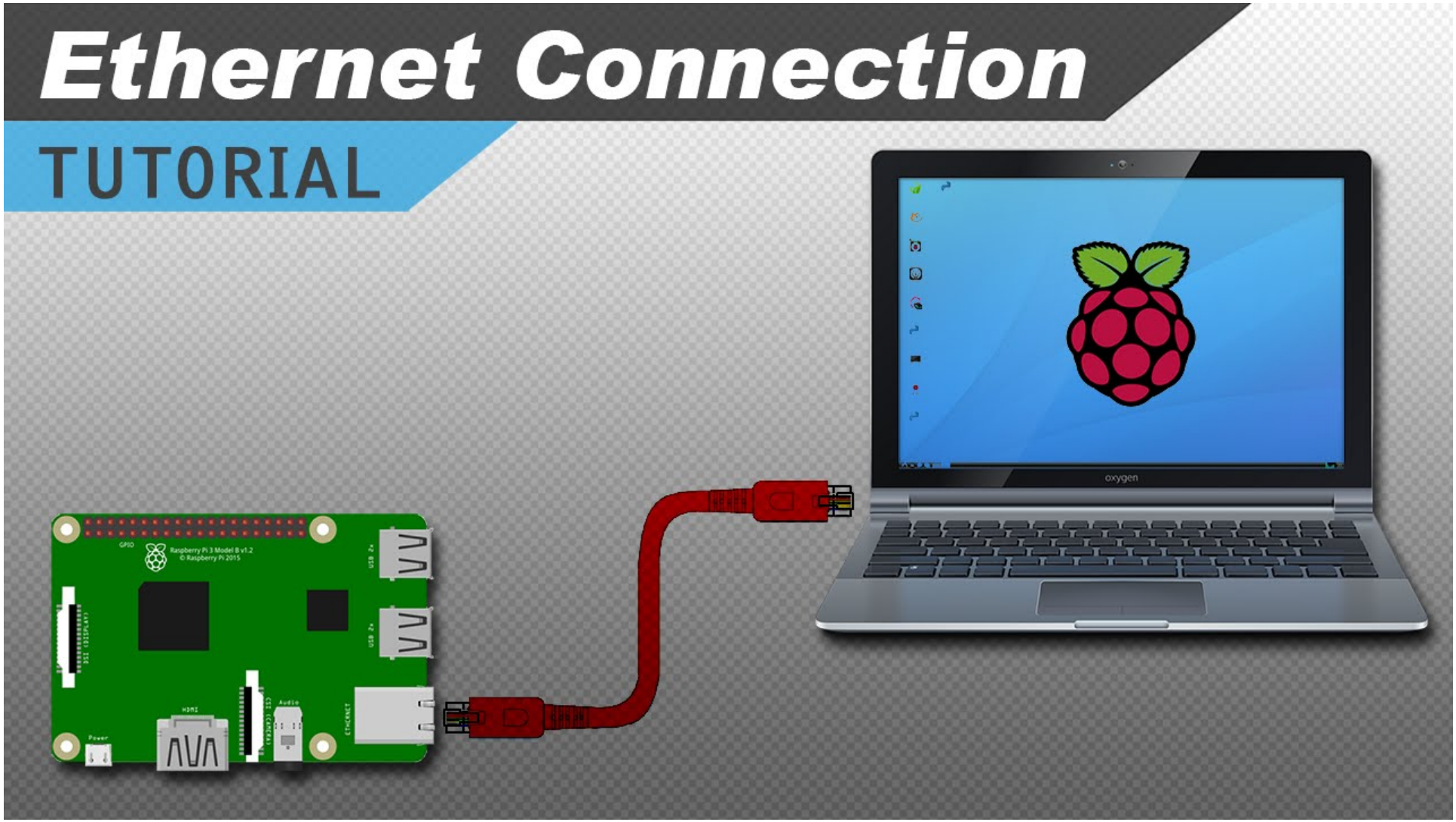
更多參數或用法請看文件

http://goo.gl/V4klcZ

https://github.com/raspberrypi/documentation/blob/master/raspbian/applications/camera.md

如何看照片和影片？

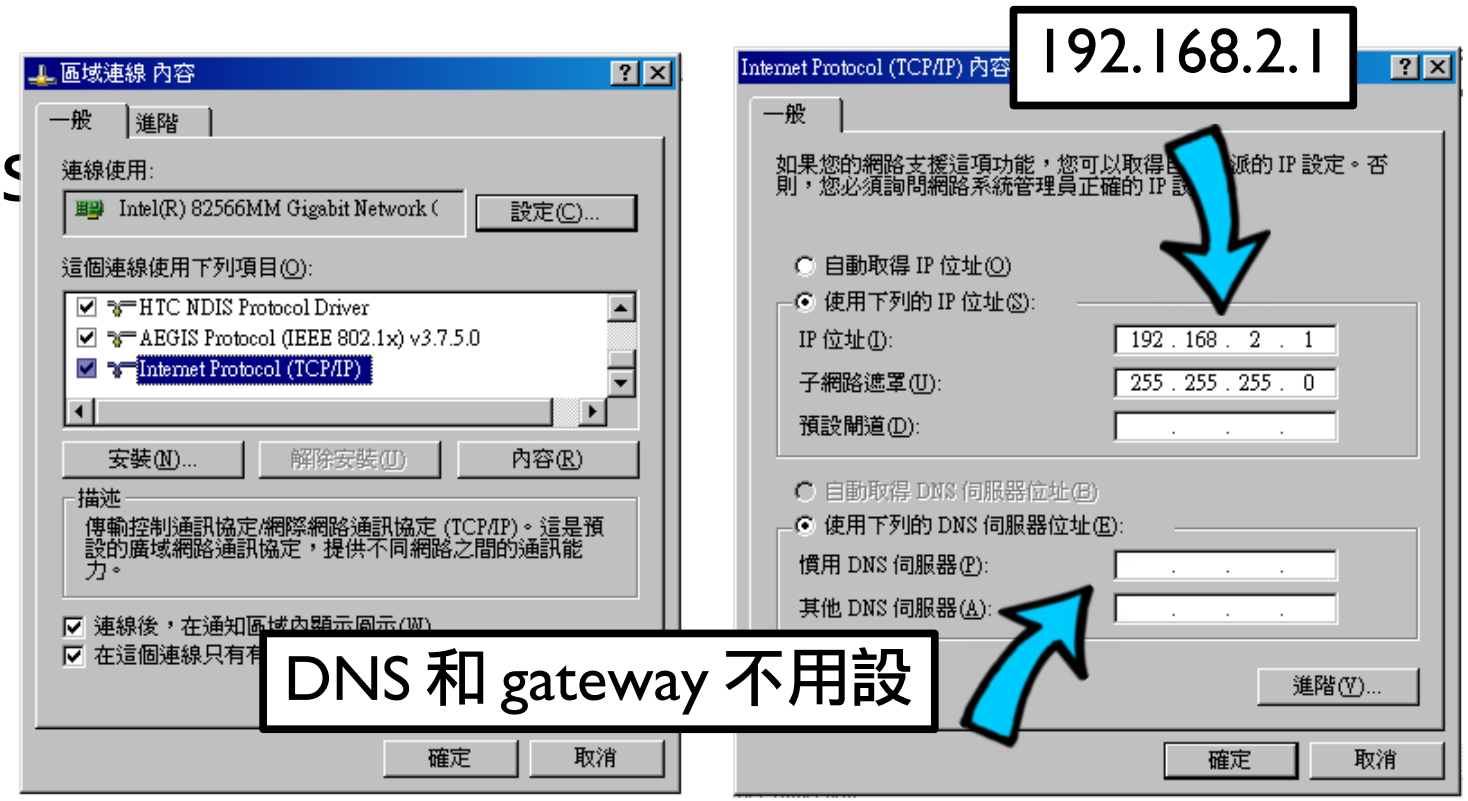
用網路線對接



https://www.youtube.com/watch?v=5DCPDQnRXm8

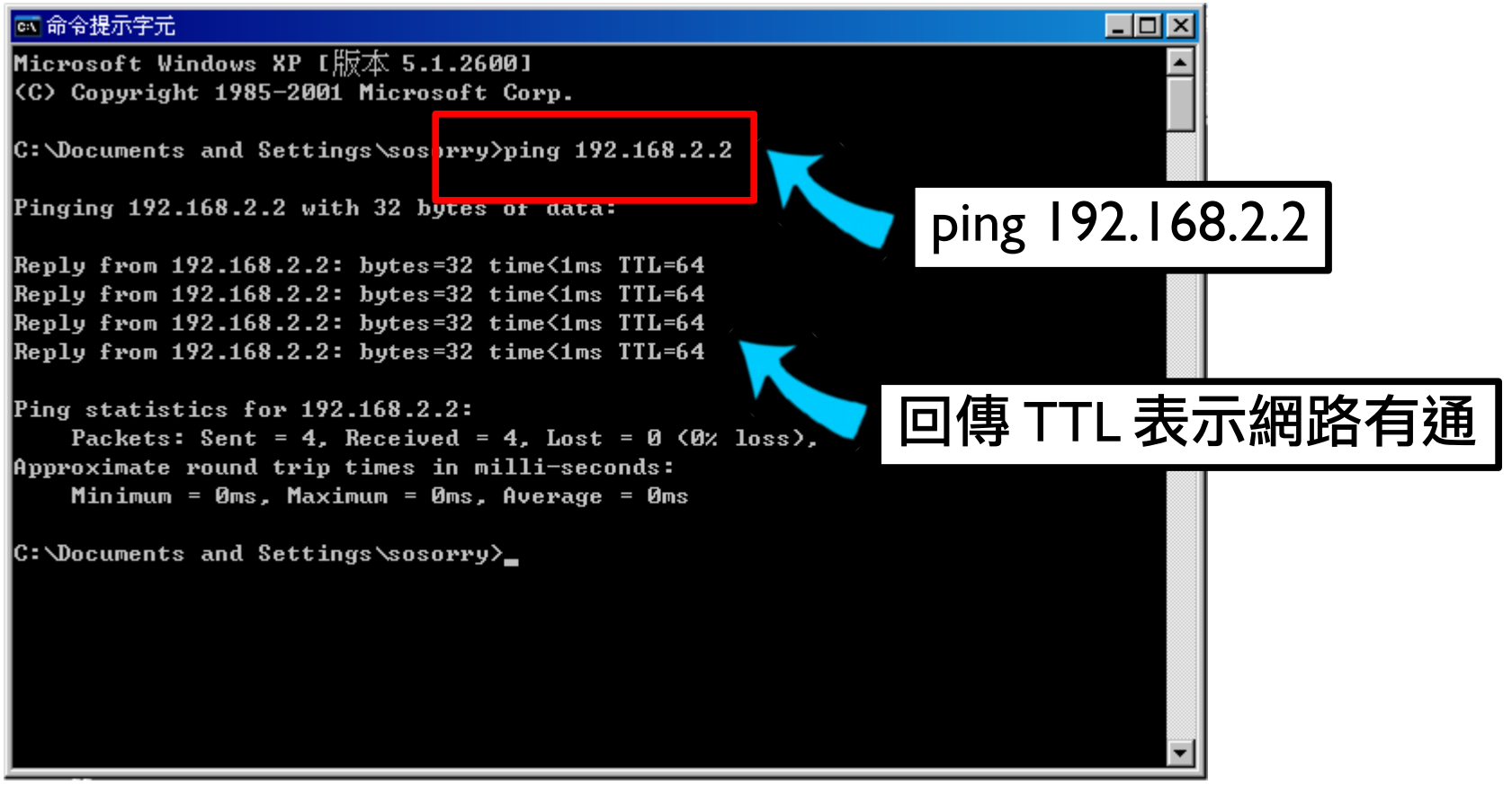
網路設定

- 把 Pi 設成 .2.2
  - $ sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.0
- 把 Windows 設成 .2.1



確認網路是否有通？

- 在 Windows 執行 cmd
- 輸入 ping 192.168.2.2 看是否有回應？

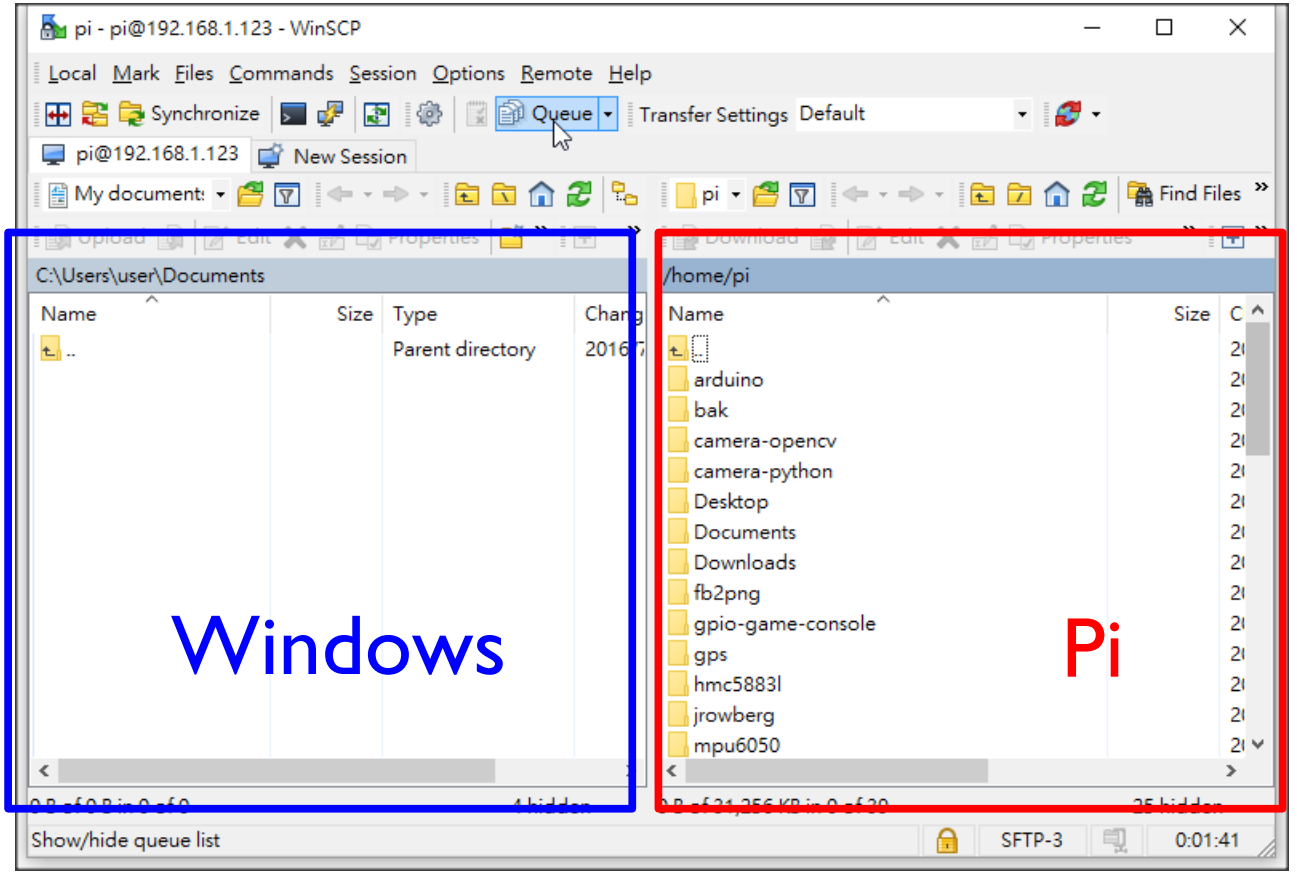


方法一：將 Pi 的檔案傳回本機端

在 Windows 上安裝 WinSCP

- 下載網址

  `http://winscp.net/eng/download.php`

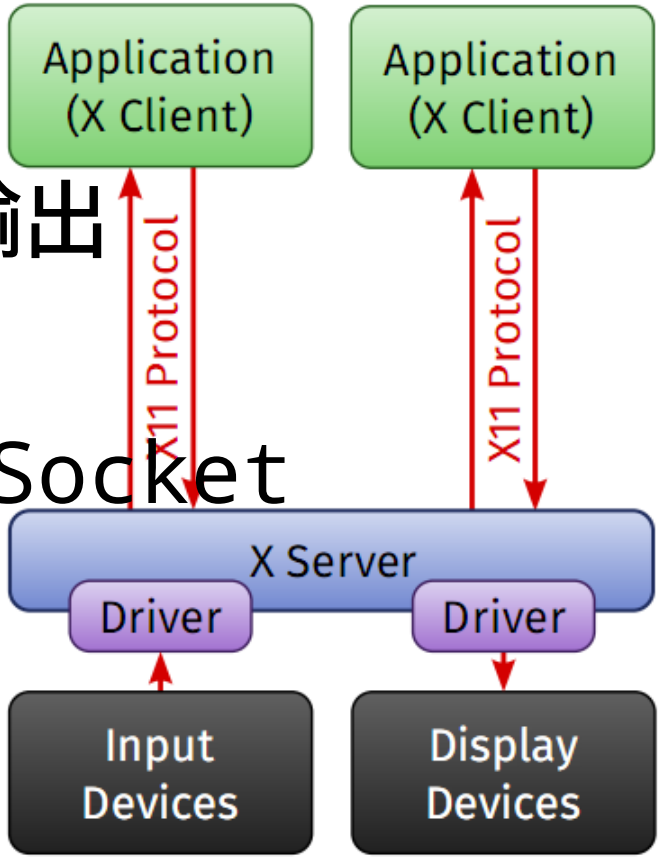


http://winscp.net/

方法二：使用 X11 Forwarding

X Window System

- 是一種圖形應用標準
- Client/Server 架構
  - X Client: 應用程式
  - X Server: 管理硬體輸入 / 輸出
- 可透過網路傳輸
  - TCP/IP 或是 Unix Domain Socket
- X11 是通訊協定名稱



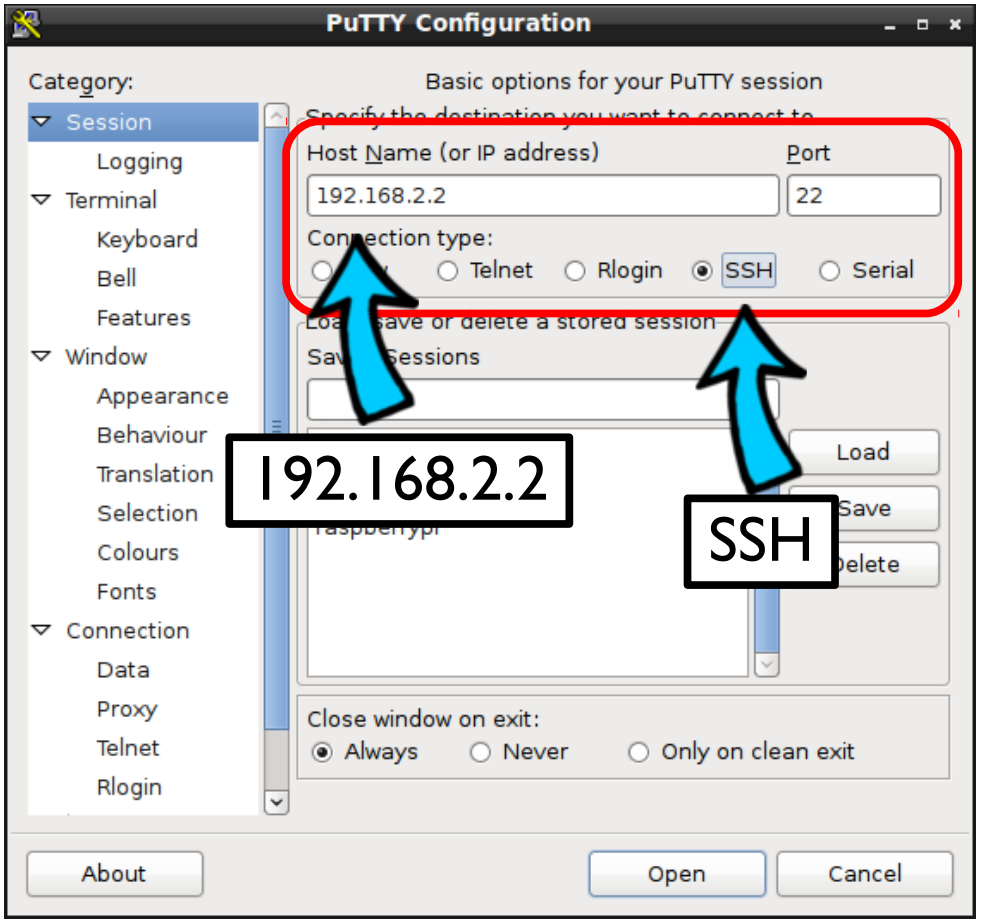
http://keyj.emphy.de/files/linuxgraphics_en.pdf

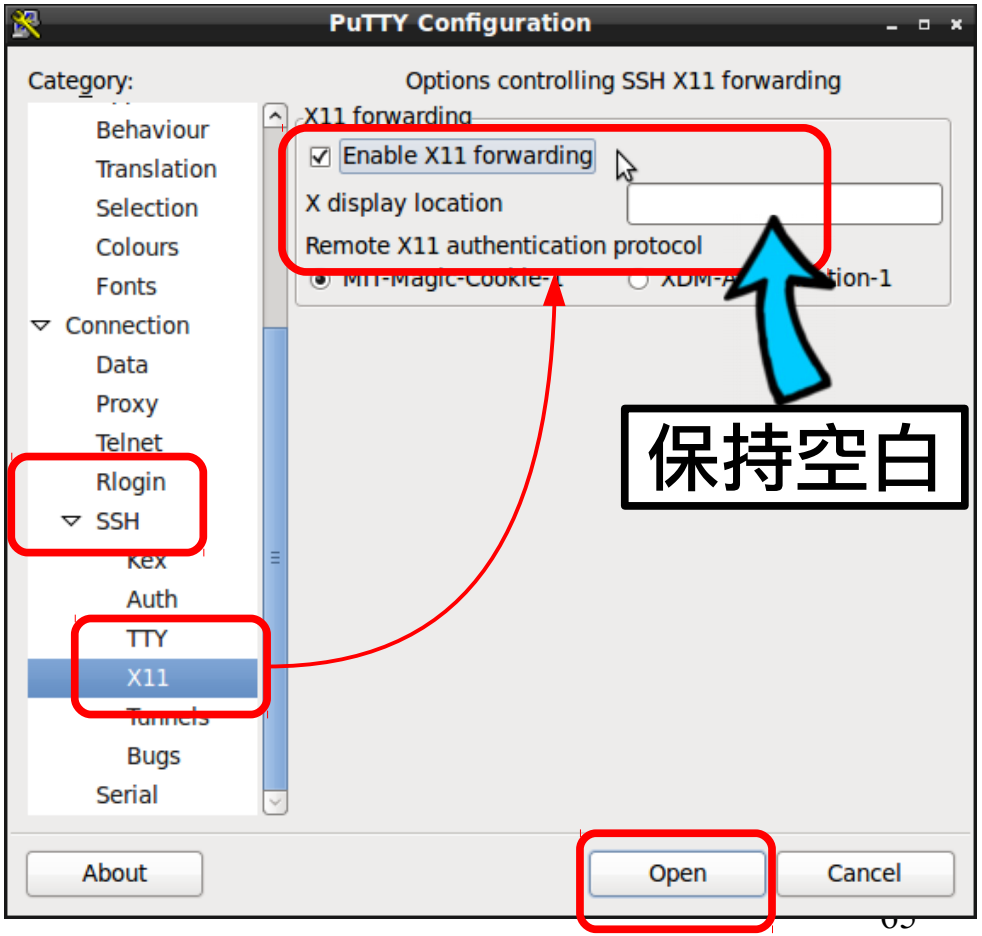
在 Windows 安裝 X Server

- 安裝 Xming，下一步到底
  http://sourceforge.net/projects/xming

在 Windows 設定 X11 Forwarding

- SSH > X11 > Enable X11 forwarding





如果是 Linux 或是 Mac OS

開啟終端機，ssh -X pi@PI 的 IP

“Can not open display” on Mac

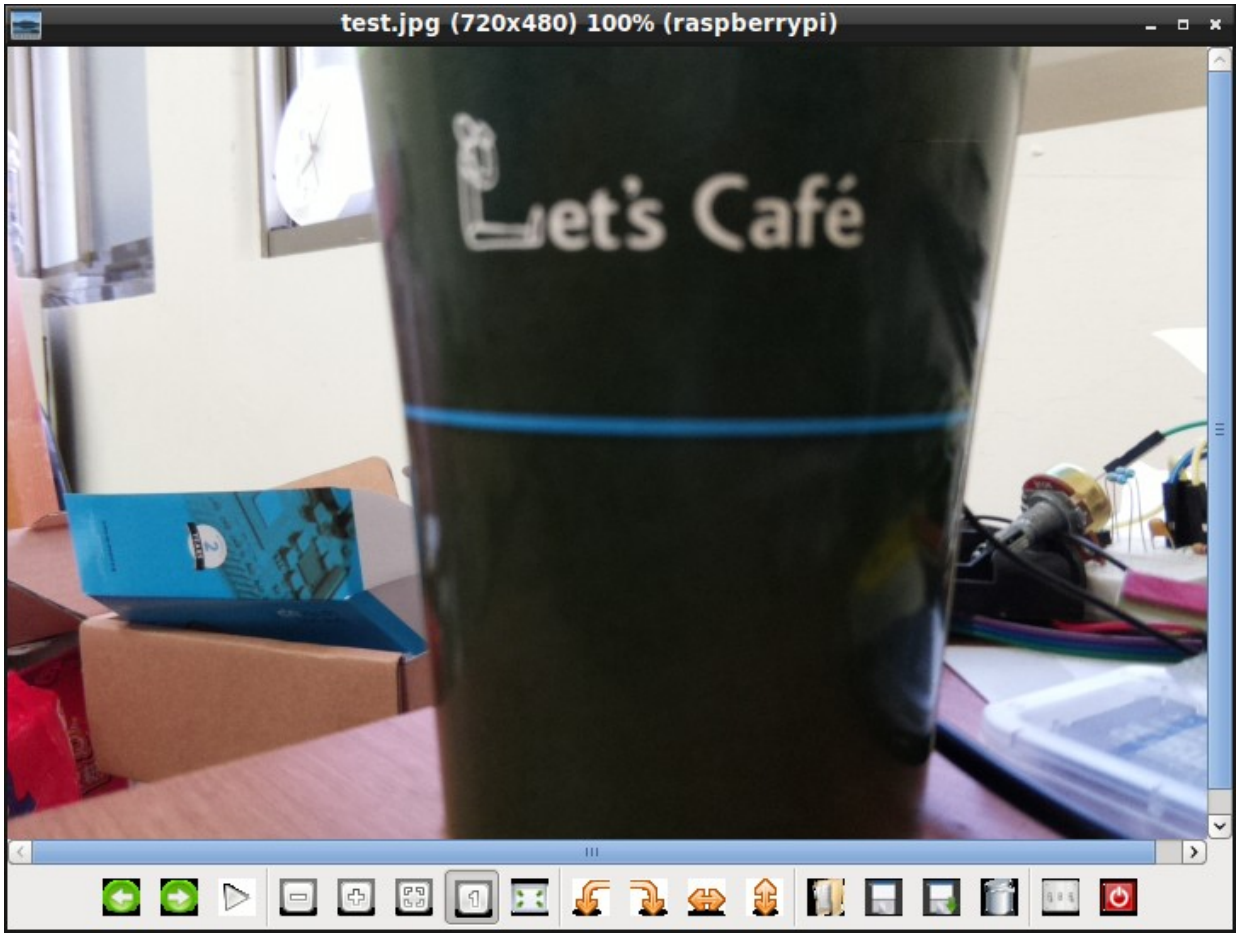
- 第一步：在 Mac 編輯 /etc/sshd_config
  (或是 /etc/ssh/sshd_config)
  修改這行 # X11Forwarding no
  把 no 改成 yes 並且把註解拿掉
- 第二步：下載安裝 XQuartz 並重開機
  http://xquartz.macosforge.org/landing/

- 感謝 Dami 和 YUN-TAO CHEN 的貢獻

https://hackpad.com/X11-Forwarding-FcyKHioKxmW

X11 Forwarding 連線成功後

- 看照片
  - `$ gpicview test.jpg`



X11 Forwarding FAQ

- Q: 什麼時候可以使用 X11 Forwarding ?
- A: 當 GUI toolkit 或是 library 是架在 X 的時候
  - 例如 Qt,Gtk,Tkinter,SDL 等等適合
  - Framebuffer,GPU 等直接輸出不能使用

- Q: 什麼時候要使用 sudo 什麼時候不用 ?
- A: X11 Forwarding 需要認證 (authorization), 所以如果要畫面回傳時都不能使用 sudo 執行

練習

- 請查詢網頁文件，將拍照後旋轉 90 度

  http://goo.gl/V4klcZ
- 一般拍照：
  - `$ raspistill -t 2000 -o normal.jpg`
- 旋轉拍照：
  - `$ raspistill -t 2000 ? -o rotation.jpg`
- 負片效果：
  - `$ raspistill -t 2000 -ifx ? -o neg.jpg`

請找出適合的參數

請找出適合的參數

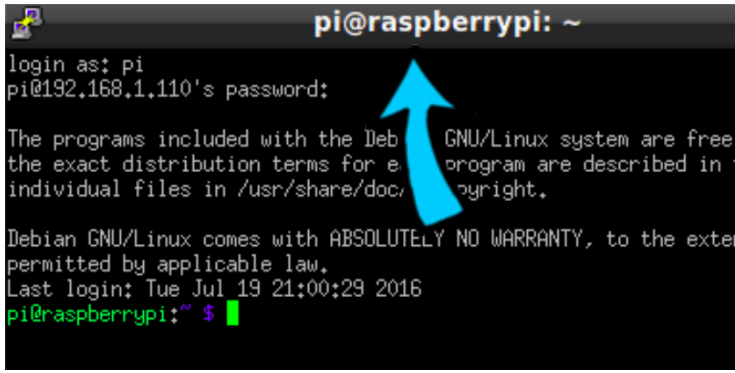
小結

- Raspberry Pi Camera 指令
  - 拍照 raspistill
  - 錄影 raspivid
- 看照片或影片
  1. 使用 scp
  2. 使用 X11 Forwarding

Serial 連線和 SSH 連線有什麼不同？

- Serial 以實體線路相連，純文字，是獨占式的連線

- SSH 是 TCP/IP 通訊協定，透過 Ethernet 或 WiFi 連線



實驗 2：寫程式控制 Camera

目的：自己的 Camera 自己做

使用picamera (Python library)

Python2 五分鐘速成

- 變數，物件，型別，註解
- 模組
- 縮排
- 迴圈
- 條件判斷
- 函式

變數，物件，型別，註解

- 動態型別(dynamic typing)

```
# 這是註解
i = 3 # 變數 i 指到數字物件 3
i = [1, 2, 3, 4, 5] # 變數 i 指到串列物件
print i[2]          # 印出串列中第三個元素
i = “abcde”         # 變數 i 指到字串物件
print i[2]          # 印出字串中第三個元素
```

模組

```
# import MODULE
import picamera

# from Module import function
from time import sleep
```

縮排

- 用縮排取代大括號
- 程式碼的區塊是用縮排分隔
- 不使用 tab，使用空白鍵
- 常見縮排為 4 個空白鍵

迴圈

- 自動迭代(iterator)

```
for i in xrange(start, stop[, step]) :
    process

for i in xrange(0, 11, 5) :
    print i
```

條件判斷

```
if condition_1 :
    process_1
elif condition_2 :
    process_2
else :
    process_3
process_4
```

函式

```
def function_name() :
    process

def function_name(param_name) :
    process

def function_name(param_name = 3) :
    process
```

兩種執行 Python 的方法

- 1. 存成檔案以後，用 python 執行
  - `$ nano test.py`
  - `$ python test.py`

- 2. 進到互動模式，可直接看輸出結果
  - `$ python`

    ```
    Python 2.7.13 (default, Jan 19 2017, 14:48:08)
    ```
  - `[GCC 6.3.0 20170124] on linux2`
  - `Type "help", "copyright", "credits" or "license" for more information.`

    ```
    >>>
    ```

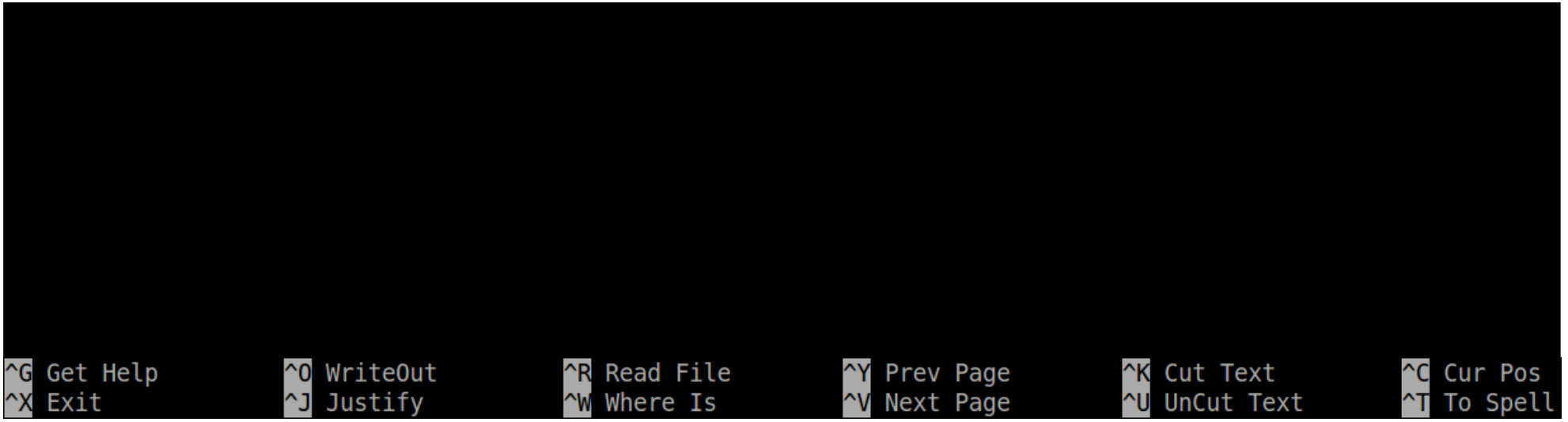
照相

```
#!/usr/bin/python
import picamera
import time

camera = picamera.PiCamera()
time.sleep(2)    # Camera warm-up time
camera.capture('test.jpg')
```

- 預設相片解析度為 720x480

http://picamera.readthedocs.io/en/release-1.13/recipes1.html

- 建立新檔案
  - $ nano <檔名，例如 take_photo.py>

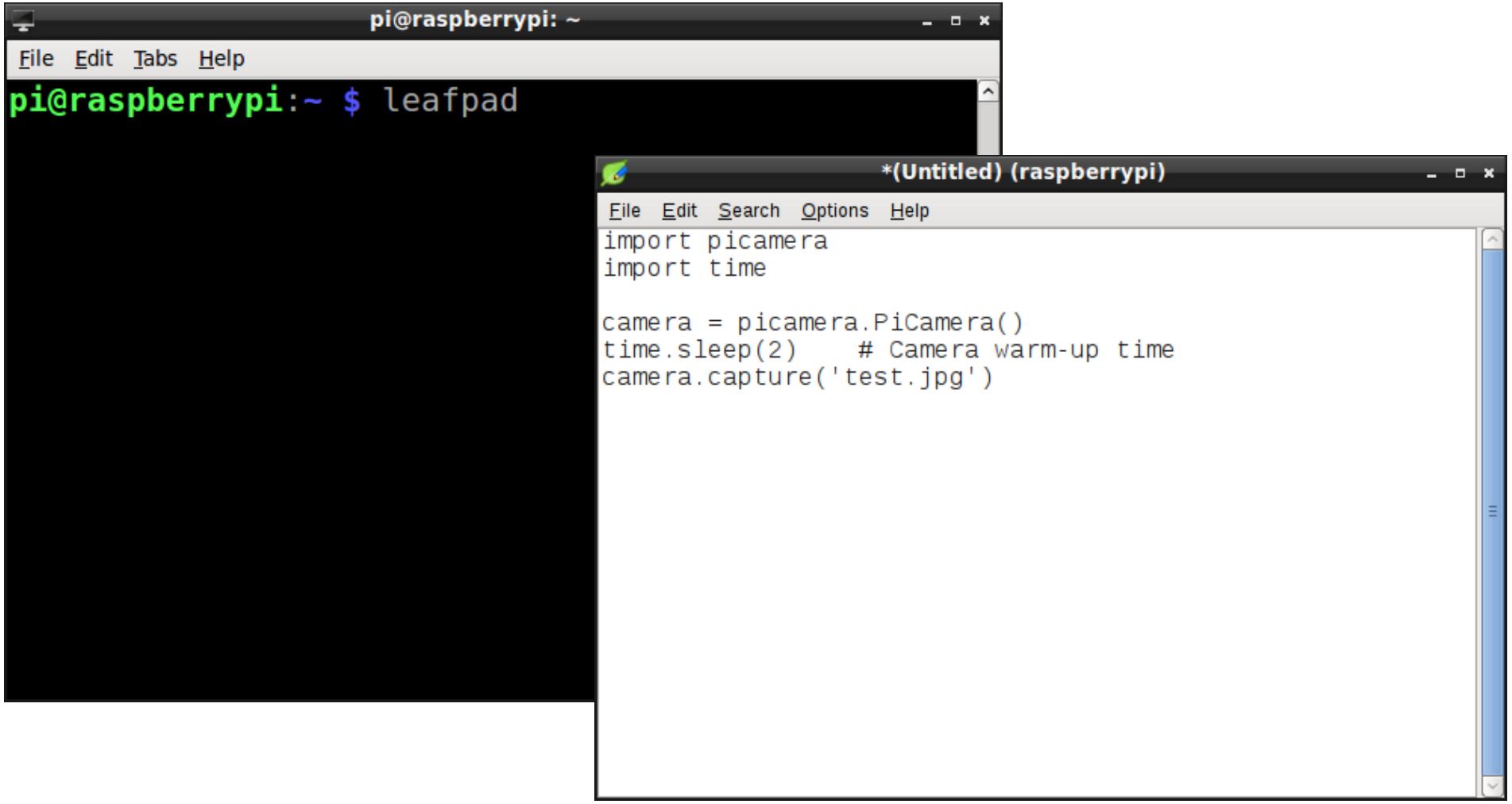


- 離開: Ctrl + x
  > 令存新檔: y
  > 不存離開: n
  > 離開: Ctrl + c

- 修改已經存在的檔案
  - $ nano <檔名，例如 /etc/rc.local>

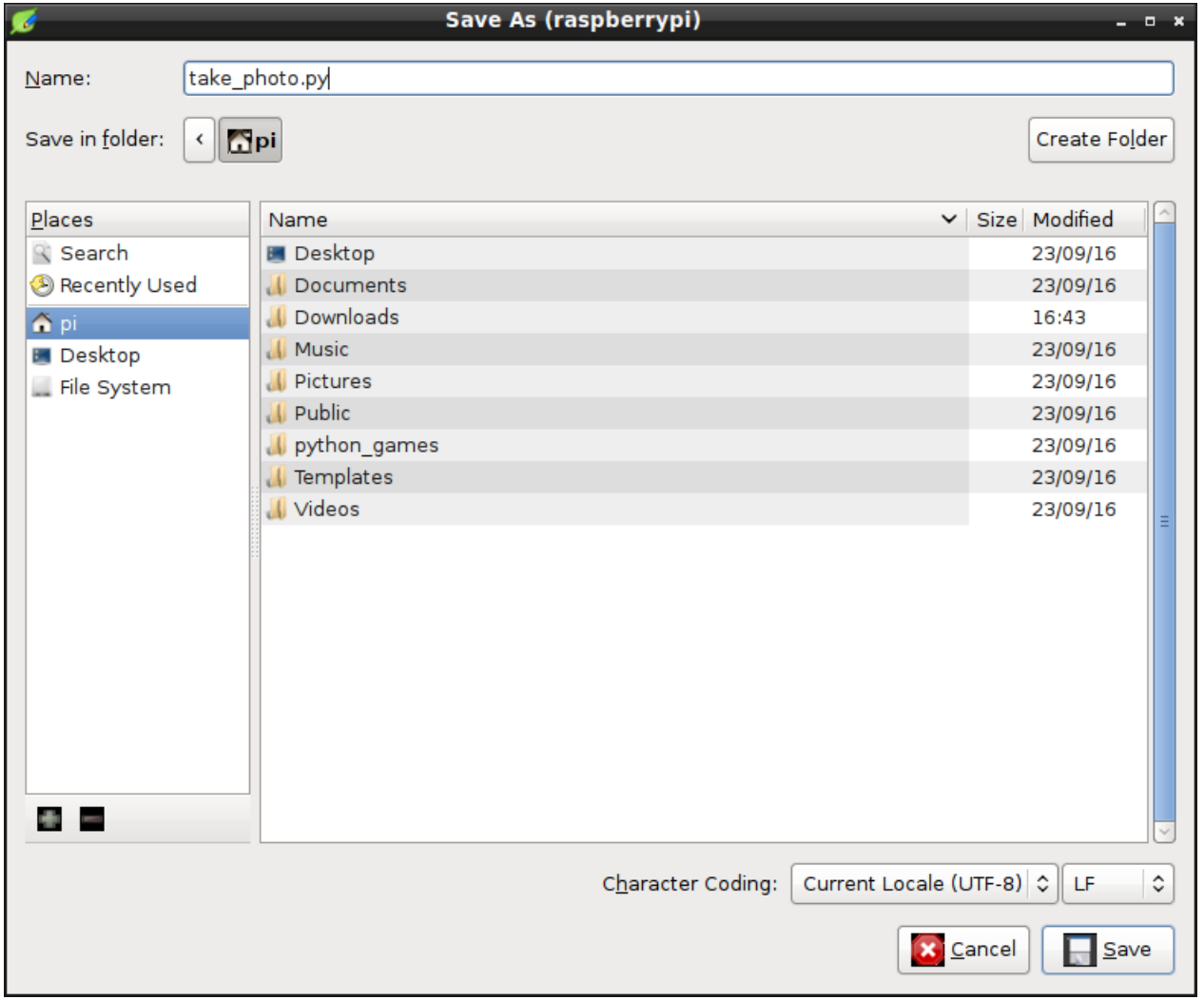
leafpad 編輯器使用

- 在 X11 forwarding 連線成功下執行 leafpad



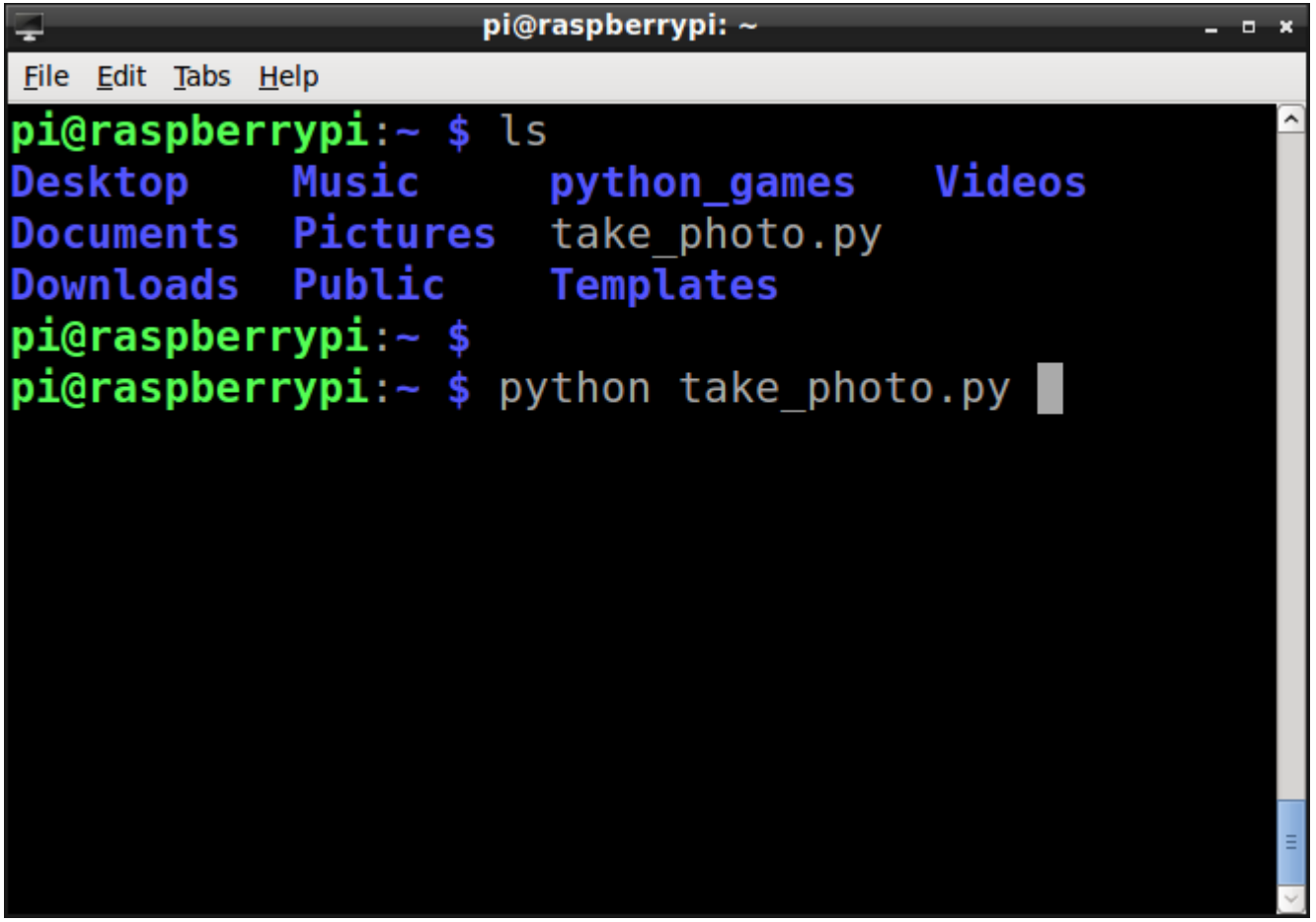
存檔

- 存檔 `take_photo.py`



執行

- `$ python take_photo.py`



DEMO
take_photo.py

```
$ cd ~/cam-py-cv/day1/02-picamera
$ python take_photo.py
```

錄影

```
#!/usr/bin/python
import picamera

camera = picamera.PiCamera()
camera.start_recording('video.h264')
camera.wait_recording(3)
camera.stop_recording()
```

- 錄 3 秒鐘影像，儲存到檔案 video.h264
- 預設錄影格式為 H.264/AVC 壓縮， 解析度 1280x800

http://picamera.readthedocs.io/en/release-1.13/recipes1.html

DEMO
record_video.py

```
$ cd ~/cam-py-cv/day1/02-picamera
$ python record_video.py
```

低光源拍照

```
import picamera
import time
from fractions import Fraction

camera = picamera.PiCamera()
camera.resolution = (640, 480)
camera.framerate = Fraction(1, 6)
camera.shutter_speed = 6000000
camera.iso = 800
time.sleep(30)
camera.exposure_mode = 'off'
camera.capture('dark.jpg')
```

http://picamera.readthedocs.io/en/release-1.13/recipes1.html

DEMO
low_light.py

```
$ cd ~/cam-py-cv/day1/02-picamera
$ python low_light.py
```

練習

- 請查詢網頁文件，在拍照後的圖片疊上文字
  http://goo.gl/2ShIrQ

- 文字包含年，月，日，分，時，秒，週
  例如 Wed Apr 19 18:37:08 2017

- 提示：疊表示 overlay

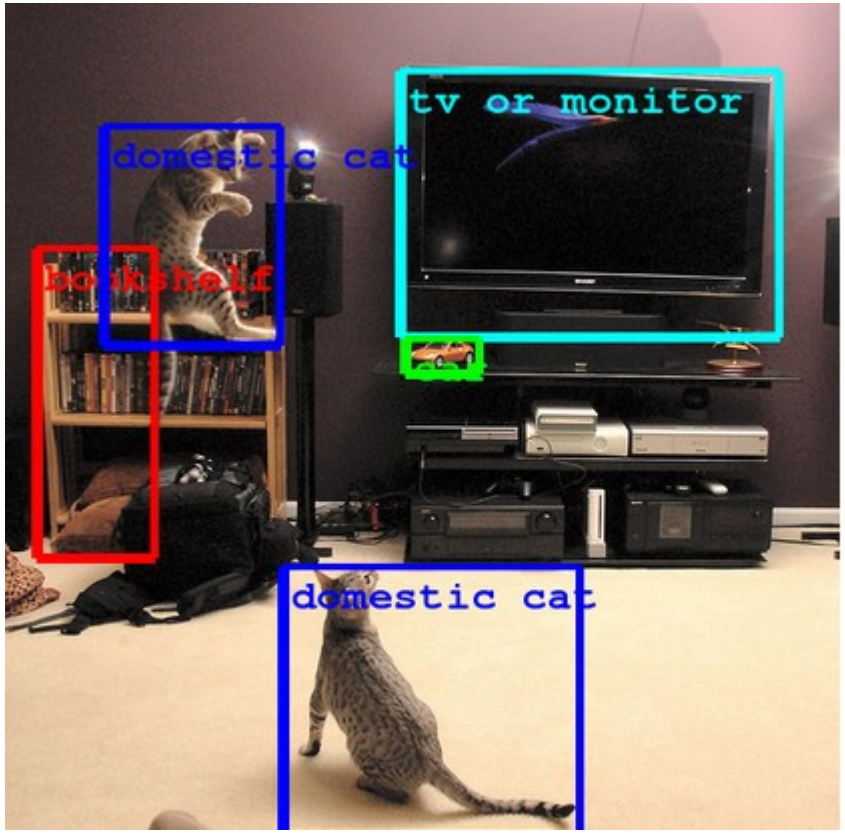
小結

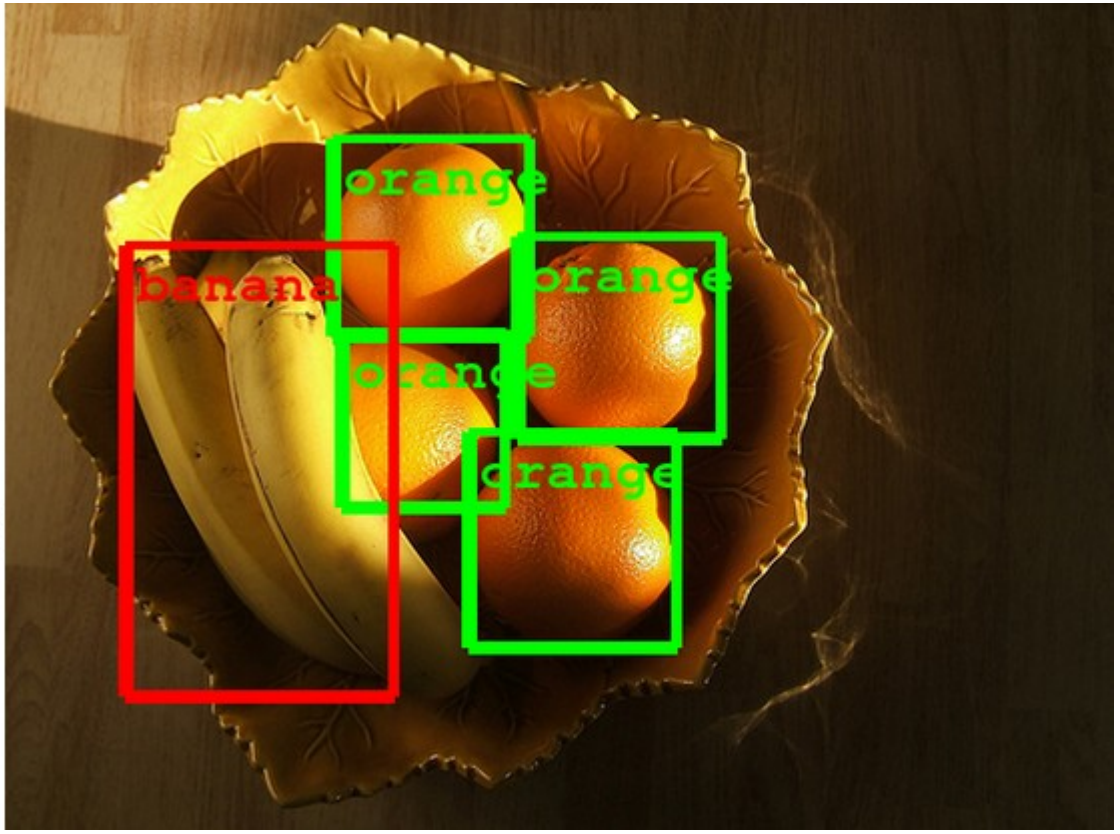
- picamera 是 Pi Camera 的 Python 套件
- 查詢網頁文件 http://goo.gl/2ShIrQ

實驗 3: 會認東西的 Camera

目的：串接網路服務

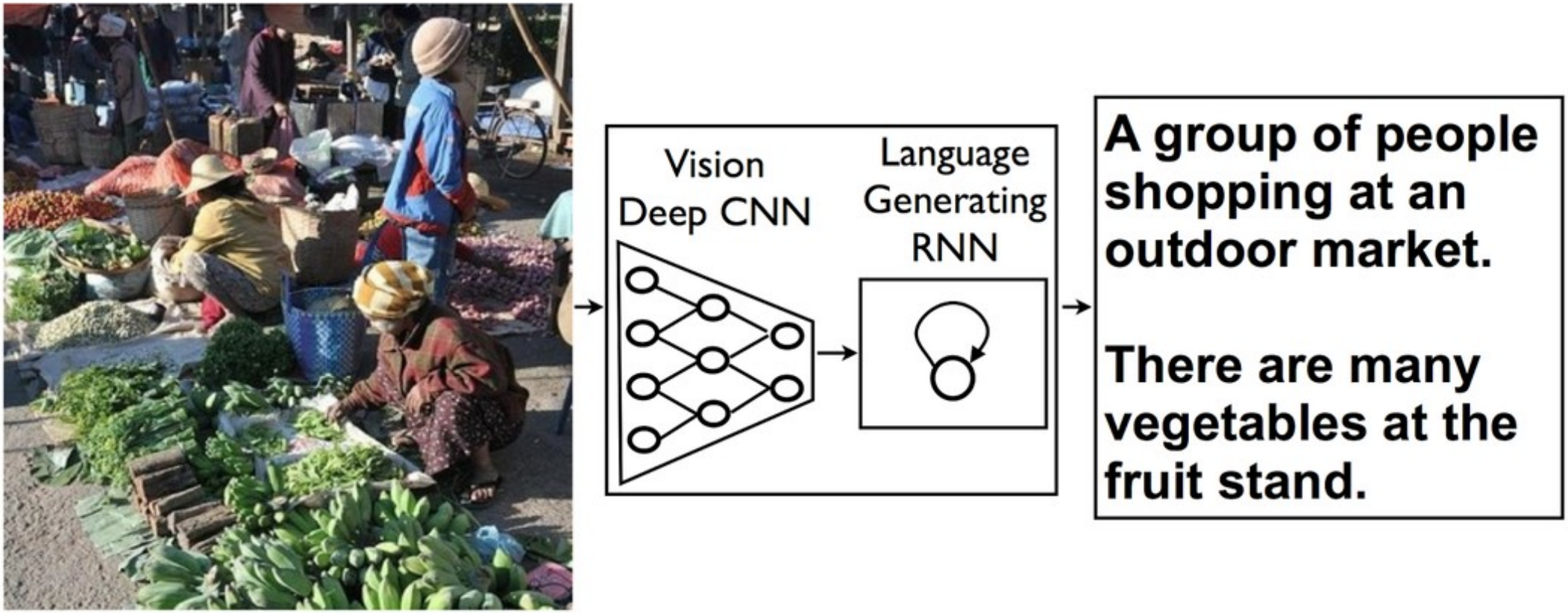
影像辨識





http://googleresearch.blogspot.tw/2014/09/building-deeper-understanding-of-images.html

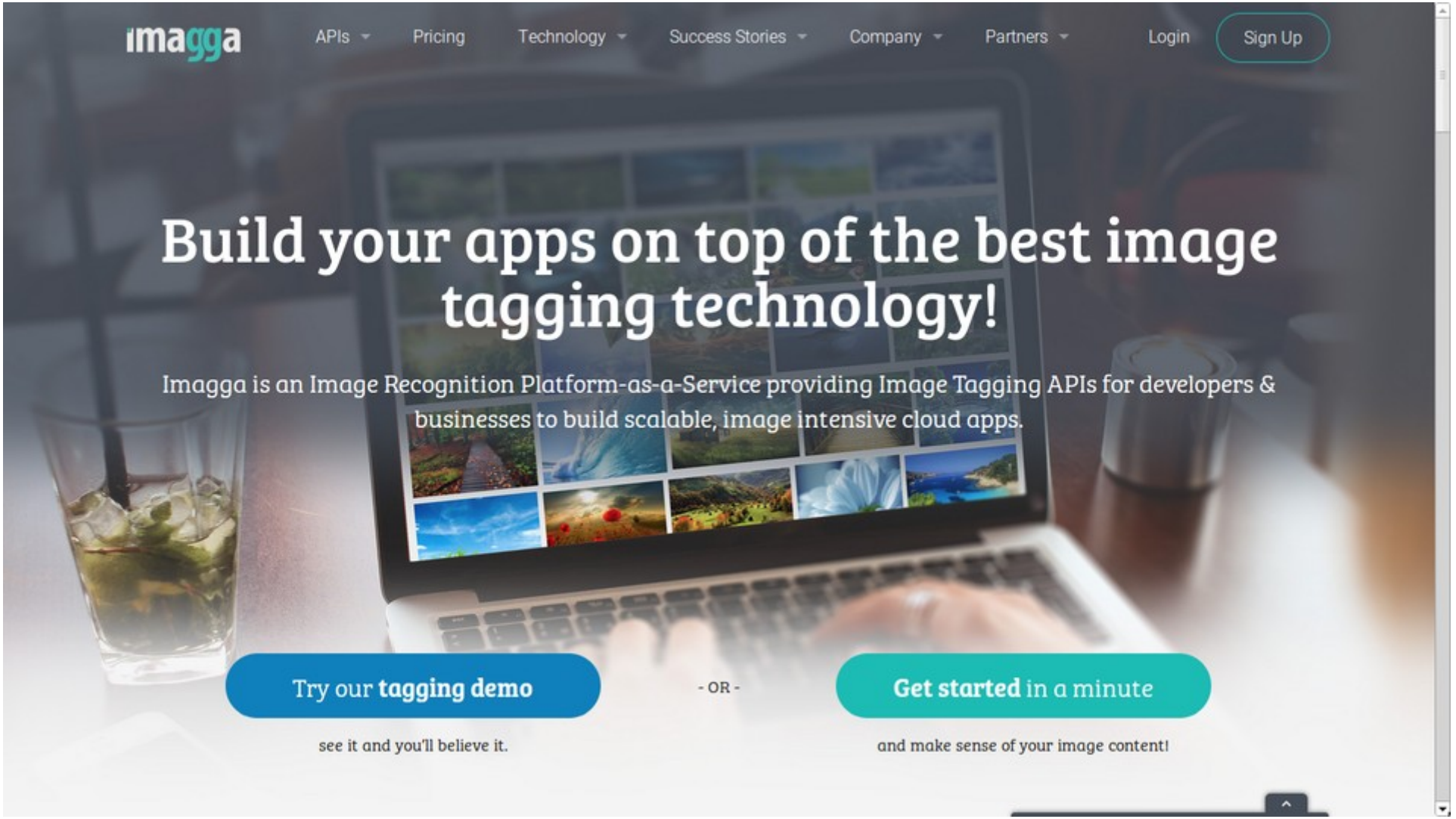
更強大的看圖說故事



CNN : Convolutional Neural Network(捲積式類神經網路)
RNN : Recurrent Neural Network(遞迴式類神經網路)

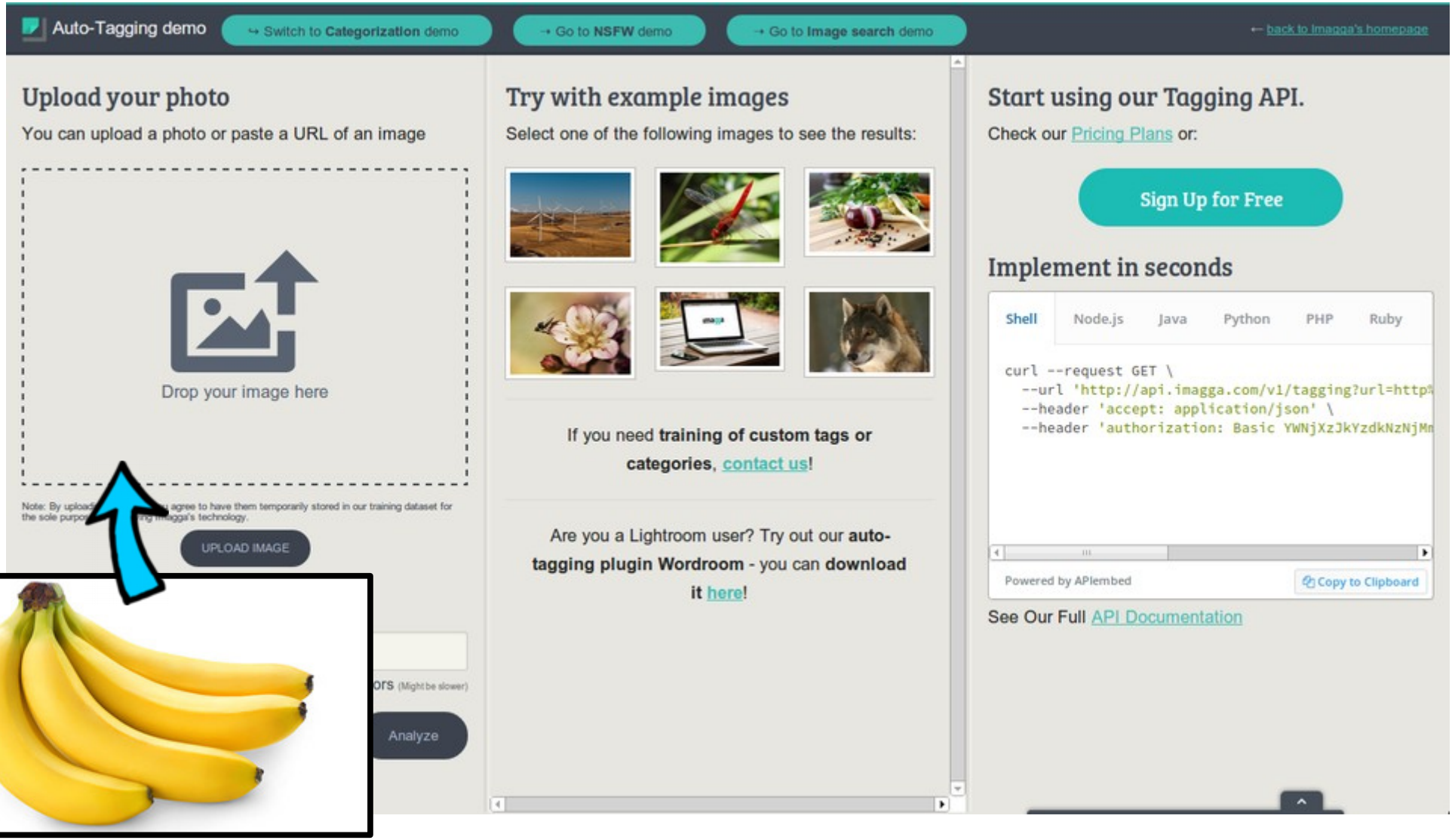
http://googleresearch.blogspot.tw/2014/11/a-picture-is-worth-thousand-coherent.html

影像分類服務



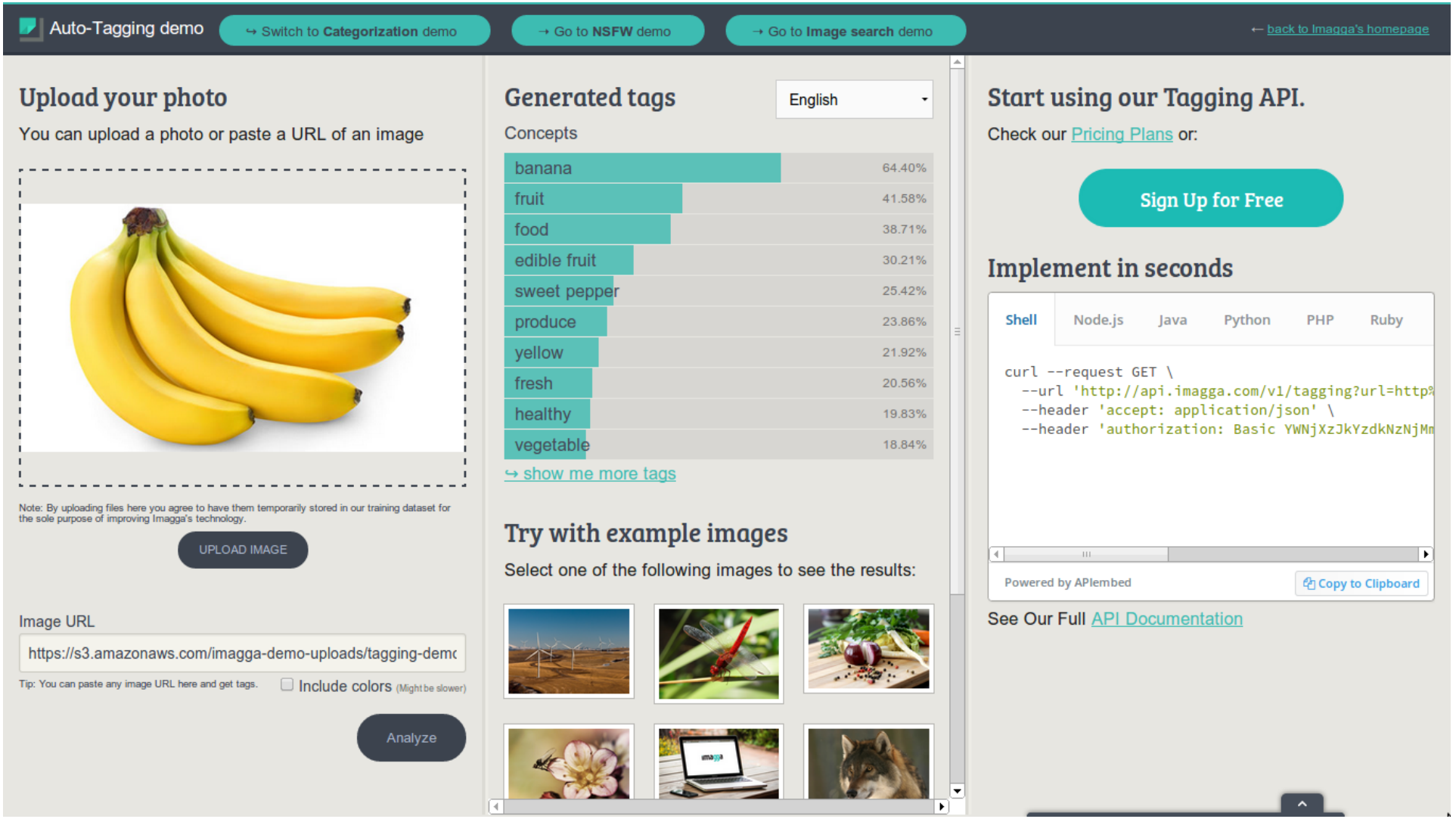
https://imagga.com/

看 DEMO



http://imagga.com/auto-tagging-demo/?key=123#

Auto-Tagging



http://imagga.com/auto-tagging-demo/?key=123#

如何開始使用服務？

1. 註冊與認證
- https://imagga.com/auth/signup

2. 取得 Authorization
- https://imagga.com/profile/dashboard

3. 串接
- input: 程式指定圖片的 URL 或是上傳圖檔
- output: JSON 字串

Dashboard

- Basic authentication 使用 authorization value

https://imagga.com/profile/dashboard

HTTP 兩三件事

GET & POST Method

- RFC 2616 / Hypertext Transfer Protocol - HTTP/1.1

- HTTP method
  - OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT

- GET 像明信片
  - 資料(query string)在URL 傳送

- POST 像信紙 + 信封
  - 資料可以在message-header
  - 也可以在message-body

把設定檔寫成獨立的 conf 檔案

- $ nano web_service.conf

```
[imagga]
authorization = Basic
YWNjXzJkYzdkNzNjMmYwODliMToxYzQ3Yzg2ZDg0YjdmY
jdjYjZjNzQ1NTQ1MmYwNTgzMQ==
```

將黃色部份換成自己的 authorization

用 import ConfigParser 讀取 conf 檔案

```
import ConfigParser

config = ConfigParser.ConfigParser()
config.read('web_service.conf')
authorization = config.get('imagga', 'authorization')
```

這是從 conf 讀出的變數

Python 程式串接 (File URL)

```
import requests

authorization = config.get('imagga', 'authorization')
url = "http://api.imagga.com/v1/tagging"
querystring = {"url":"網路上圖檔的URL網址"}
headers = {
    'accept':"application/json",
    'authorization': authorization
    }

response = requests.request("GET", url, headers=headers,
params=querystring)
```

tagging API 的 URL

這是從 conf 讀出的變數

http://docs.imagga.com/

回傳結果 (JSON)


```
{
    "results": [
        {
            "image": "http://playground.imagga.com/static/img/example_photo.jpg",
            "tags": [
                {
                    "confidence": 100,
                    "tag": "shore"
                },
                {
                    "confidence": 52.19003265126093,
                    "tag": "sea"
                },
                {
```


- JSON(JavaScript Object Notation)是一種資料結構
  - 物件(object)以 { } 表示
  - 鍵 / 值(collection)以 : 表示
  - 陣列(array)以 [ ] 表示

http://docs.imagga.com/

解析 JSON

```
import json

authorization = config.get('imagga', 'authorization')
url = "http://api.imagga.com/v1/tagging"
querystring = {"url":"網路上圖檔的URL網址"}
headers = {
    'accept':"application/json",
    'authorization': authorization
    }

response = requests.request("GET", url, headers=headers,
params=querystring)

data = json.loads(response.text.encode("ascii"))
print data["results"][0]["tags"][0]["tag"].encode("ascii")
```

這是從 conf 讀出的變數

解析 JSON

```
{
    "results": [
        {
            "image": "http://playground.imagga.com/static/img/example_photo.jpg",
            "tags": [
                {
                    "confidence": 100,
                    "tag": "shore"
                },
                {
                    "confidence": 52.19003265126093,
                    "tag": "sea"
                },
                {
```

```
data = json.loads(response.text.encode("ascii"))
print data["results"][0]["tags"][0]["tag"].encode("ascii")
```

解析 JSON

```
{
    "results": [
        {
            "image": "http://playground.imagga.com/static/img/example_phc
            "tags": [
                {
                    "confidence": 100,
                    "tag": "shore"
                },
                {
                    "confidence": 52.19003265126093,
                    "tag": "sea"
                },
                {
```

result[0]

```
data = json.loads(response.text.encode("ascii"))
print data["results"][0]
```

解析 JSON

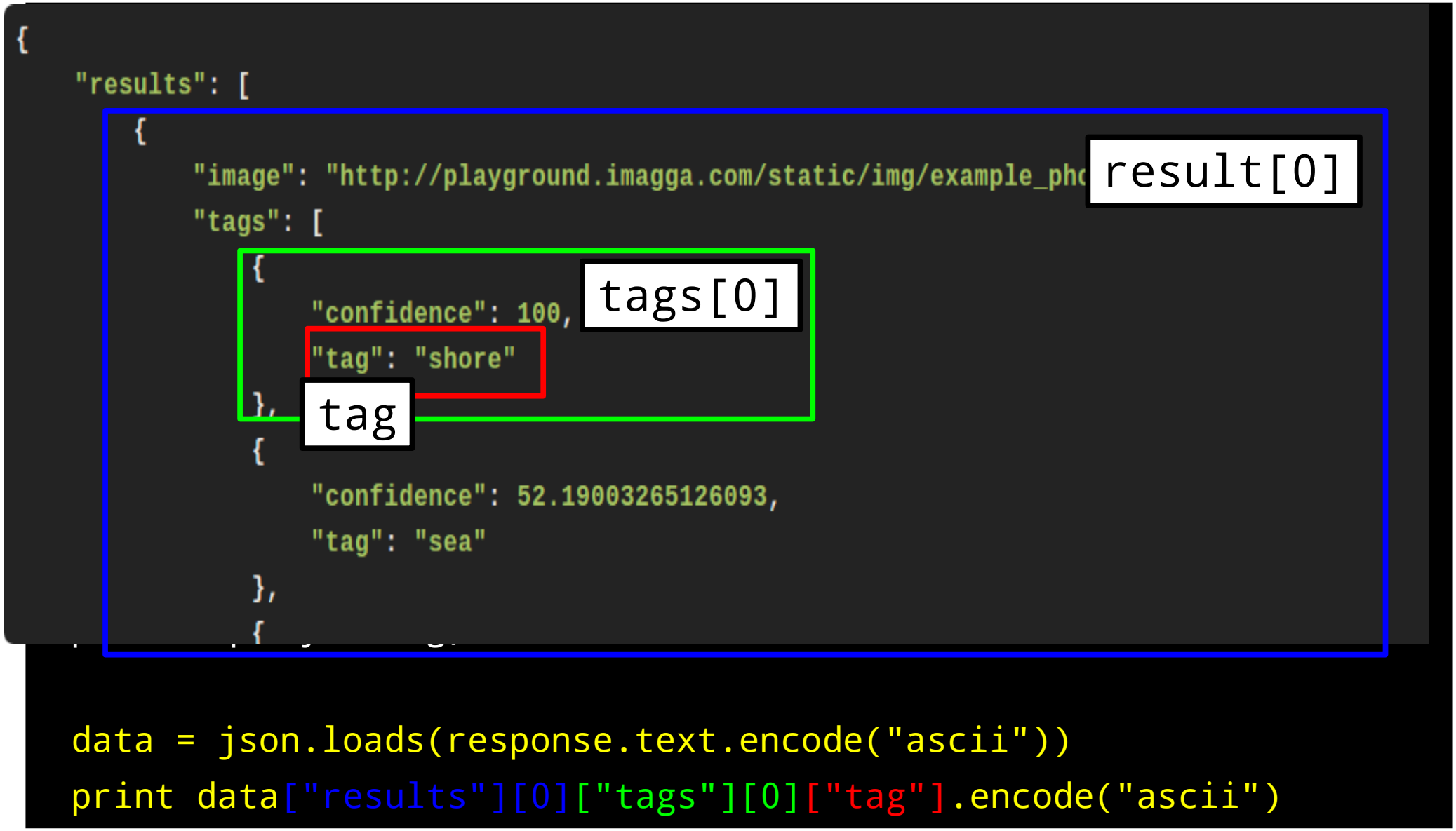

```
{
    "results": [
        {
            "image": "http://playground.imagga.com/static/img/example_phc
            "tags": [
                {
                    "confidence": 100,
                    "tag": "shore"
                },
                {
                    "confidence": 52.19003265126093,
                    "tag": "sea"
                },
                {
```


result[0]

tags[0]

```
data = json.loads(response.text.encode("ascii"))
print data["results"][0]["tags"][0]
```

解析 JSON


{
"results": [
{
"image": "http://playground.imagga.com/static/img/example_phc
result[0]
"tags": [
{
"confidence": 100,
tags[0]
"tag": "shore"
},
tag
{
"confidence": 52.19003265126093,
"tag": "sea"
},
data = json.loads(response.text.encode("ascii"))
print data["results"][0]["tags"][0]["tag"].encode("ascii")

DEMO
imagga_tag_file_url.py

```
$ cd ~/cam-py-cv/day1/03-imagga_web_service
$ python imagga_tag_file_url.py
```

Python 程式串接 (Upload File)

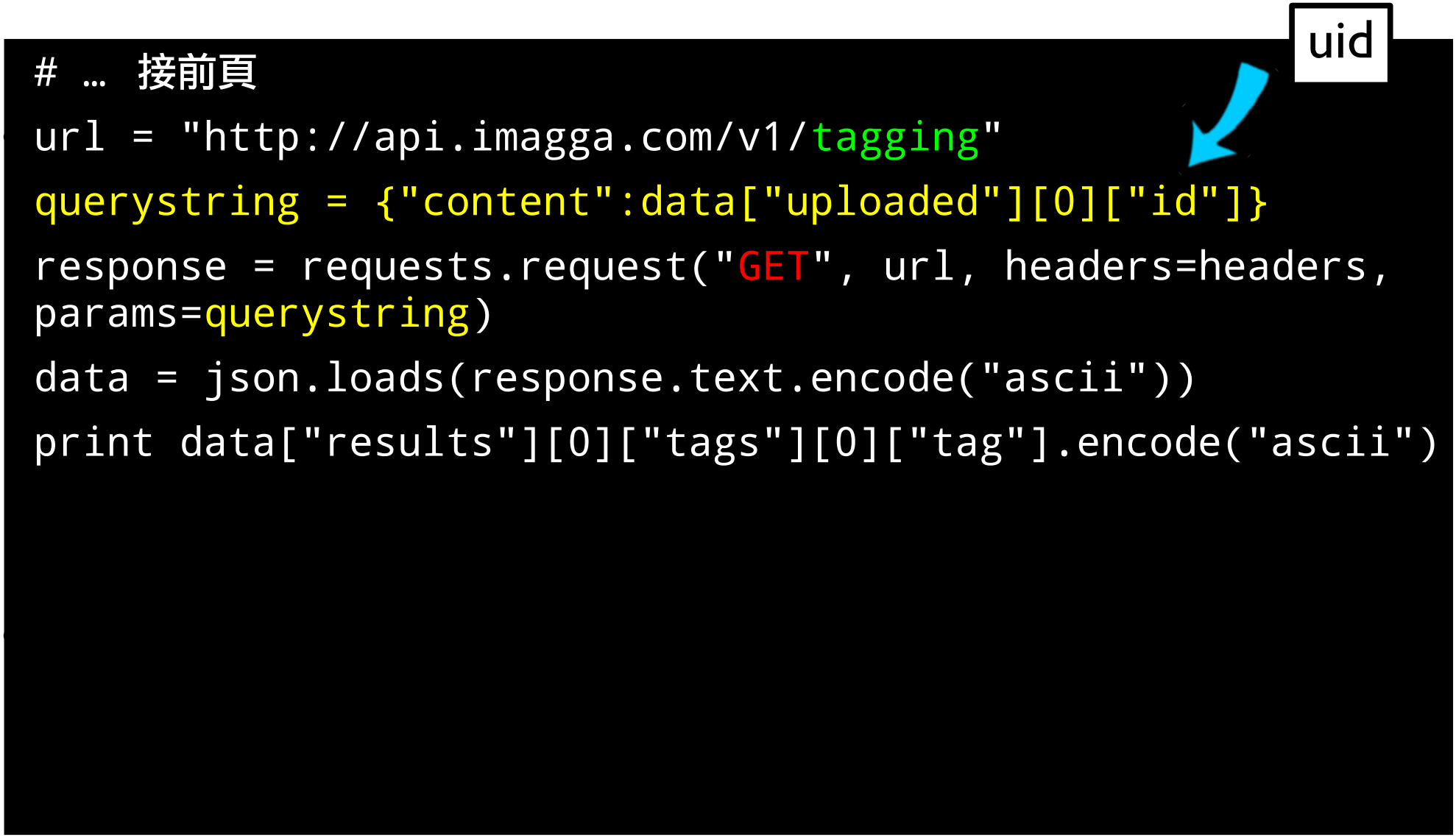
- 先拍張照片
- 根據文件得知查詢上傳的檔案需要兩個步驟
  - 上傳檔案後取得檔案 uid
  - 將 uid 以參數方式送出查詢

上傳檔案取得檔案 uid

不同的 URL

```
import requests
import json

url = “http://api.imagga.com/v1/content”
files = {“file”: open(“/home/pi/test.jpg”,”rb”)}
headers = {
    'accept':"application/json",
    'authorization': authorization
    }
response = requests.post(url, files=files, headers=headers)
print(response.text)
data = json.loads(response.text.encode("ascii"))
print data["uploaded"][0]["id"]
```

將 uid 以參數方式送出查詢



DEMO
imagga_tag_upload_file.py

```
$ cd ~/cam-py-cv/day1/03-imagga_web_service
$ python imagga_tag_upload_file.py
```

實做雲端相機

拆解功能

- 執行 python 程式後的步驟
  1. 拍照 & 存檔
  2. 將檔案上傳後取得 uid
  3. 再將 uid 以參數方式送出查詢
  4. 將查詢結果用 TTS(Text To Speech) 發聲

```
$ echo tag | festival --tts
```

由 imagga 回傳的 tag

拍照後的連續動作

```
camera = picamera.PiCamera()
camera.capture("test.jpg")
files = {"file": open("test.jpg", "rb")}

url = "http://api.imagga.com/v1/content"
response = …

url = "http://api.imagga.com/v1/tagging"
querystring = ...
data = json.loads(...)

obj = data[...][“tag”]
print "<< " + obj + " >>"
cmd = "echo " + obj + " | festival --tts"
os.system(cmd)
```

上傳圖檔

查詢圖檔 tag

喇叭發聲 (TTS)

練習

- 實做能自動辨識物體的相機，可在拍照後的圖片疊上第一個 tag(文字)+ 第一個信心水準 (confidence)

  http://goo.gl/2ShIrQ

- 拆解功能
  1. 拍照 & 存檔
  2. 將檔案上傳後取得 uid
  3. 再將 uid 以參數方式送出查詢
  4. 將文字疊在照片上，信心水準型態為 float 需轉型

小結

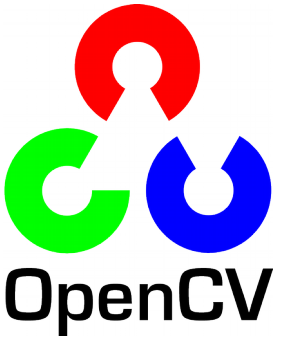
- Python 的 request 套件可提供 get 和 post 方法
- 網路 RESTful API 回傳結果通常為 JSON 格式

實驗 4:Video4Linux 2nd(V4L2)

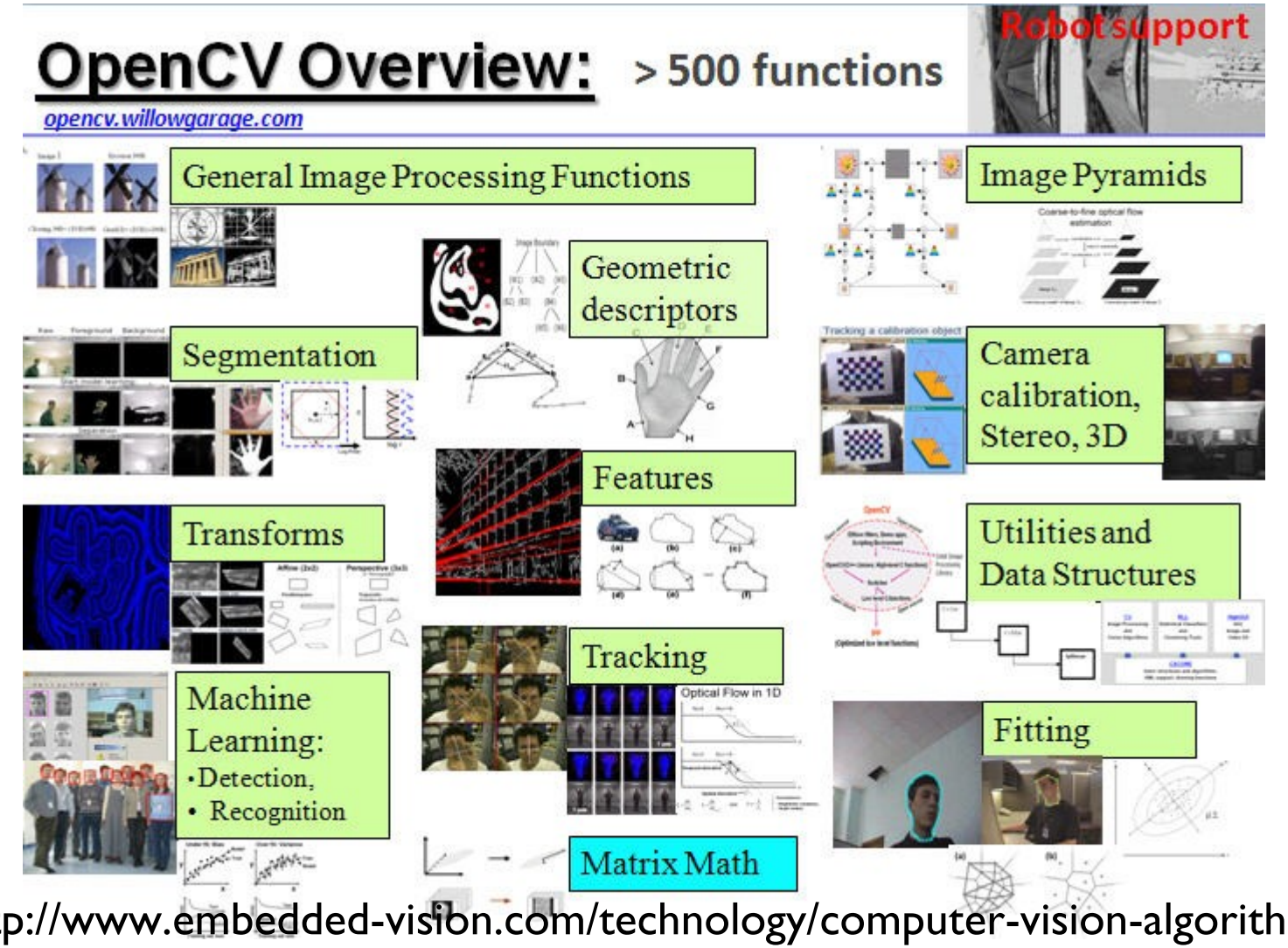
目的：從 OpenCV 看 Camera 和 Webcam

OpenCV

- Open Source Computer Vision Library

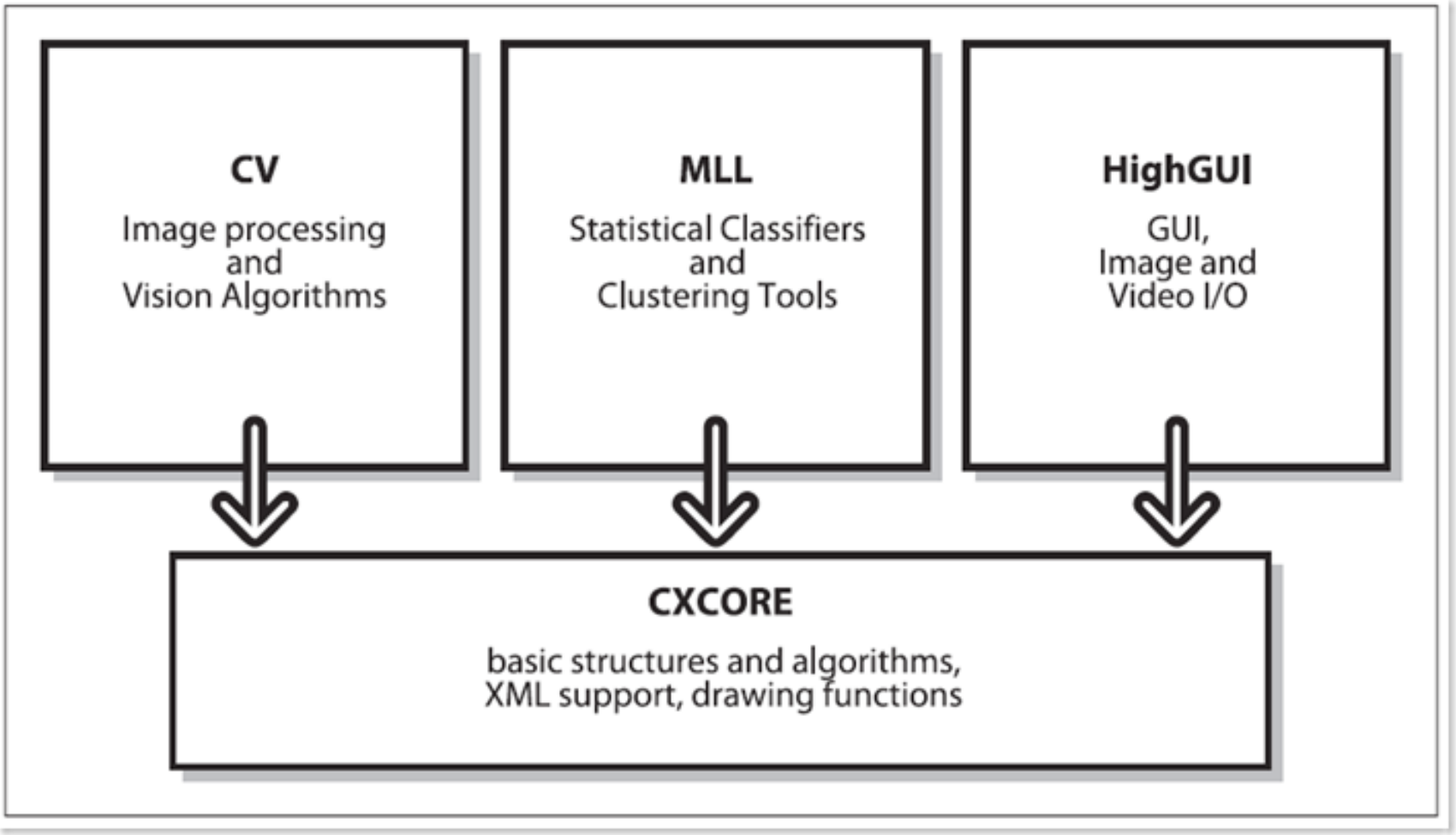


- 跨平台的計算機函式庫，主要由 C/C++ 撰寫



http://www.embedded-vision.com/technology/computer-vision-algorithms

OpenCV 的架構



http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/OReilly%20Learning%20OpenCV.pdf

載入圖檔並顯示

```
import cv2
import sys

imagePath = sys.argv[1]
image = cv2.imread(imagePath)

cv2.imshow("preview", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

DEMO
image_load.py

```
$ cd ~/cam-py-cv/day1/04-webcam
$ python image_load.py /home/pi/test.jpg
```

OpenCV 還提供了許多好用的函式

- 在圖上疊字

  `cv2.putText(img, TEXT, (x,y), font, scale, color)`

- 範例：

```
cv2.putText(image, "HELLO", (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255))
                                        BGR
```


http://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html

除了開檔與顯示以外

- 將影像存成檔案

  `cv2.imwrite(filename, image)`

- 範例：

```
cv2.imwrite("puttext.png", image)
```


http://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html

DEMO
puttext.py

```
$ cd ~/cam-py-cv/day1/04-webcam
$ python puttext.py abba.png
```

除了拍照存檔後開圖以外，

可以即時預覽 Camera 的結果嗎？

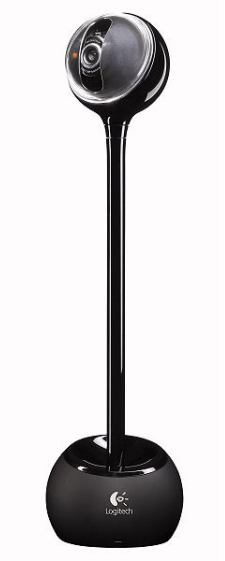
讀取 Camera 並顯示

0 是 V4L2 的裝置節點

```
import cv2

cap = cv2.VideoCapture(0)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH,  320)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, 240)

while True:
    ret, frame = cap.read()
    cv2.imshow(“preview”, frame)
    if cv2.waitKey(1) & 0xFF == ord(“q”):
        break

cap.release()
cv2.destroyAllWindows()
```

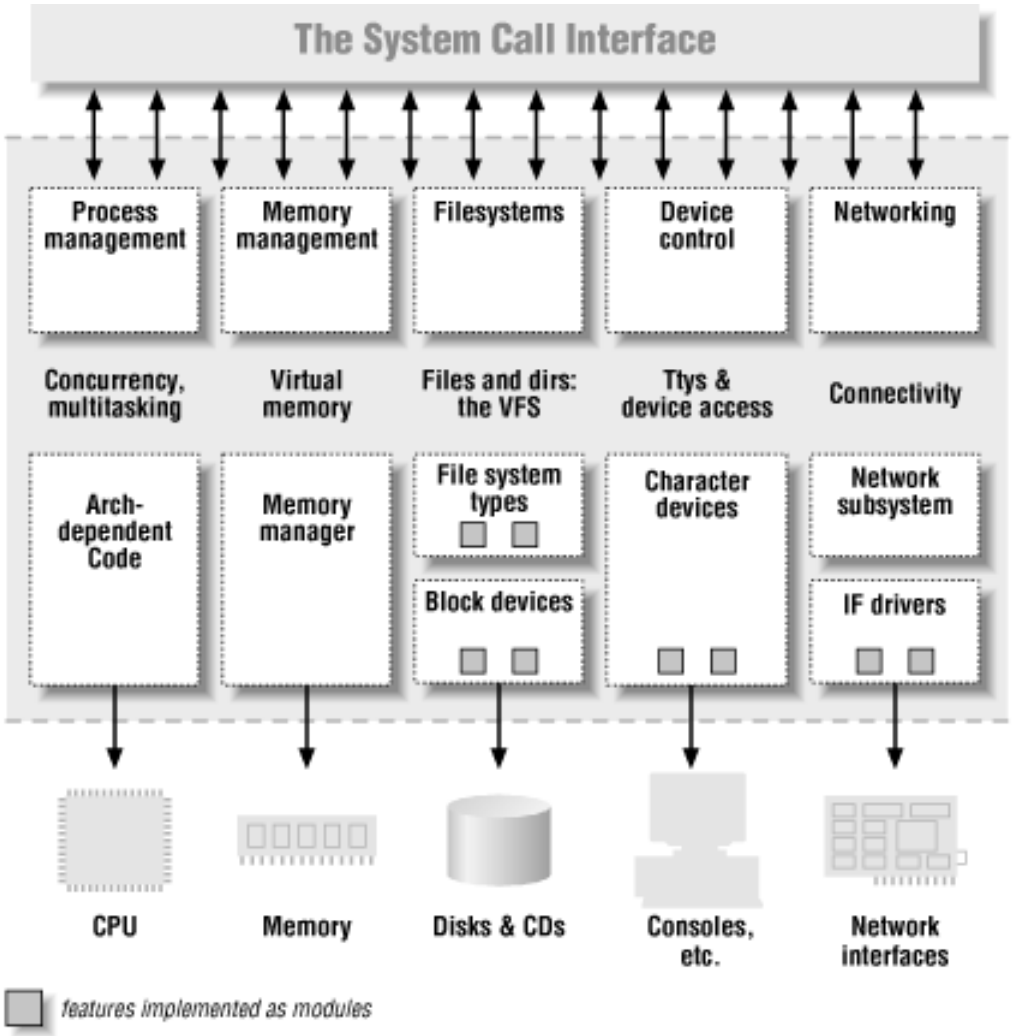
回來看 Camera
你有用過 Webcam 嗎？

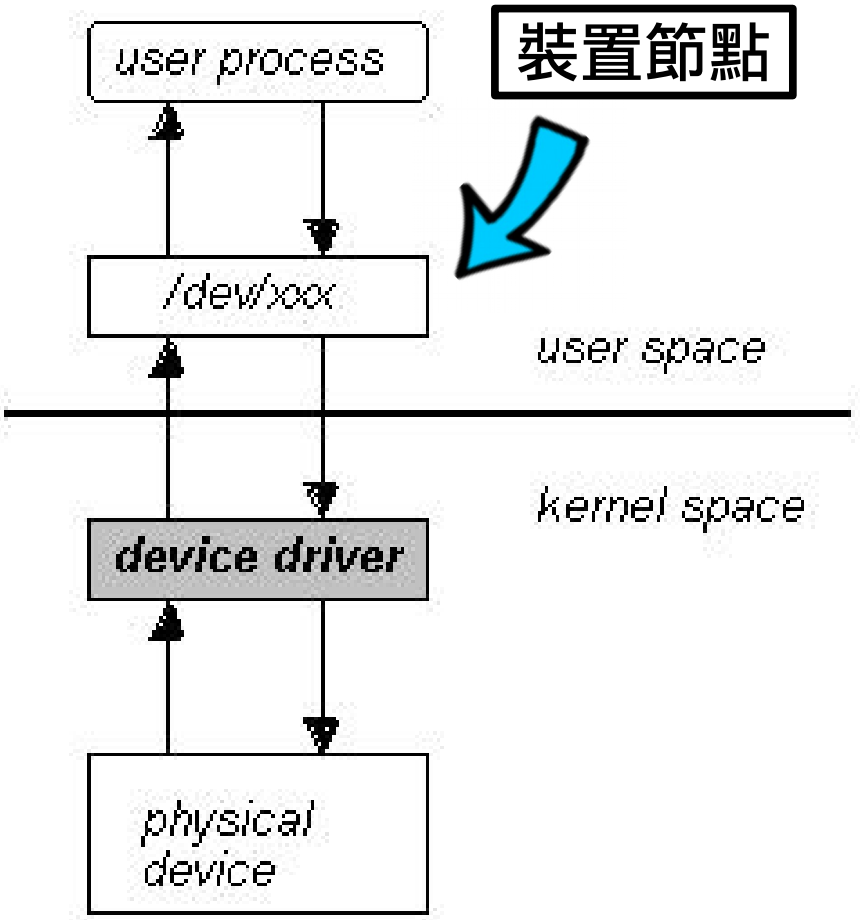
Webcam



http://www.slideshare.net/gxben/abs-2014-android-kit-kat-internals

Linux 如何存取硬體？

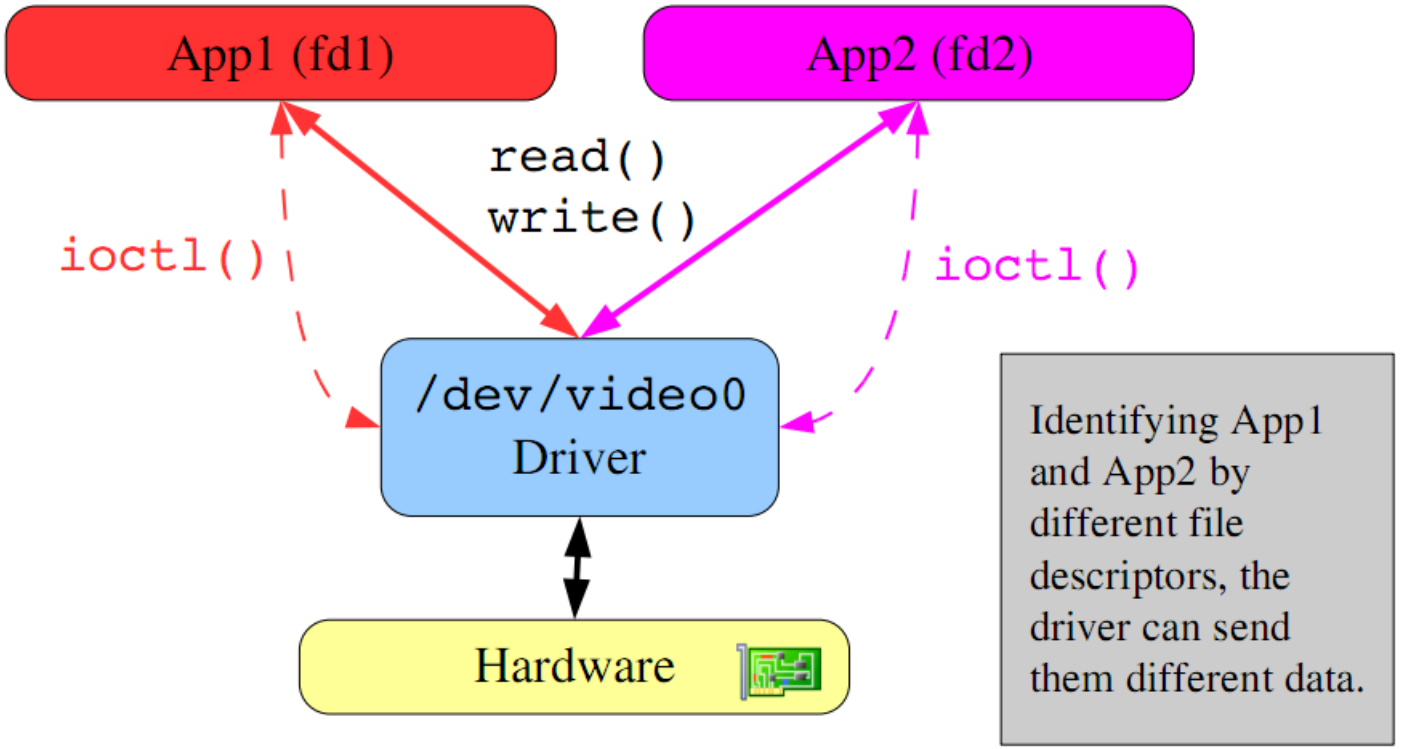




http://www.xml.com/ldd/chapter/book/ch01.html

Video For Linux 2nd(V4L2)

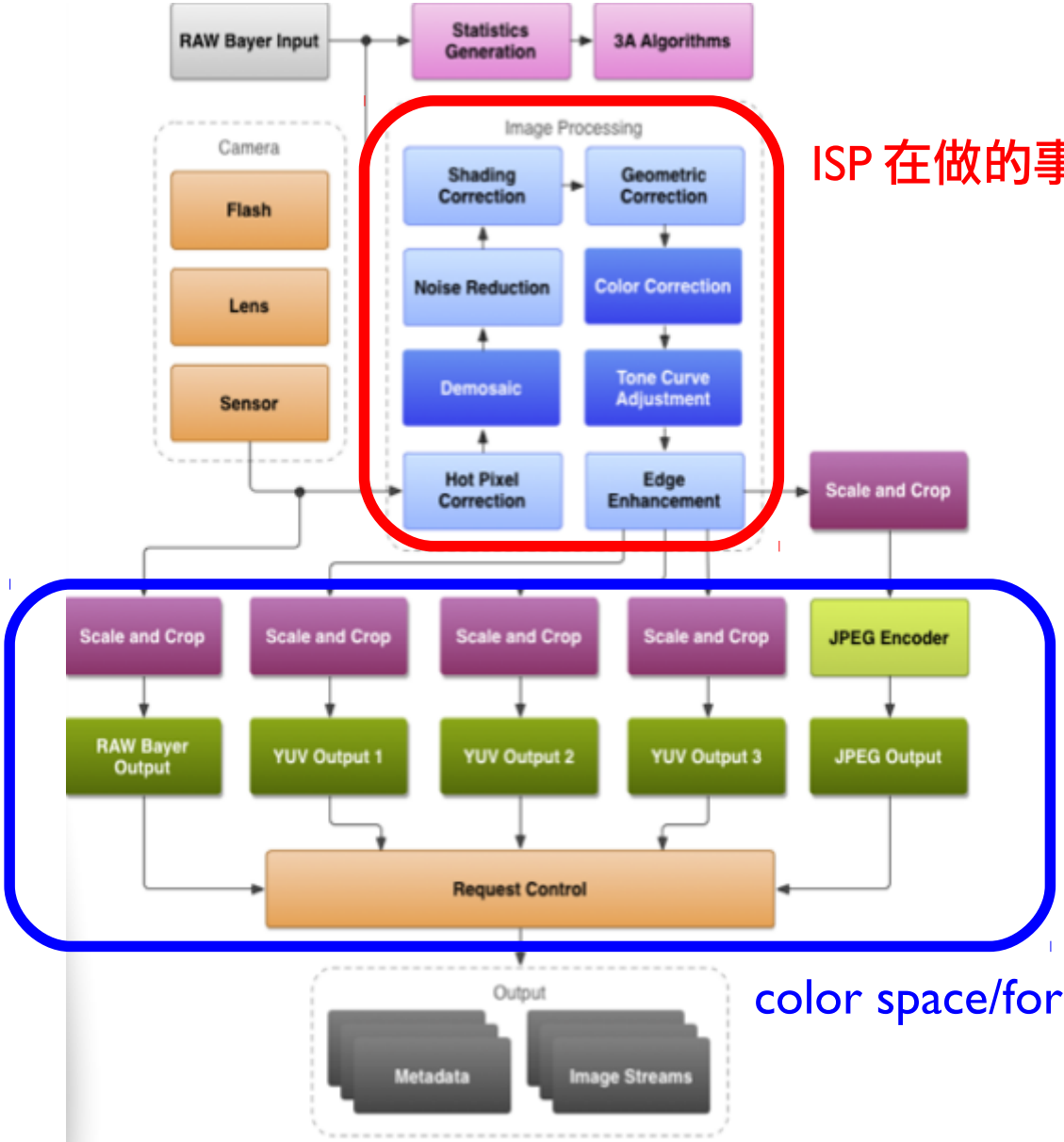
- 是 Linux 對視訊設備（如 Webcam）的 Userspace API



http://free-electrons.com/doc/embedded_linux_multimedia.pdf
https://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html

Camera ≠ Webcam

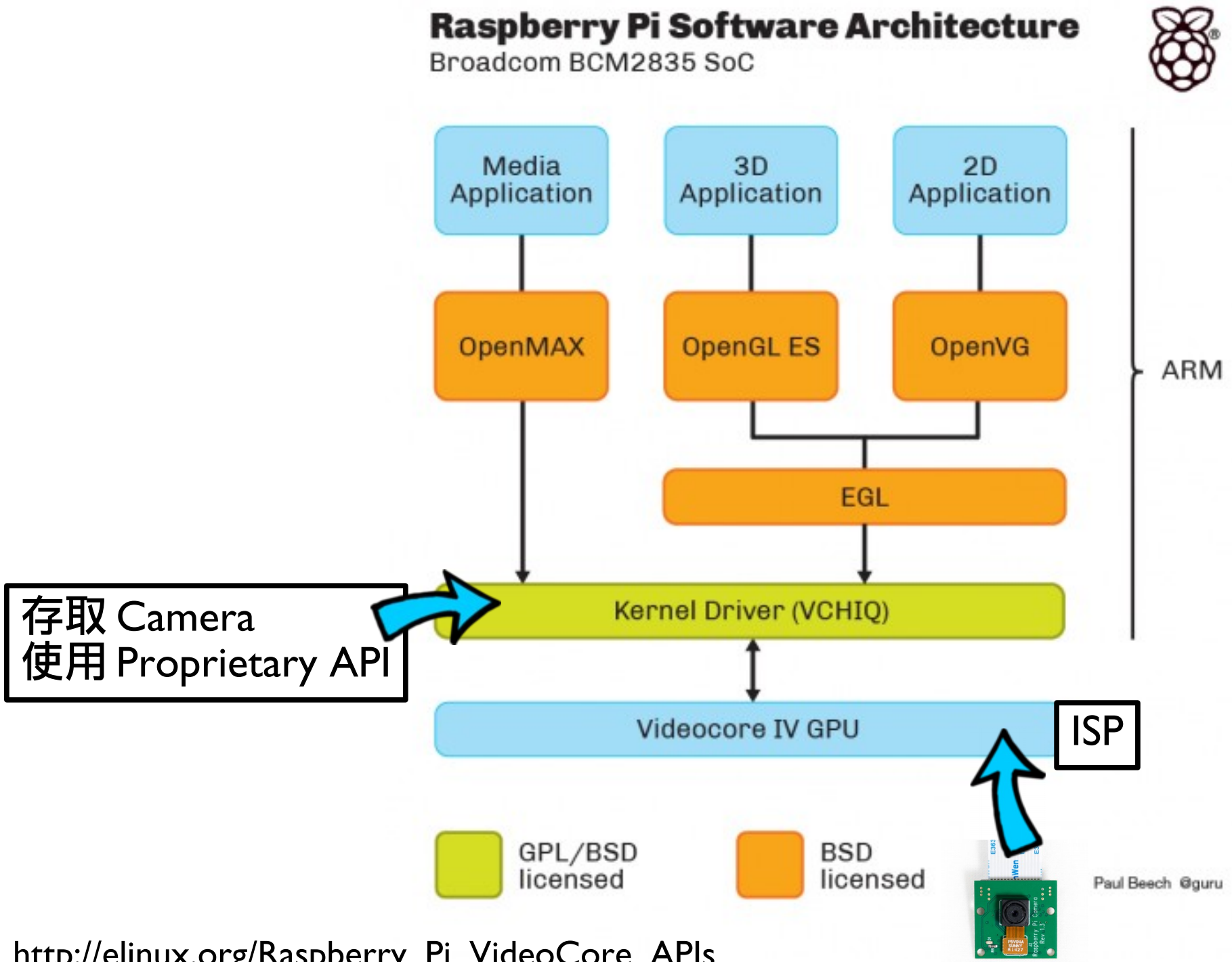
Image Processing Pipeline



ISP 在做的事 (HW)

color space/format 轉換 (SW)

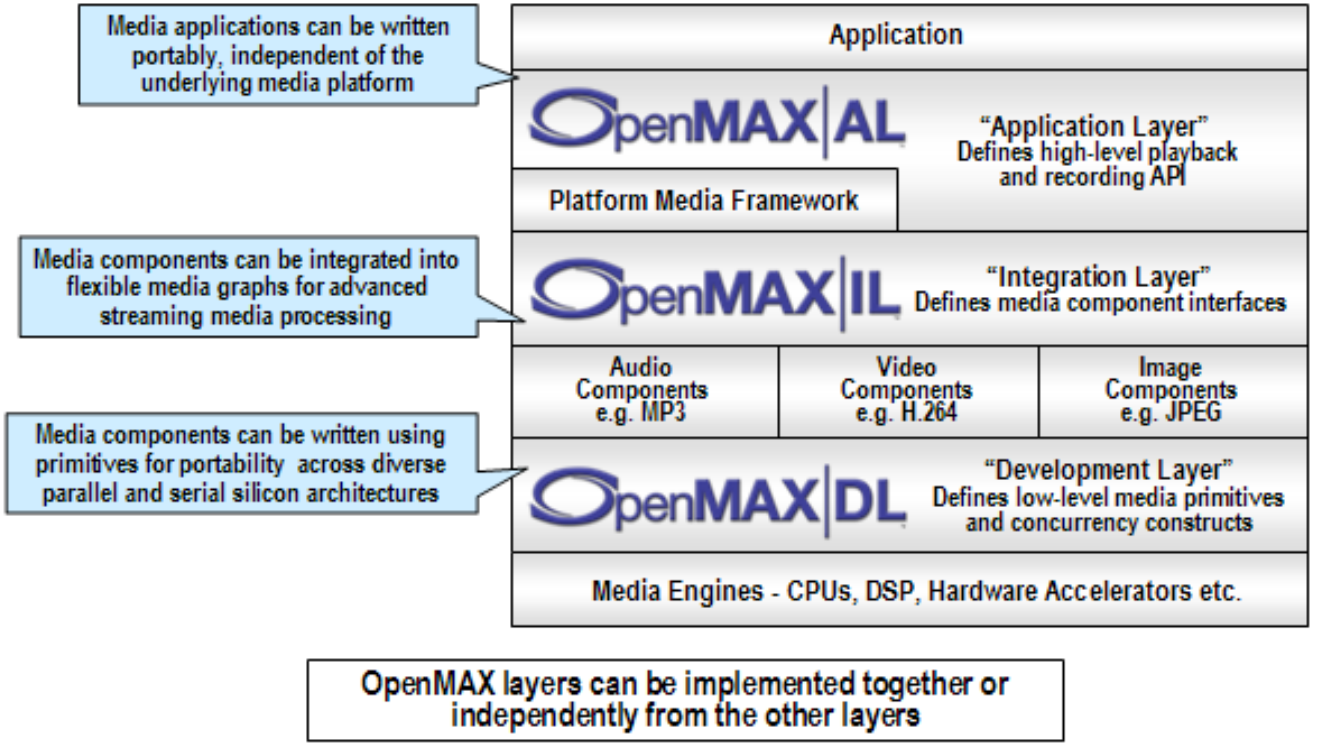
http://www.slideshare.net/gxben/abs-2014-android-kit-kat-internals



http://elinux.org/Raspberry_Pi_VideoCore_APIs

OpenMAX

- 開放多媒體加速層 (Open Media Acceleration)
  - 由 Khronos Group 提出的標準
  - 統一的介面，加速大量多媒體資料的處理



https://www.khronos.org/openmax/

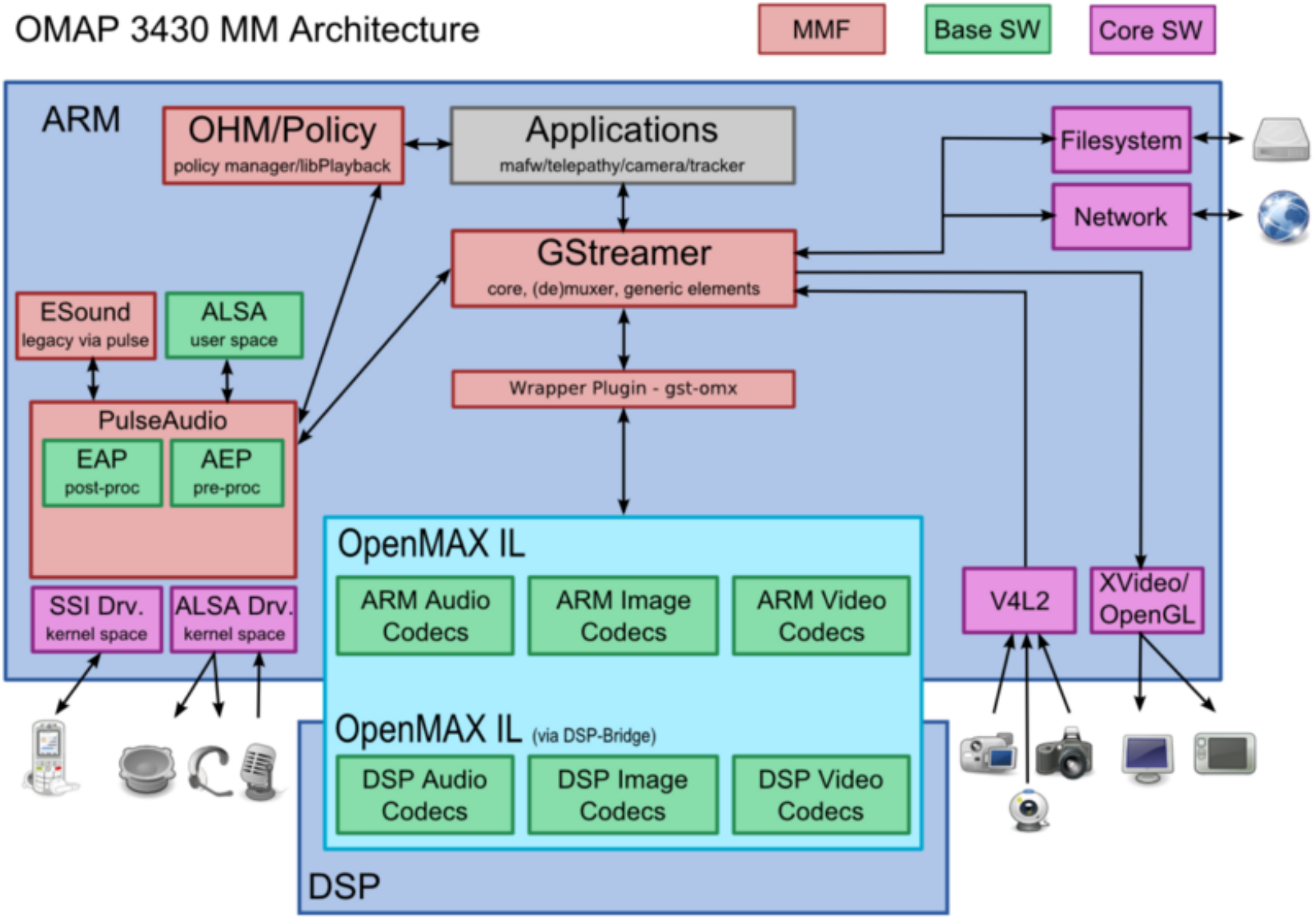
官方 V4L2 驅動程式

不是數字 1，是小寫 L

- Kernel driver
- 使用 camera 像是 webcam 一樣
  - $ sudo modprobe bcm2835-v4l2
- 可直接存取 /dev/videoX
  - $ v4l2-ctl --list-devices
  - $ v4l2-ctl --list-formats
  - $ v4l2-ctl -L

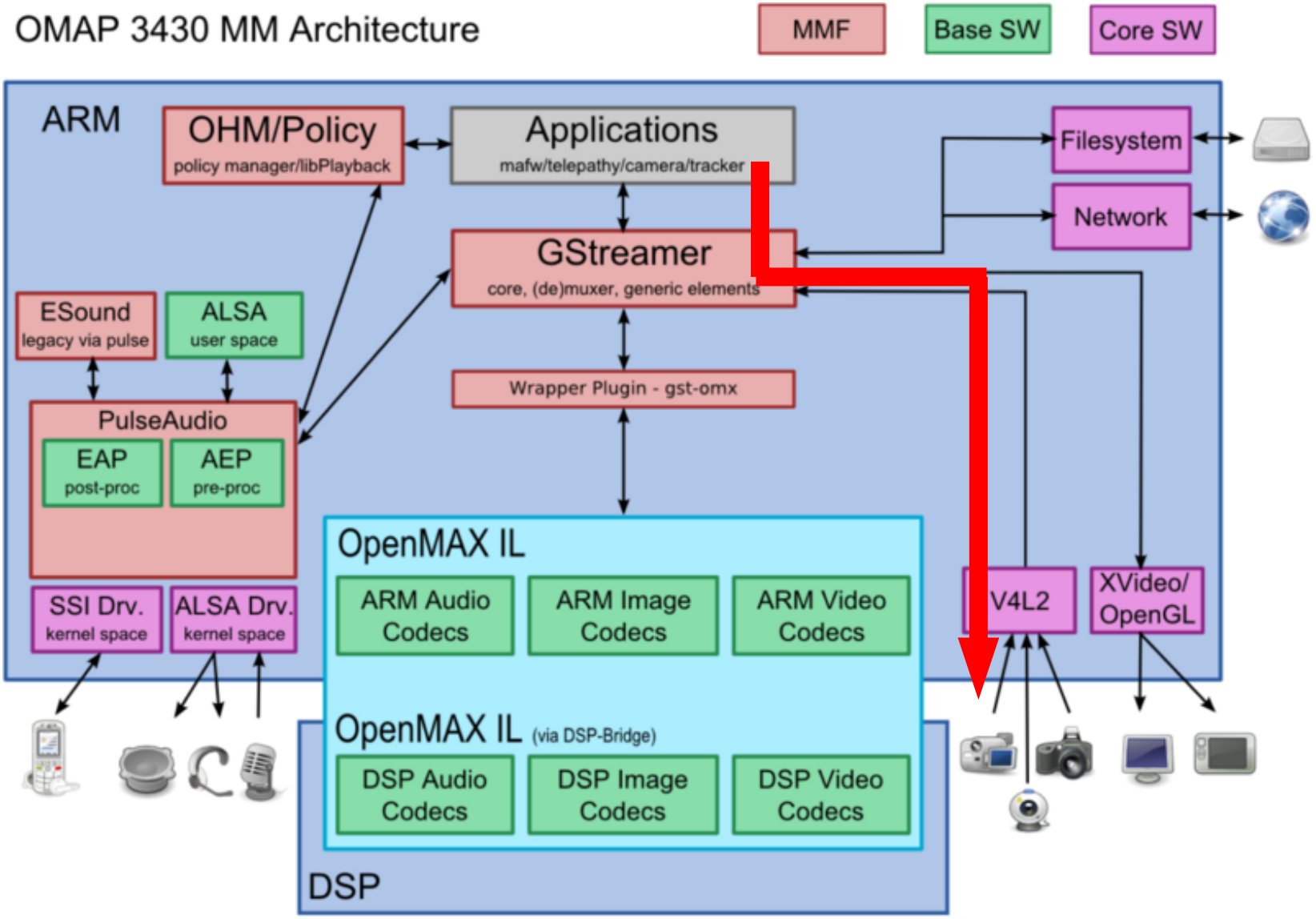
https://github.com/raspberrypi/linux/blob/rpi-3.10.y/Documentation/video4linux/bcm2835-v4l2.txt

Multimedia Stack Example
- TI OMAP 3430



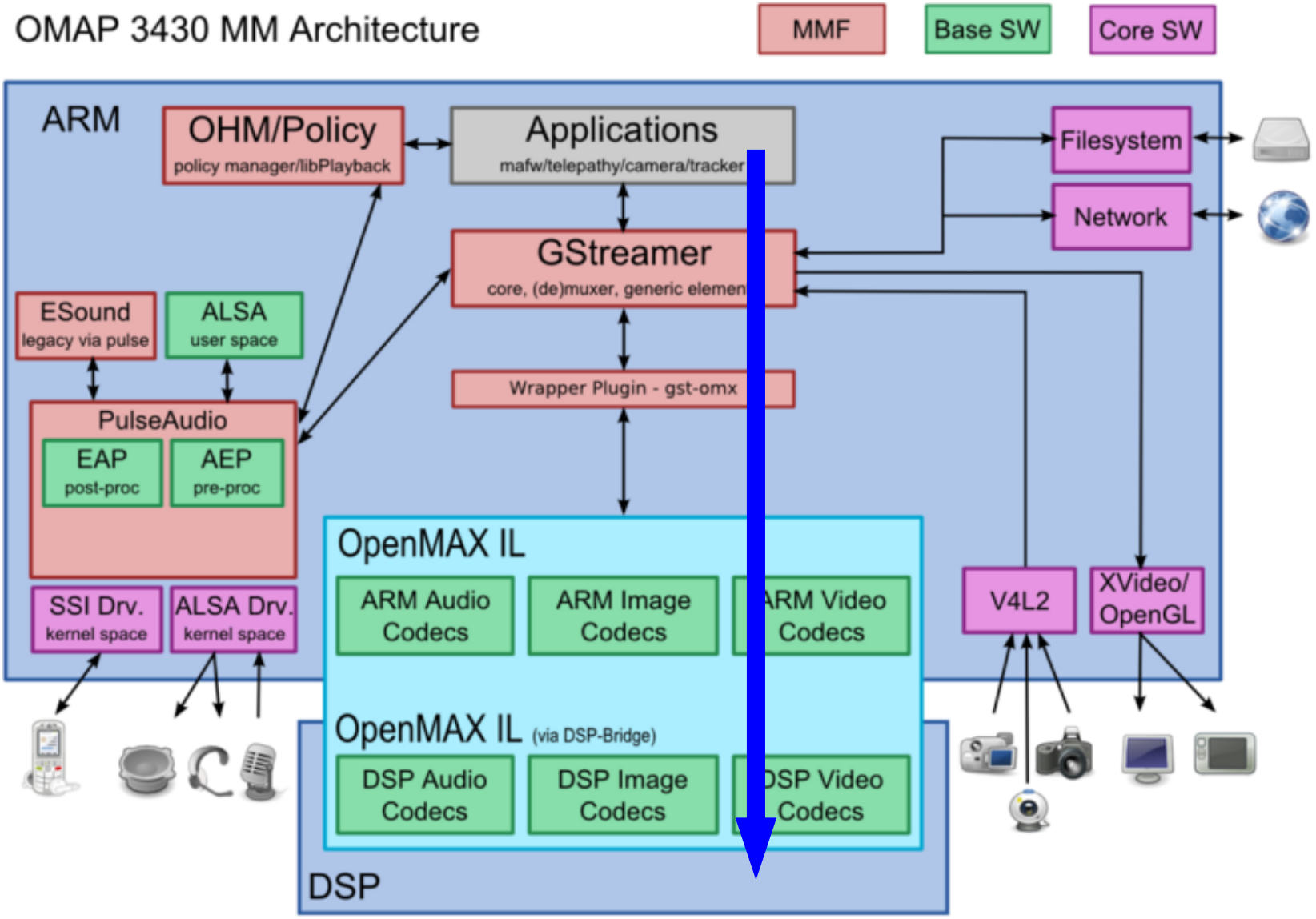
http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Architecture/Multimedia_Domain

走 V4L2



http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Architecture/Multimedia_Domain

走 OpenMAX



http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Architecture/Multimedia_Domain

比較兩種路徑的分別

- OpenMAX

  優點：

  - 接到 GPU，可硬解

  缺點：

  - 範例程式少
  - MMAL 只有標頭檔
  - 不易接 OpenCV

- V4L2

  優點：

  - 標準的 Linux API
  - 多數應用程式可支援
  - 可直接接到 OpenCV

  缺點：

  - CPU 運算，軟解，慢

但我想使用 C 語言接到 OpenMAX

- 參考這幾隻程式吧
- raspicam
  - https://github.com/raspberrypi/userland/tree/master/host_applications/linux/apps/raspicam
- rpi-omx-tutorial
  - https://github.com/SonienTaegi/rpi-omx-tutorial
- omxcam
  - https://github.com/gagle/raspberrypi-omxcam
- rpi-mmal-demo
  - https://github.com/tasanakorn/rpi-mmal-demo/tree/develop

使用前記得要先載入模組

$ sudo modprobe bcm2835-v4l2

不是數字 1，是小寫 L

DEMO

camera_preview.py

```
$ cd ~/cam-py-cv/day1/04-webcam
$ python camera_preview.py
```

小結

- 指令 sudo modprobe bcm2835-v4l2 可讓 camera 使用 V4L2 的 API
- 模組可以放在 /etc/rc.local 開機自動載入

```
sudo modprobe bcm2835-v4l2
```

- 也可以放在 /etc/modules 開機自動載入

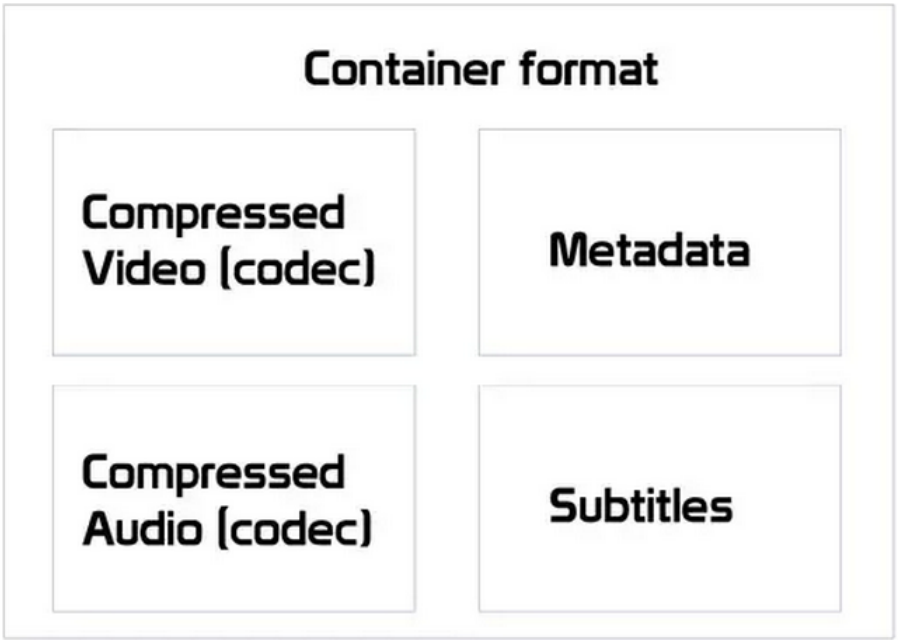
```
bcm2835-v4l2
```

實驗 5:Video Streaming

目的：瞭解如何做影像串流

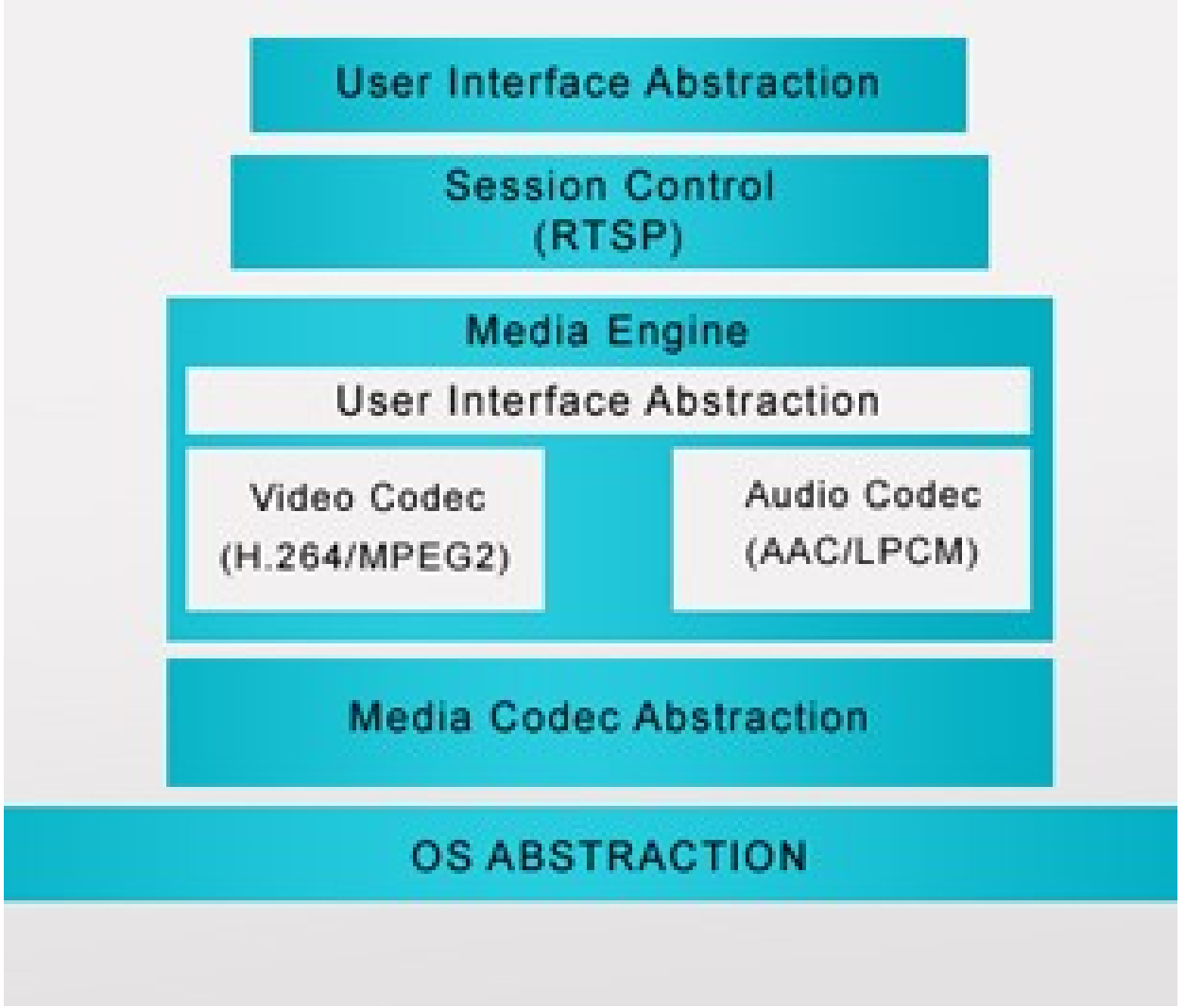
Codec and Container

- Codec 是 CODer+DECoder
- Codec 是資料壓縮 / 解壓縮的演算法
- Container 是裝載 Video 和 Audio 的容器



https://www.quora.com/What-are-containers-and-codecs

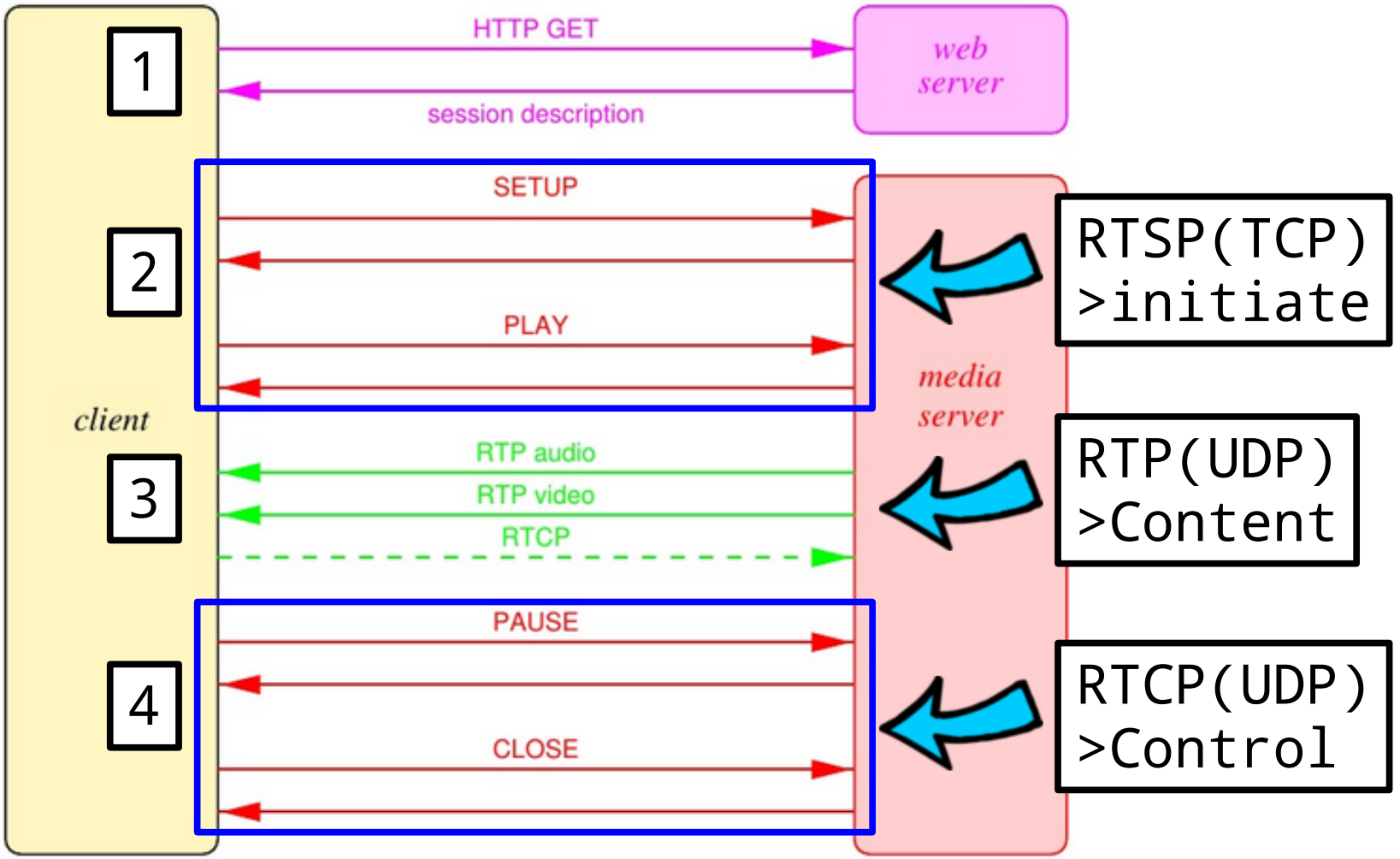
Streaming Media 技術架構



http://www.consilient-tech.com/strmmeda.shtml

使用 RTSP + H.264

RTSP 如何運作？



https://www.slideshare.net/Ronny72/internet-mediaondemand-the-realtime-streaming-protocol

RTSP Server

- 在Raspberry Pi

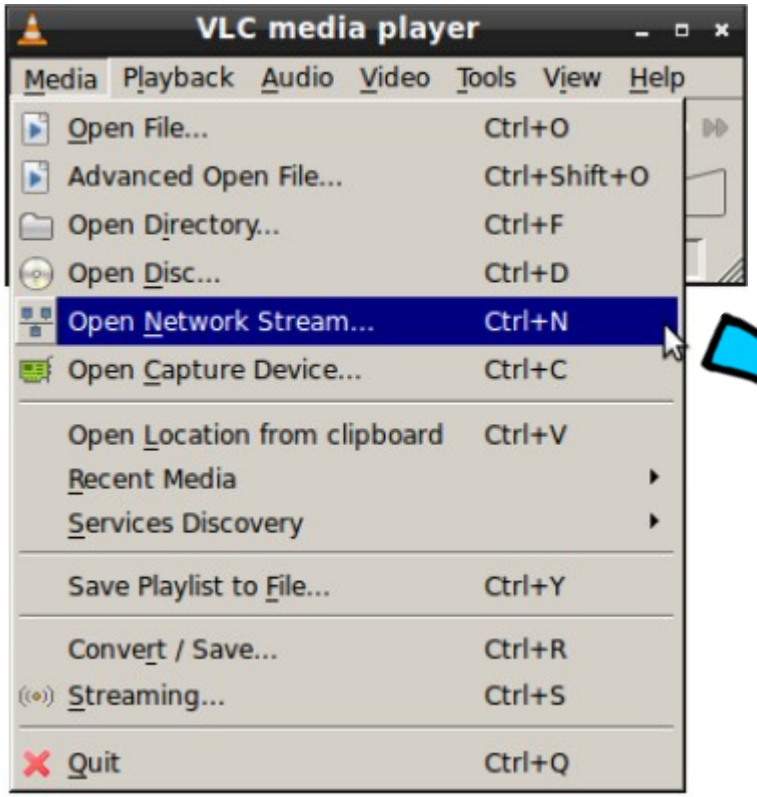
```
$ raspivid -o - -t 0 -hf -w 320 -h
240 -fps 15 | cvlc -vvv
stream:///dev/stdin --sout
'#rtp{sdp=rtsp://:8554}' :demux=h264
```

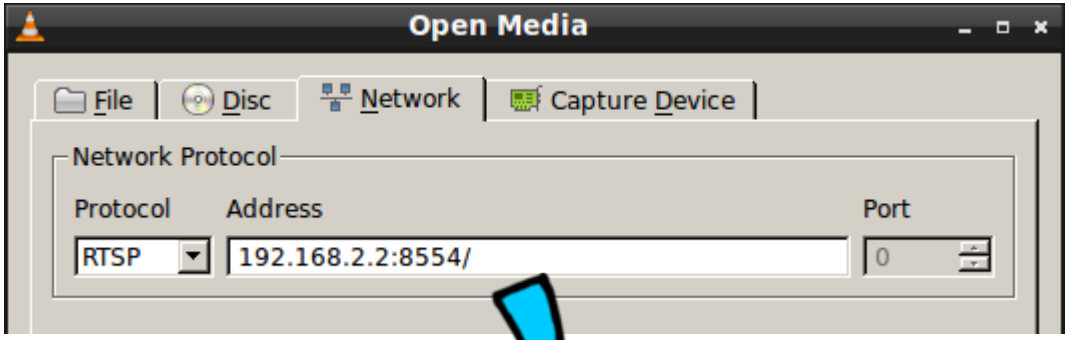
DEMO
rtsp_streaming.sh

```
$ cd ~/cam-py-cv/day1/05-streaming
$ ./rtsp_streaming.sh
```

RTSP Client

- 在 Windows 使用 VLC

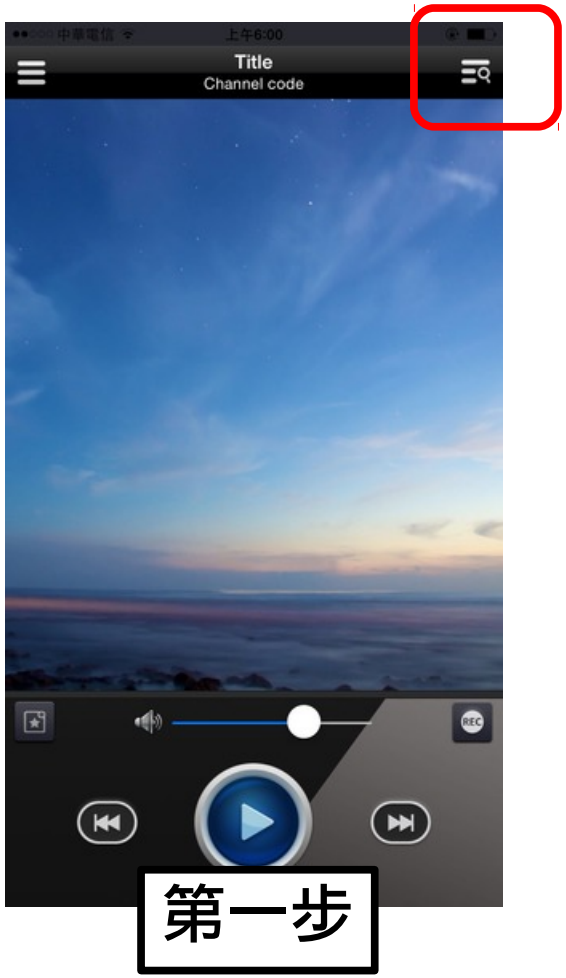


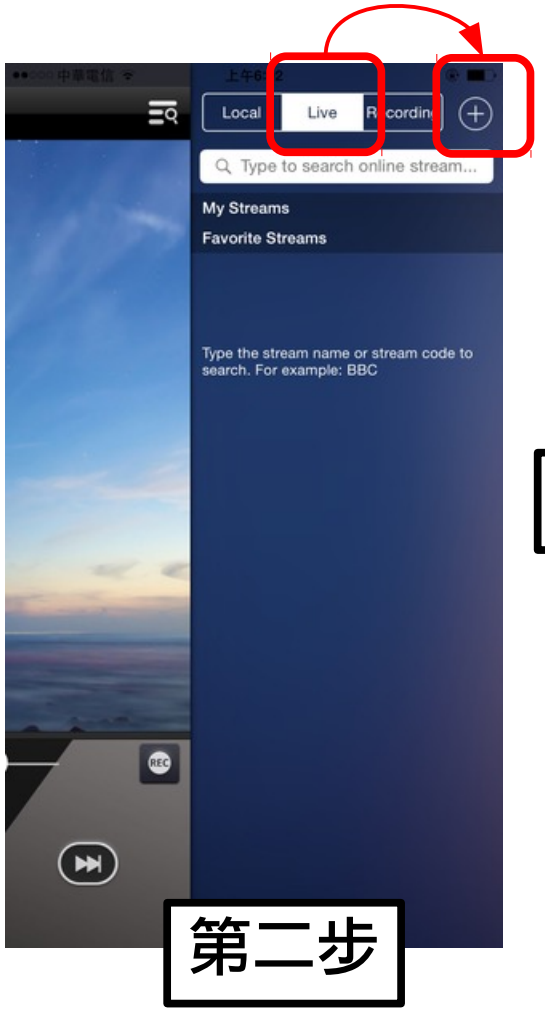


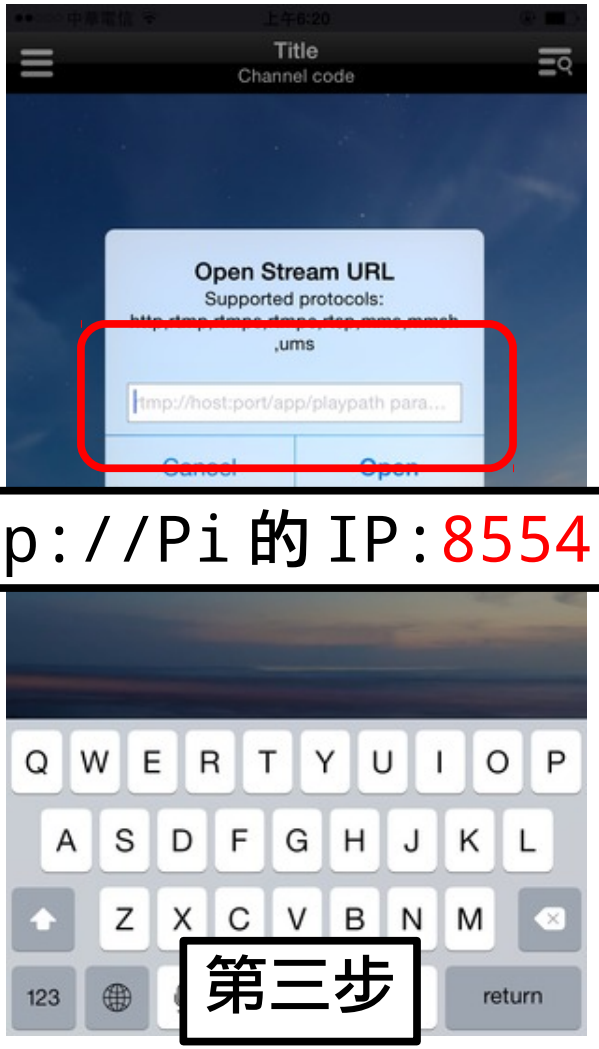
Pi 的 IP:8554/ 斜線很重要

RTSP Client

- 在 iPhone/iPad 使用 Live Media Player



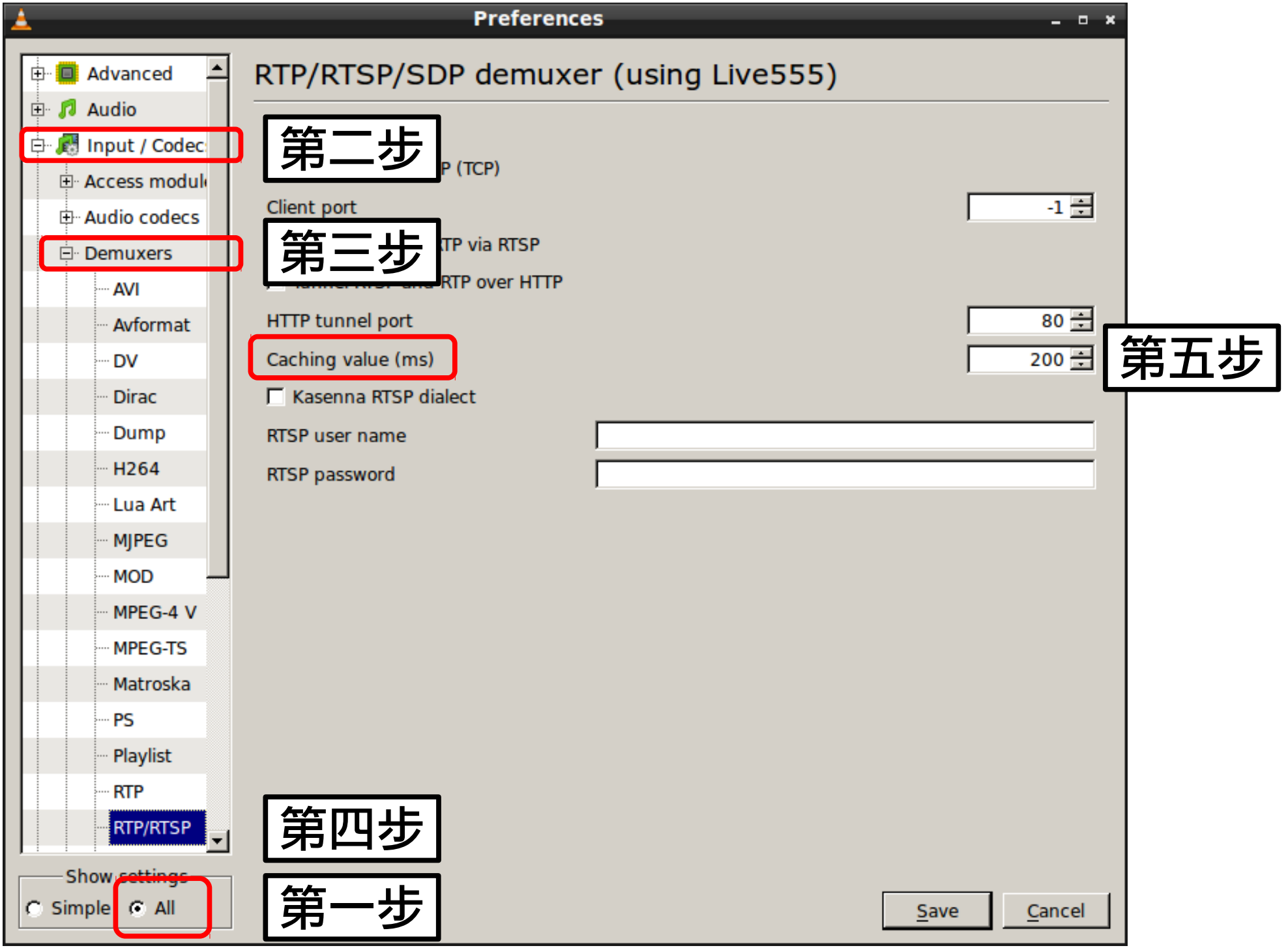


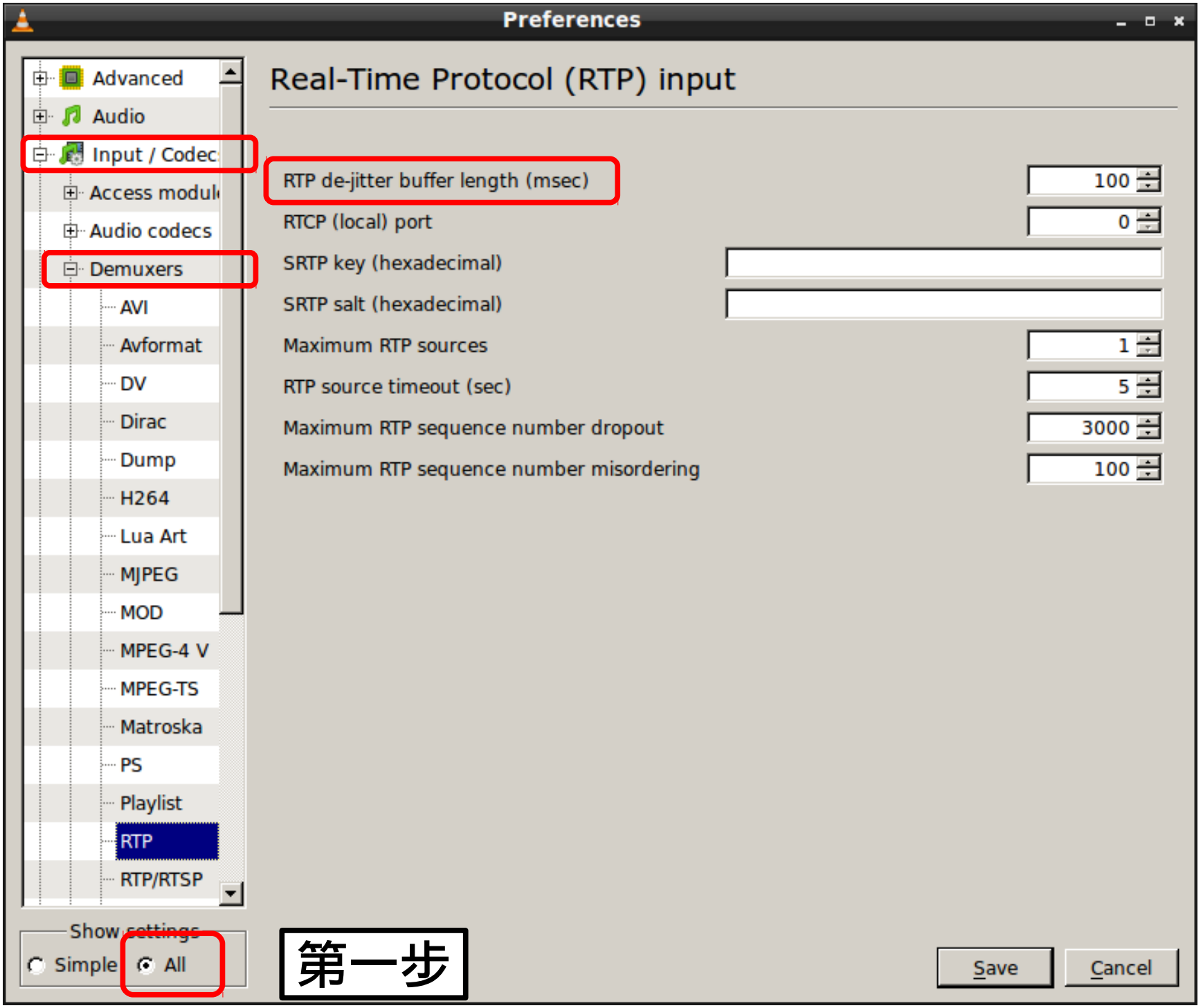


如何降低延遲？

- 方法一：調整 VLC 的參數
  - tools->preferences->all->input/codecs->demuxers->RTP/RTSP->caching value
  - tools->preferences->all->input/codecs->demuxers->RTP->RTP de-jitter buffer length

https://stackoverflow.com/questions/9119106/how-to-reduce-the-delay-vlc-streaming-from-a-web-cam


Preferences
Advanced
Audio
Input / Codec
Access modul
Audio codecs
Demuxers
AVI
Avformat
DV
Dirac
Dump
H264
Lua Art
MJPEG
MOD
MPEG-4 V
MPEG-TS
Matroska
PS
Playlist
RTP
RTP/RTSP
Show settings
Simple
All
RTP/RTSP/SDP demuxer (using Live555)
第二步
P (TCP)
Client port
-1
第三步
TP via RTSP
RTP over HTTP
HTTP tunnel port
80
Caching value (ms)
200
第五步
Kasenna RTSP dialect
RTSP user name
RTSP password
第四步
第一步
Save
Cancel


Preferences
Advanced
Audio
Input / Codec
Access modul
Audio codecs
Demuxers
AVI
Avformat
DV
Dirac
Dump
H264
Lua Art
MJPEG
MOD
MPEG-4 V
MPEG-TS
Matroska
PS
Playlist
RTP
RTP/RTSP
Real-Time Protocol (RTP) input
RTP de-jitter buffer length (msec)
100
RTCP (local) port
0
SRTP key (hexadecimal)
SRTP salt (hexadecimal)
Maximum RTP sources
1
RTP source timeout (sec)
5
Maximum RTP sequence number dropout
3000
Maximum RTP sequence number misordering
100
Show settings
Simple
All
第一步
Save
Cancel

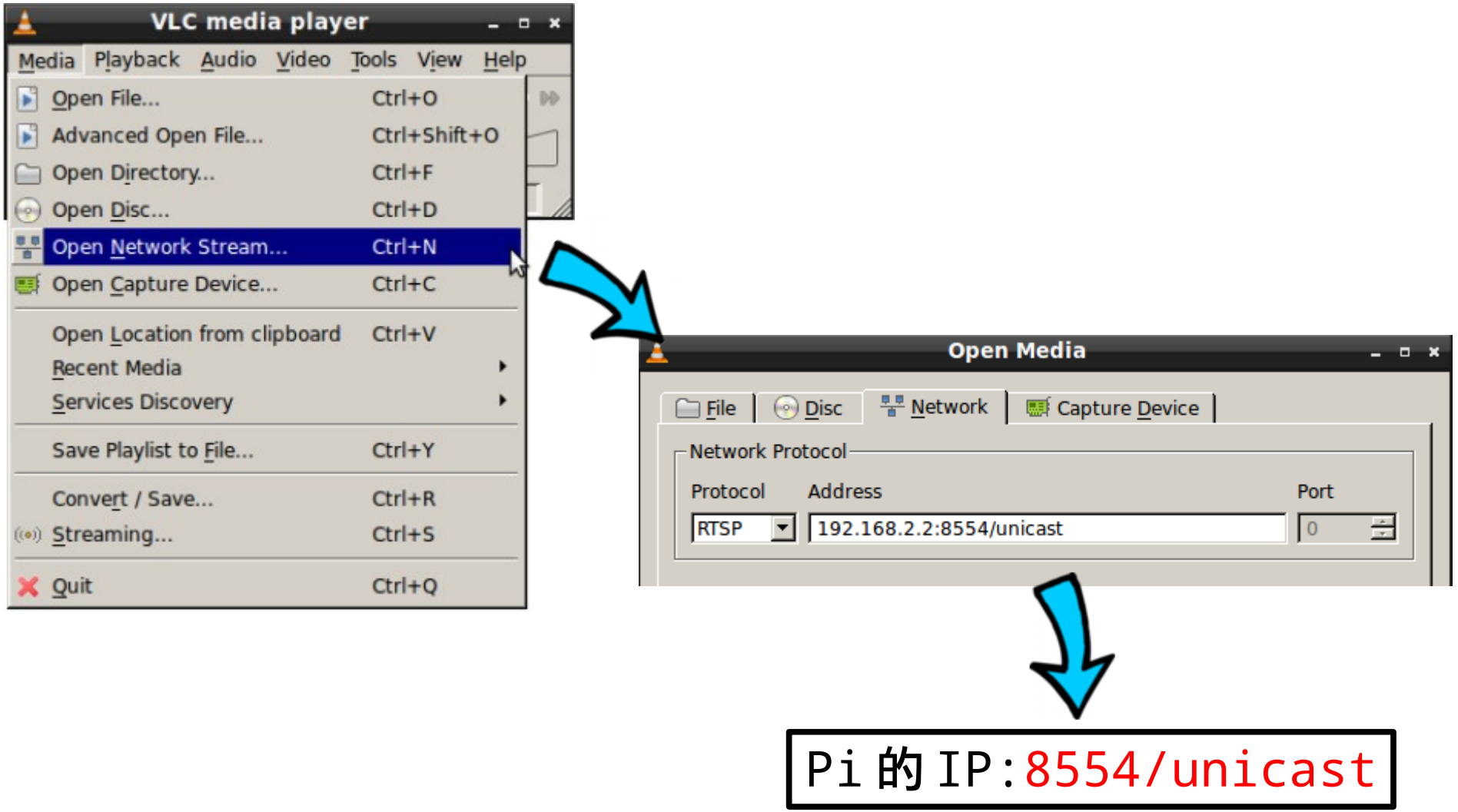
如何降低延遲？

- 方法二：安裝輕量級的RTSP Server(live555)
  - $ cd ~
  - $ git clone https://github.com/mpromonet/h264_v4l2_rtspserver.git
  - $ sudo apt-get install liblivemedia-dev libv4l-dev cmake libpthread-stubs0-dev
  - $ cd h264_v4l2_rtspserver
  - $ cmake .
  - $ make -j4
  - $ sudo ./h264_v4l2_rtspserver -F 15 -W 320 -H 240 -P 8554 /dev/video0

https://hpcc.uk/discussion/30/making-a-raspberry-pi-hd-camera

RTSP Client

- 在 Windows 使用 VLC

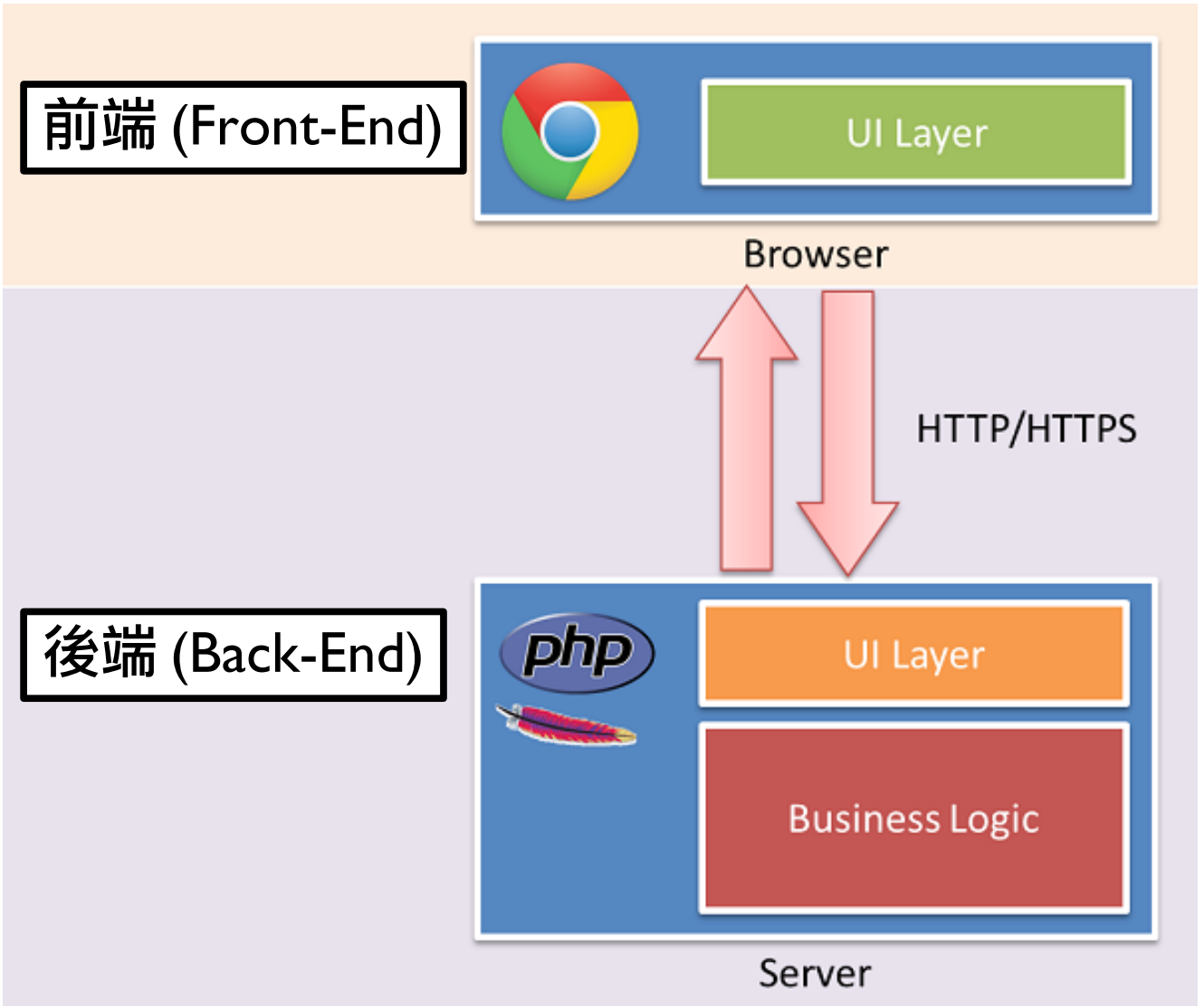


使用 HTTP + MJPG

使用 HTTP 做 Video Streaming

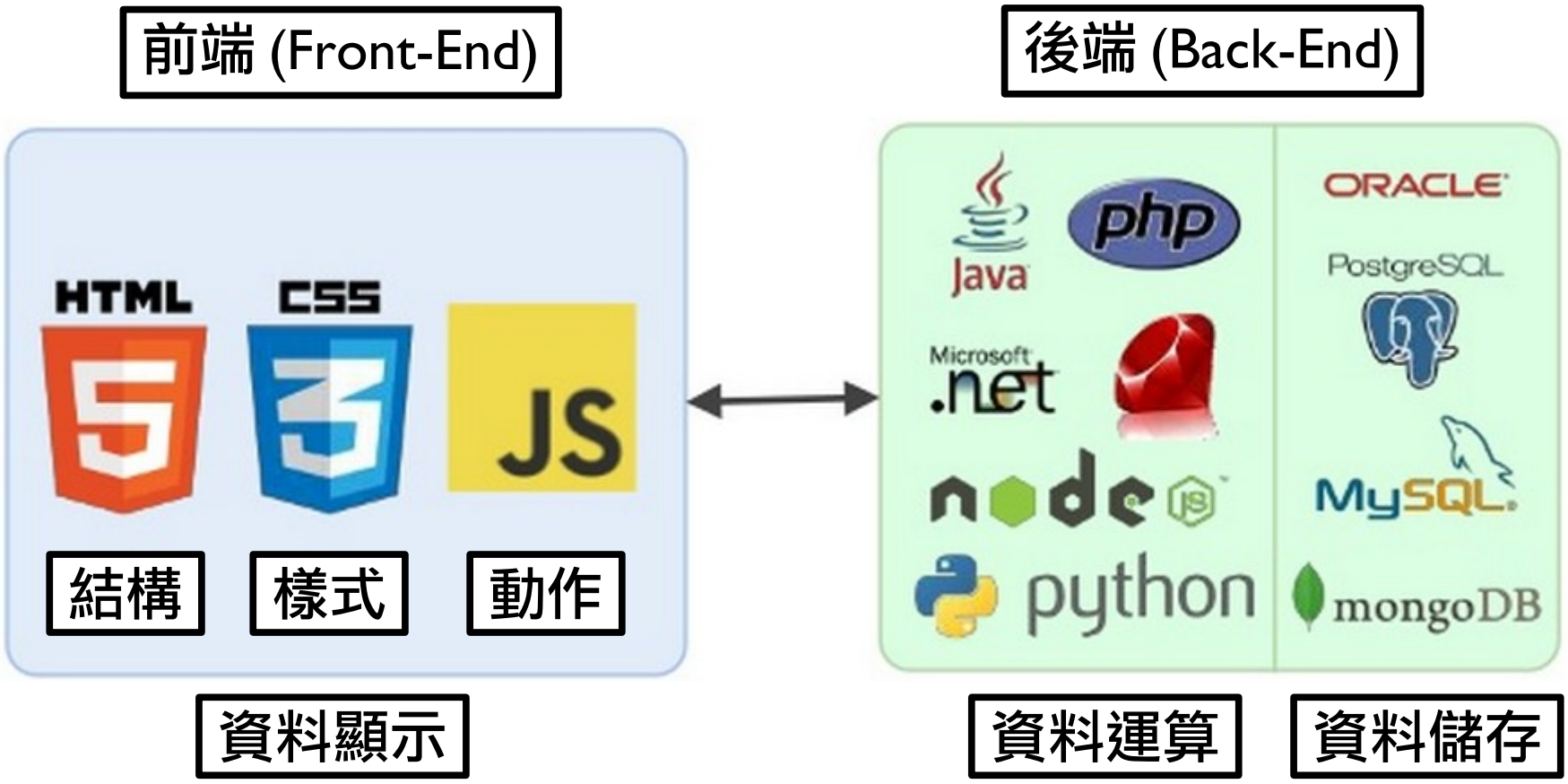
- 原理
  - 向 Web Server 請求一個很大的檔案
  - 該檔案是一個即時的資料

網頁的運作原理



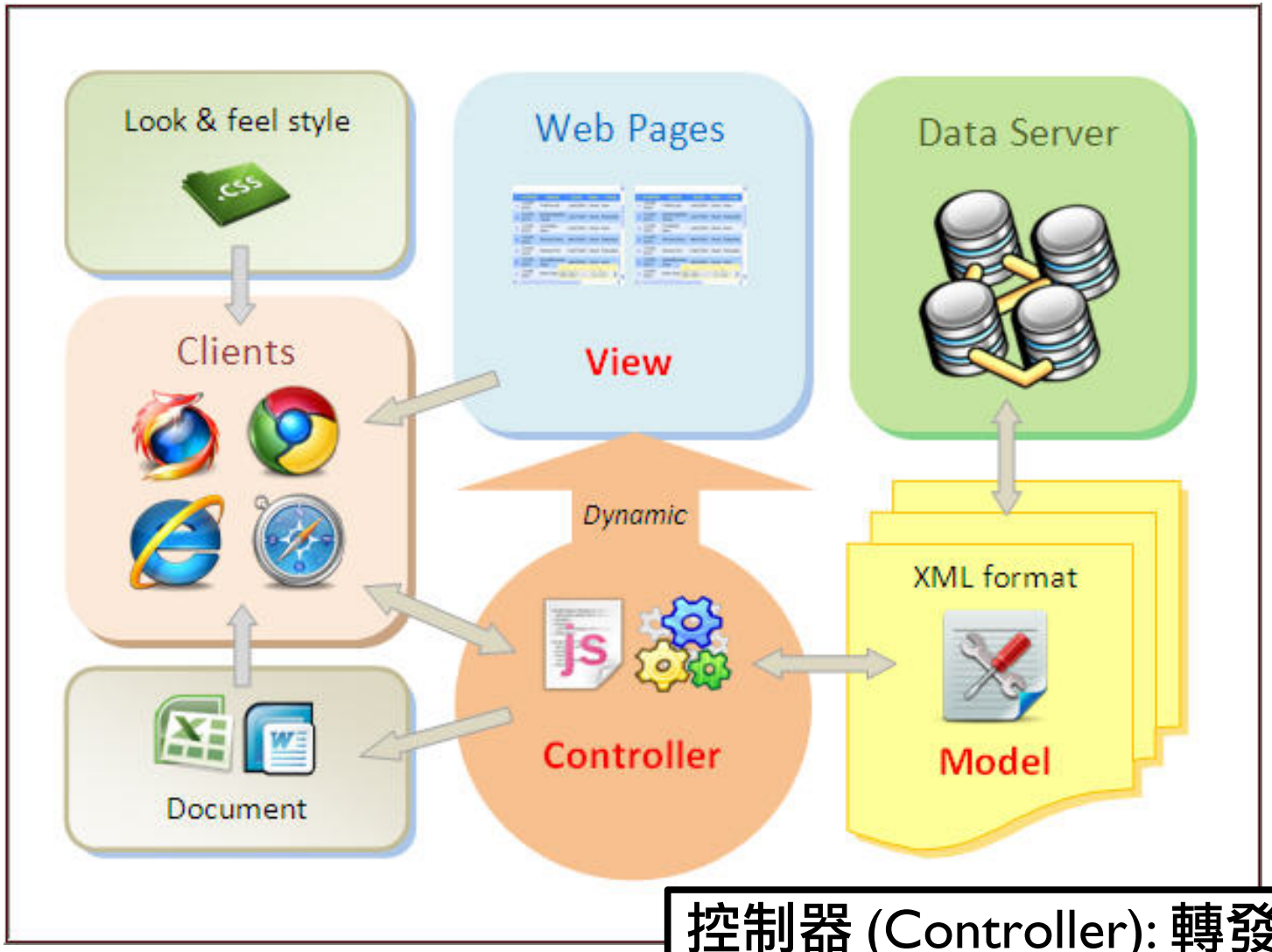
https://www.nczonline.net/blog/2013/10/07/node-js-and-the-new-web-front-end/

可拆解成不同的實做



http://goo.gl/yZj6mH

Model-View-Controller 設計模式

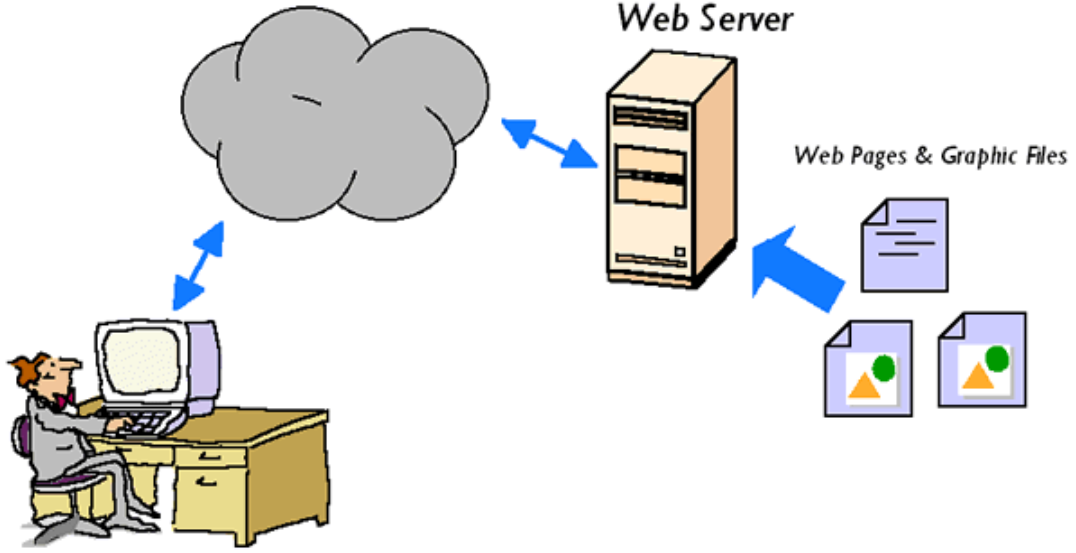


控制器 (Controller): 轉發請求
視圖 (View): 圖形界面顯示
模型 (Model): 實現邏輯與資料儲存

https://webformdeveloping.wordpress.com/

網頁伺服器 (Web Server)

- 是一個軟體
- 回應從 80/8080 port 進來的 HTTP 要求
- 可透過 CGI 或 module 方式擴充
- 如 Apache, Nginx, Boa



http://www.resultantsys.com/index.php/general/what-is-a-web-application-server/

Flask

web development,
one drop at a time

- A Python Microframework

app-hello.py

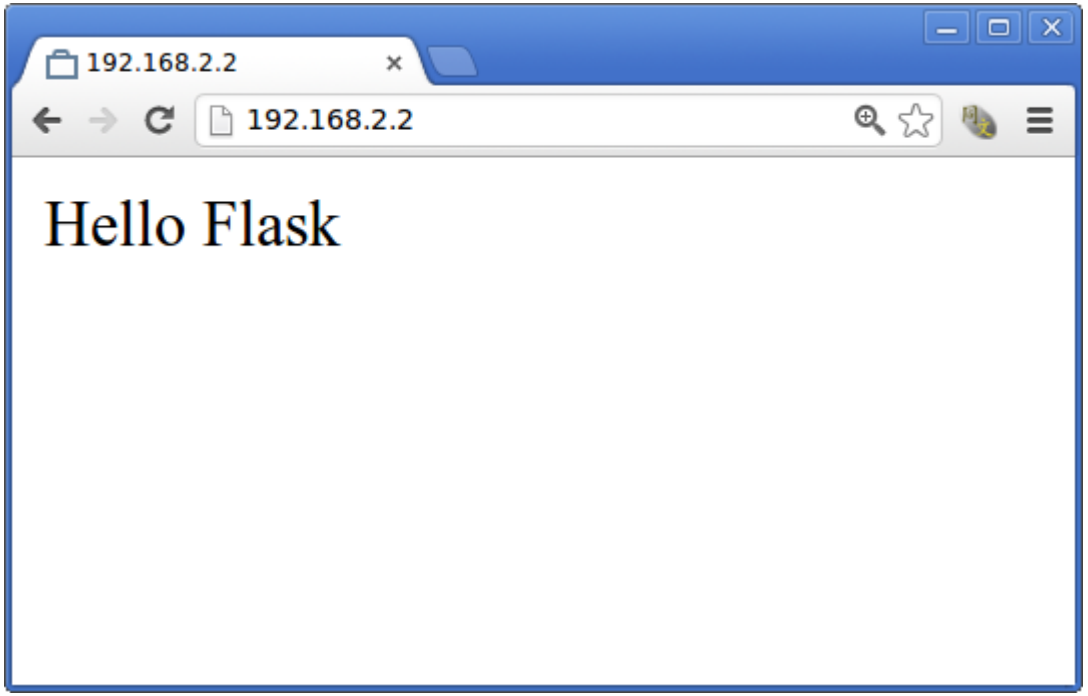
```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def index():
    return "Hello Flask"

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
```

Controller

View

執行

```
$ sudo python app-hello.py
 * Running on http://0.0.0.0:80/
 * Restarting with reloader
192.168.2.1 - - [07/May/2017 15:04:39] "GET / HTTP/1.1" 200 -
192.168.2.1 - - [07/May/2017 15:04:39] "GET /favicon.ico
HTTP/1.1" 404 -
```



DEMO
app-hello.py

```
$ cd ~/cam-py-cv/day1/05-streaming
$ sudo python app-hello.py
```

樣板引擎 (Template Engine)

動態網頁

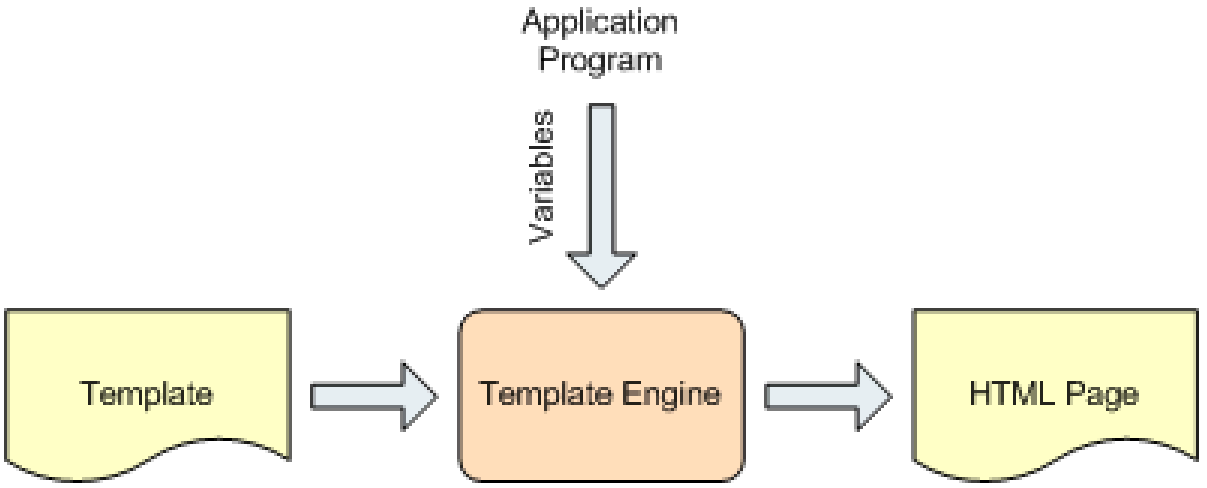
https://tw.stock.yahoo.com

動態網頁

https://tw.stock.yahoo.com

動態網頁

- 是由靜態 HTML 加上動態文字產生
- 樣板引擎將程式碼與使用者介面分離



- Flask 預設使用 Jinja 做為樣板引擎

http://www.source-code.biz/MiniTemplator/

新增一個 route & template

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('link.html')

@app.route("/foo")
def foo():
    extns = ['Flask', 'Jinja2', 'Awesome']
    return render_template('bar.html', extns=extns)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
```

View 可以由樣板產生

樣板可以塞變數

建立 template

```
$ mkdir templates
$ nano templates/link.html
```

```
<h1>Hello Template</h1>
<a href="{{ url_for('foo') }}">foo</a>
```

```
$ nano templates/bar.html
```

```
<ul>
{% for ext in extns %}
<li>{{ ext }}</li>
{% endfor %}
</ul>
```

執行

```
$ sudo python app-route.py
```

DEMO
app-route.py

```
$ cd ~/cam-py-cv/day1/05-streaming
$ sudo python app-route.py
```

Streaming 圖片 (app-stream.py)

Camera 類別

```
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```

回傳內容

```
@app.route('/video_feed')
def video_feed():
    return Response(gen(Camera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
```

回傳型態

建立 Camera 類別 (stream_pi.py)

```
from time import time


class Camera(object):
    def __init__(self):
        self.frames = [open(f + '.jpg', 'rb').read()
for f in ['1', '2', '3']]

    def get_frame(self):
        return self.frames[int(time()) % 3]
```

修改 template

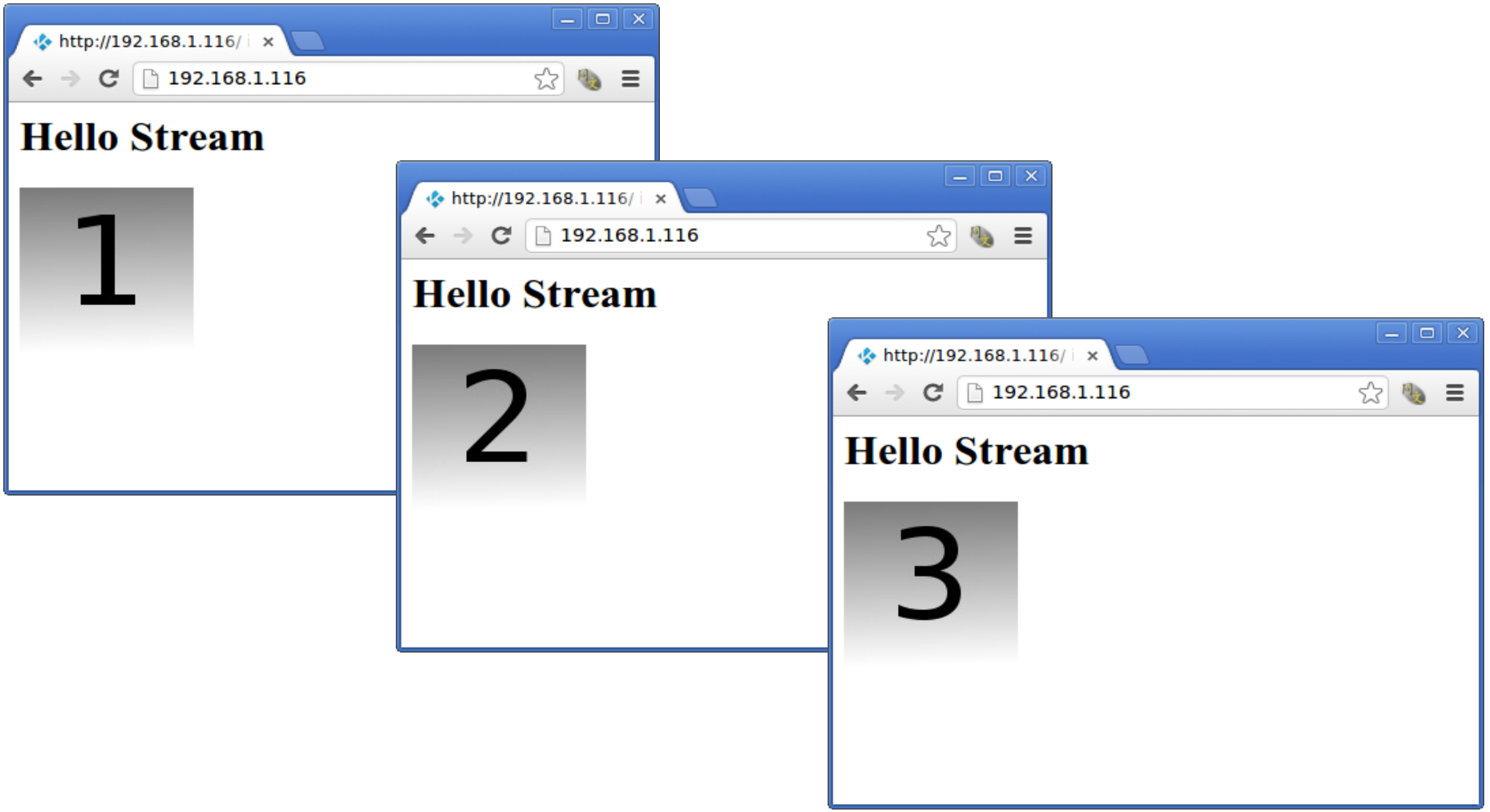
```
$ nano templates/stream.html
```

```
<h1>Hello Stream</h1>
<img id="bg" src="{{ url_for('video_feed') }}">
```

- 補個圖 1.jpg, 2.jpg, 3.jpg

執行

```
$ sudo python app-stream.py
```



DEMO
app-stream.py

```
$ cd ~/cam-py-cv/day1/05-streaming
$ sudo python app-stream.py
```

HTTP + MJPEG

- MJPEG = Motion JPEG
  - 一種視訊壓縮格式
  - 每一個 frame 都使用 JPEG 編碼
  - 對運算能力與記憶體的需求較低

從 Camera 讀取影像 (camera_pi.py)

```
import cv2

class Camera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        ret, jpeg = cv2.imencode('.jpg', image)
        return jpeg.tostring()
```

開啟 /dev/videoX

V4L2 API

使用前記得要先載入模組

```
$ sudo modprobe bcm2835-v4l2
```

不是數字 1，是小寫 L

DEMO

app-camera.py

```
$ cd ~/cam-py-cv/day1/05-streaming
$ sudo python app-camera.py
```

如何降低延遲？ (camera_pi.py)

```
import cv2

class Camera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.video.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH,  320)
        self.video.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, 240)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        ret, jpeg = cv2.imencode('.jpg', image, [1, 50])
        return jpeg.tostring()
```

下 ioctl 改變寬和高

調整 JPG 壓縮品質

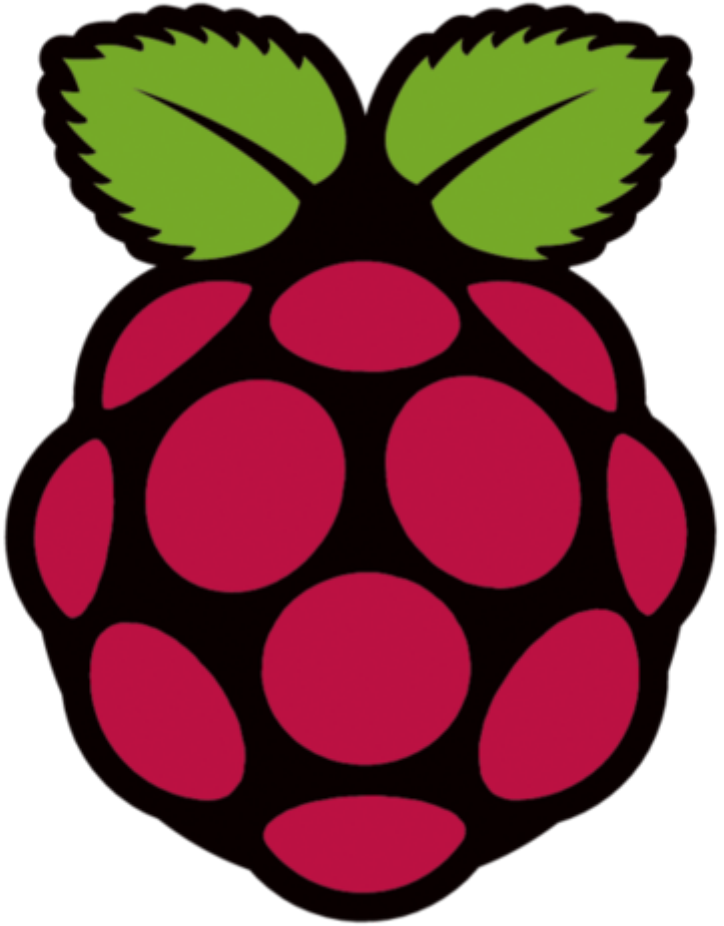
練習

- 使用 OpenCV 套件所提供的函式，讓 Camera 的 Streaming 顯示日期

小結

- 串流服務可使用 RTSP+H.264 或是 HTTP+MJPG 等多種組合
- 使用自己寫的 HTTP 串流，可以在 JPG 丟出去前再進行處理
- 如果要多人連線，可以實做 videocopy 或是在服務前面新增一個 host 儲存目前影像結果，新連線連到 host 後直接回傳儲存結果

Raspberry Pi Rocks the World



Thanks

更多 : 串接 picamera 到 OpenCV

```
import picamera
from picamera.array import PiRGBArray
import time
import cv2

camera = picamera.PiCamera()
camera.resolution = (320, 240)
rawCapture = PiRGBArray(camera)
stream = camera.capture_continuous(rawCapture,
format="bgr",use_video_port=True)

for f in stream:
    frame = f.array
    rawCapture.truncate(0)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
```