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、距離感測器
- 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)

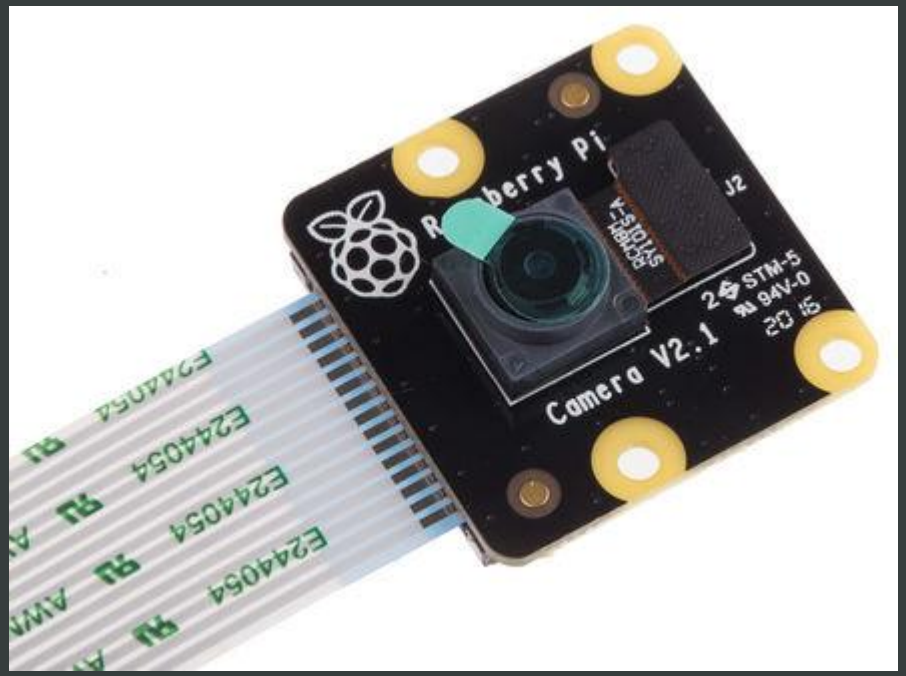
NoIR Camera V2

樹莓派專用 紅外線夜視 800萬像素

https://24h.pchome.com.tw/prod/DSAJ2B-A9007B1NY?fq=/S/DSAJ2B

攝影機規格

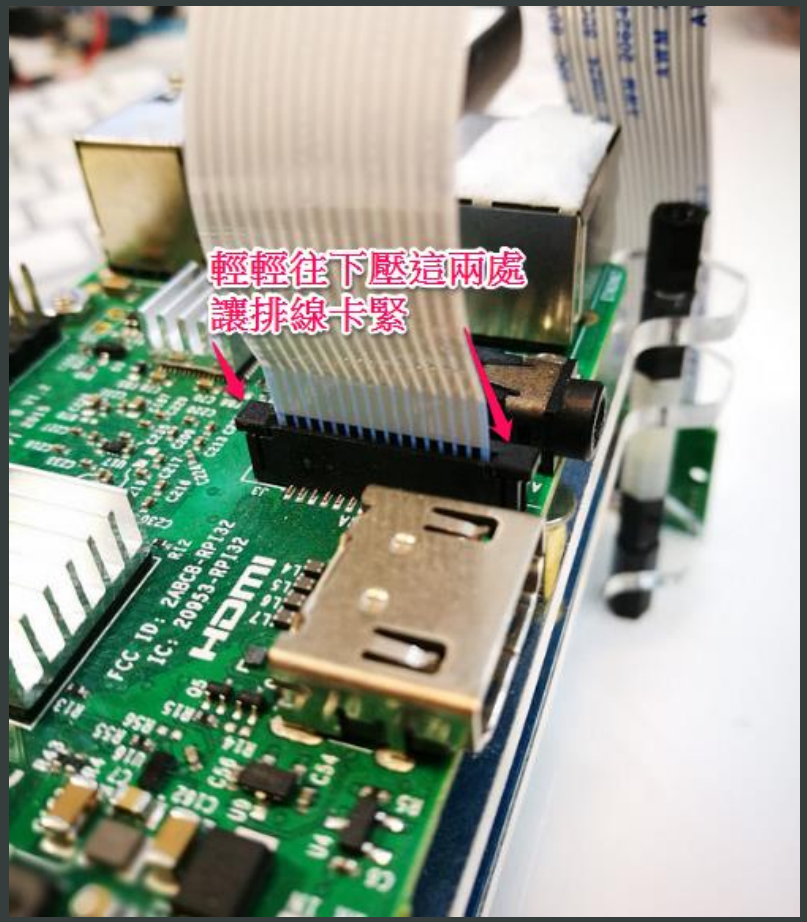
- SONY IMX219感測器
- 800萬像素
- 最高支援 3280 x 2464 靜態拍照
- 支援1080p30, 720p60, 640x480 p60/90
- 支援紅外線感測(可夜間拍攝)
- 使用CSI介面
- 定焦鏡頭



安裝攝影機模組 CSI介面(Camera Serial Interface)

!!!請先斷電後再接攝影機介面!!!

安裝攝影機模組 CSI介面(Camera Serial Interface)



確認安裝方向

測試Rpi camera拍照

- 在SSH輸入以下指令

```
$ raspistill –o img.jpg –t 1000
```

拍攝一張1秒的照片並輸出檔名為img.jpg

預設放置位置在/home/pi底下

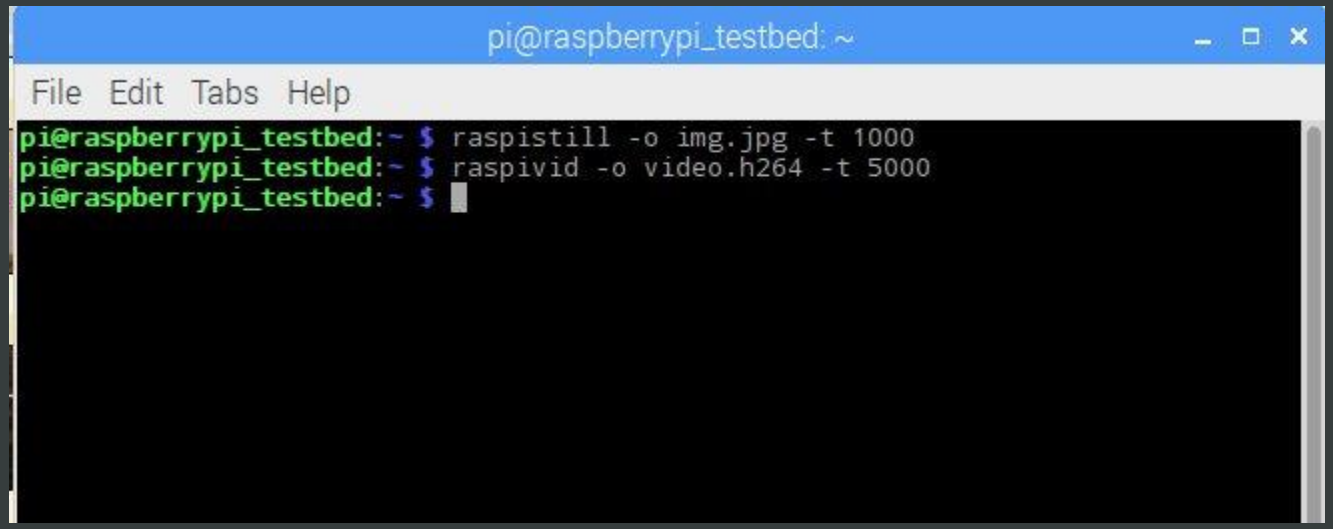
測試Rpi camera錄影

```
$ raspivid –o video.h264 –t 5000
```

錄製一段5秒的影片並輸出檔名為video.h264

```
$ omxplayer –o hdmi video.h264
```

撥放video.h264檔案到hdmi畫面



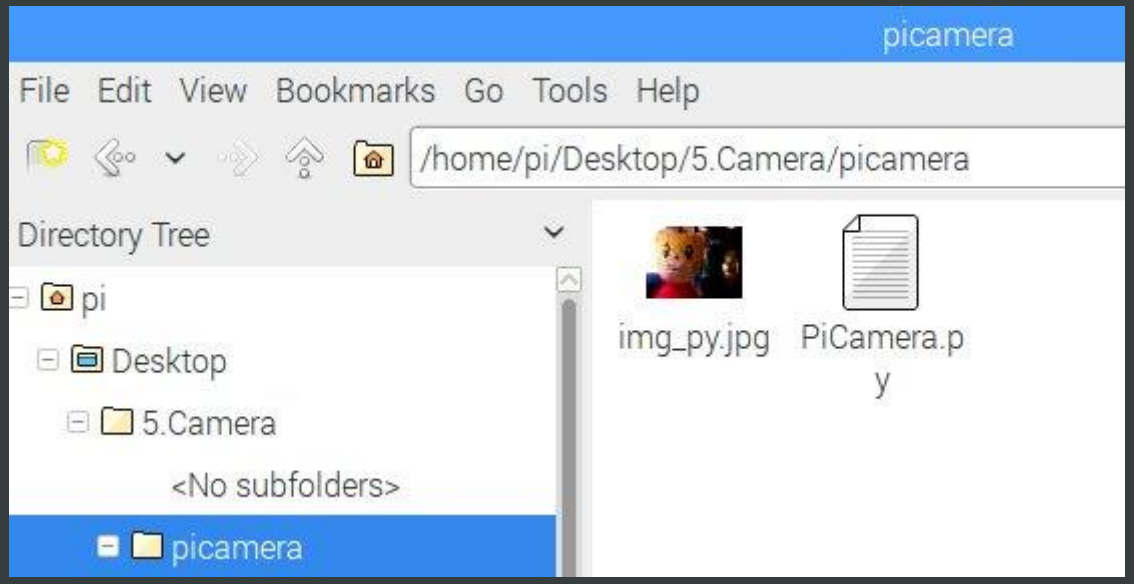
用Python控制Rpi camera

- 使用Picamera套件
- 打開Python中控台來寫一小段測試程式碼

```
import picamera
camera = picamera.PiCamera()
camera.capture('img_py.jpg')
```

```
#匯入picamera套件
#建立camera物件
#調用caputre()，儲存檔名為img_py.jpg
```

PiCamera.py



打開preview觀看預覽拍攝畫面

```
import picamera
camera = picamera.PiCamera()

camera.start_preview()
camera.capture('img.jpg')
camera.stop_preview()
```

```
#匯入picamera套件
#建立camera物件

#顯示preview畫面
#調用caputre()，儲存檔名為img.jpg的圖檔
#結束preview畫面
```

PiCamera_preview.py

顯示畫面會在HDMI螢幕上輸出

常用的PiCamera攝影機調整參數

- 銳利度 camera.sharpness
- 對比 camera.contrast
- 亮度 camera.brightness
- 飽和度 camera.saturation
- ISO camera.ISO
- 防手震 camera.video_stabilization
- 曝光模式 camera.exposure_mode
- AWB camera.awb_mode
- 旋轉 camera.rotation
- 反轉 camera.hflip / camera.vflip
- 影像效果 camera.image_effect
- 顏色效果 camera.color_effects
- 局部切割 camera.crop
- 快門速度 camera.shutter_speed
- 格率 camera.framerate_range

API參考：https://picamera.readthedocs.io/en/release-1.13/

翻轉影像/鏡向影像

```
import picamera
camera = picamera.PiCamera()
#camera.hflip=True
camera.vflip=True
camera.rotation=90
#camera.rotation=180
#camera.rotation=270
#camera.rotation=0
camera.start_preview()
camera.capture('img.jpg')
camera.stop_preview()
```

```
#匯入picamera套件
#建立camera物件
#水平翻轉
#垂直翻轉
#旋轉90度
#旋轉180度
#旋轉270度
#旋轉0度
#顯示preview畫面
#調用caputre()，儲存檔名為img.jpg的圖檔
#結束preview畫面
```

PiCamera_flip.py

影像風格效果

```
import picamera
import time

camera = picamera.PiCamera()

camera.resolution = (640, 480)
camera.start_preview()
camera.image_effect = 'colorswap'
time.sleep(2)
camera.capture('colorswap.jpg')
camera.stop_preview()
```

```
#匯入picamera套件
#匯入time套件



#更改解析度為640,480
#顯示preview畫面
#改變攝影機影像效果為colorswap

#調用caputre()，儲存檔名為colorswap.jpg
#結束preview畫面
```

Image_effect.py

風格效果參數

- none
- negative
- solarize
- sketch
- denoise
- emboss
- oilpaint
- hatch
- gpen
- pastel
- watercolor
- film
- blur
- saturation
- colorswap
- washedout
- posterise
- colorpoint
- colorbalance
- cartoon
- deinterlace1
- deinterlace2

調整攝影機參數

```
import picamera                          #匯入picamera套件
import time                              #匯入time套件

camera = picamera.PiCamera()

camera.resolution = (640, 480)           #更改解析度為640,480
camera.start_preview()
camera.brightness = 30                   #改變攝影機亮度0~100
camera.contrast = 50                     #改變攝影機對比0~100
camera.exposure_mode = 'night'           #改變攝影機曝光模式=晚上

time.sleep(2)                            #暫停2秒

camera.capture('adj.jpg')                #輸出adj.jpg檔案
camera.stop_preview()
```

Camera_adj.py

攝影機參數修改https://picamera.readthedocs.io/en/release-1.13/api_camera.html#

Timelapse / Stop-motion 縮時/定格拍攝

```
import picamera
import time

camera = picamera.PiCamera()
camera.resolution=(640,480)
camera.framerate=30
camera.ISO = 200
time.sleep(2)
camera.shutter_speed = camera.exposure_speed
camera.exposure_mode = 'off‘
g = camera.awb_gains
camera.awb_mode = 'off‘
camera.awb_gains = g

for i in range(0, 10):
    camera.start_preview()
    time.sleep(0.5)
    camera.capture('img%s.jpg' % i)
    camera.stop_preview()
    time.sleep(0.5)
```

```
#匯入picamera套件
#匯入time套件

#建立camera物件
#修改攝影機解析度
#famerate速度
#ISO = 200
#等待攝影機快門速度
#設定快門速度與曝光速度一致
#關閉曝光模式
#白平衡增益
#關閉自動白平衡模式
#白平衡增益重新校正

#連續拍攝10張照片
#顯示preview畫面
#等待0.5秒
#調用caputre()，儲存檔名為img0~9.jpg的圖檔
#結束preview畫面
#等待0.5秒
```

Stop-motion.py

https://vimeo.com/87701971

光軌拍攝

光軌拍攝

```
import picamera
import time
import fractions

camera = picamera.PiCamera()
camera.resolution = (1280, 720)
camera.framerate = fractions.Fraction(1,6)
camera.shutter_speed = 600000
camera.exposure_mode = 'off'
camera.ISO = 100
#time.sleep(10)

camera.capture('dark.jpg')
print('Finished')
```

```
#匯入picamera套件
#匯入time套件
#匯入fractions套件，要引用分數function

#建立camera物件
#修改攝影機解析度 1280 x 720
#famerate速度= 1/6 fps
#快門速度(單位是微秒μs)，曝光6秒
#關閉曝光模式
#ISO = 100 ~ 800
#若要長時曝光則可設定sleep秒數*曝光時間

#儲存檔案為dark.jpg
#印出完成訊息
```

lowLight.py

監視器錄影功能

```
import picamera

camera = picamera.PiCamera()
camera.resolution = (1280, 720)
camera.start_recording('my_video.h264')
camera.wait_recording(10)
camera.stop_recording()
```

```
#匯入picamera套件



#開始錄影，錄製檔名為my_video.h264
#錄影長度，單位為秒，這裡錄製了10秒
#停止錄影
```

Recording_video.py

在ssh內執行，只能在hdmi螢幕上顯示

```
$ omxplayer –o hdmi my_video.h264
```

將h264轉mp4

- 安裝MP4Box套件

```
$ sudo apt-get install gpac
```

- 安裝完成後執行以下指令進行轉檔

```
$ MP4Box –fps 30 –add my_video.h264 my_video.mp4
```

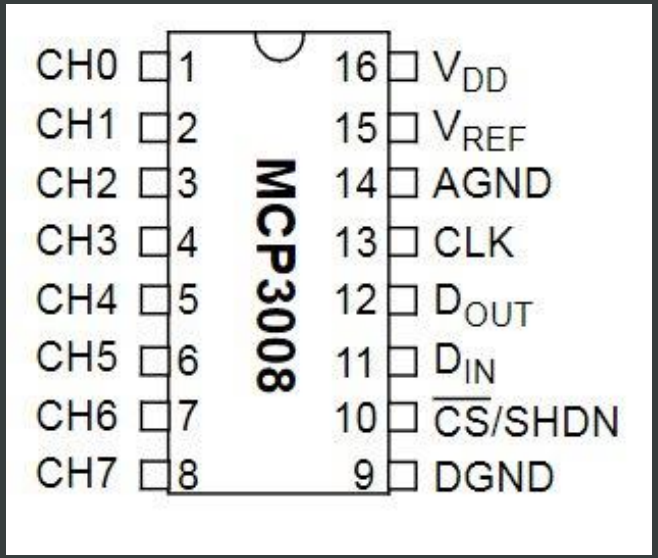
- 用omxplayer播放mp4檔案

```
$ omxplayer –o hdmi my_video.mp4
```

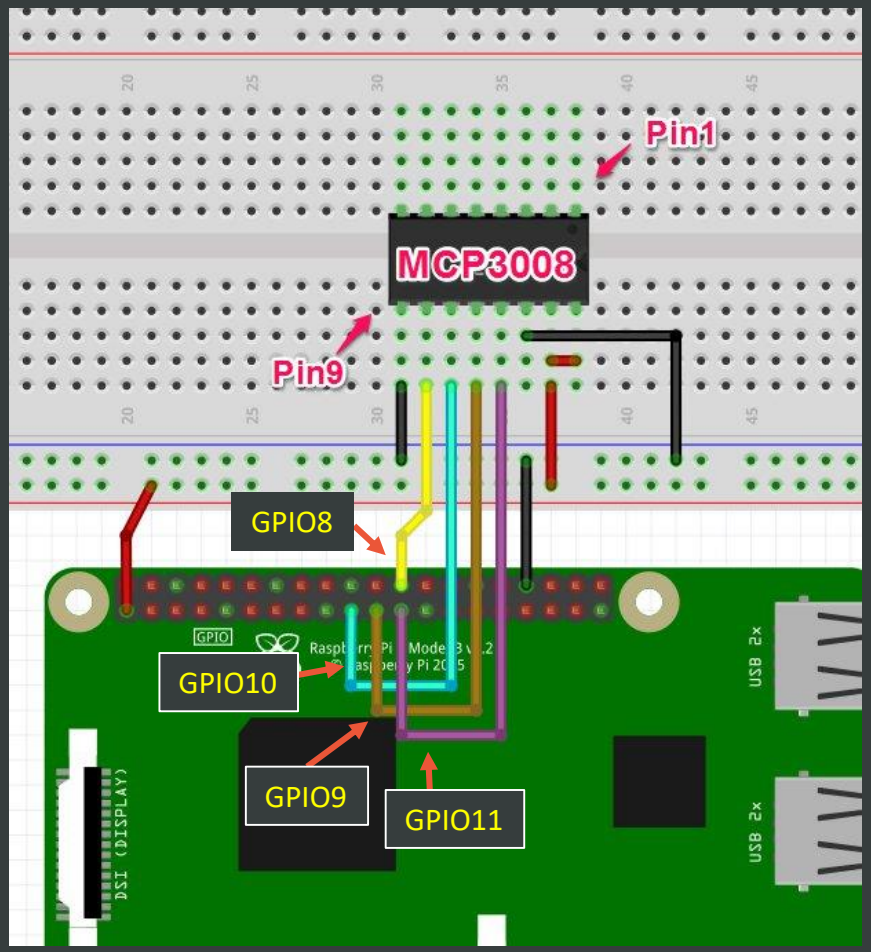
MCP3008

MCP3008

- 10bit ADC 8通道晶片
- 輸入ADC可有8組輸入
- 解析度從0~1023
- 通訊介面SPI
- 可多組並聯串出更多ADC輸入
- 強大的抗躁能力，輸入腳不被其他腳位干擾
- 支援2.7V~5.5V電壓工作

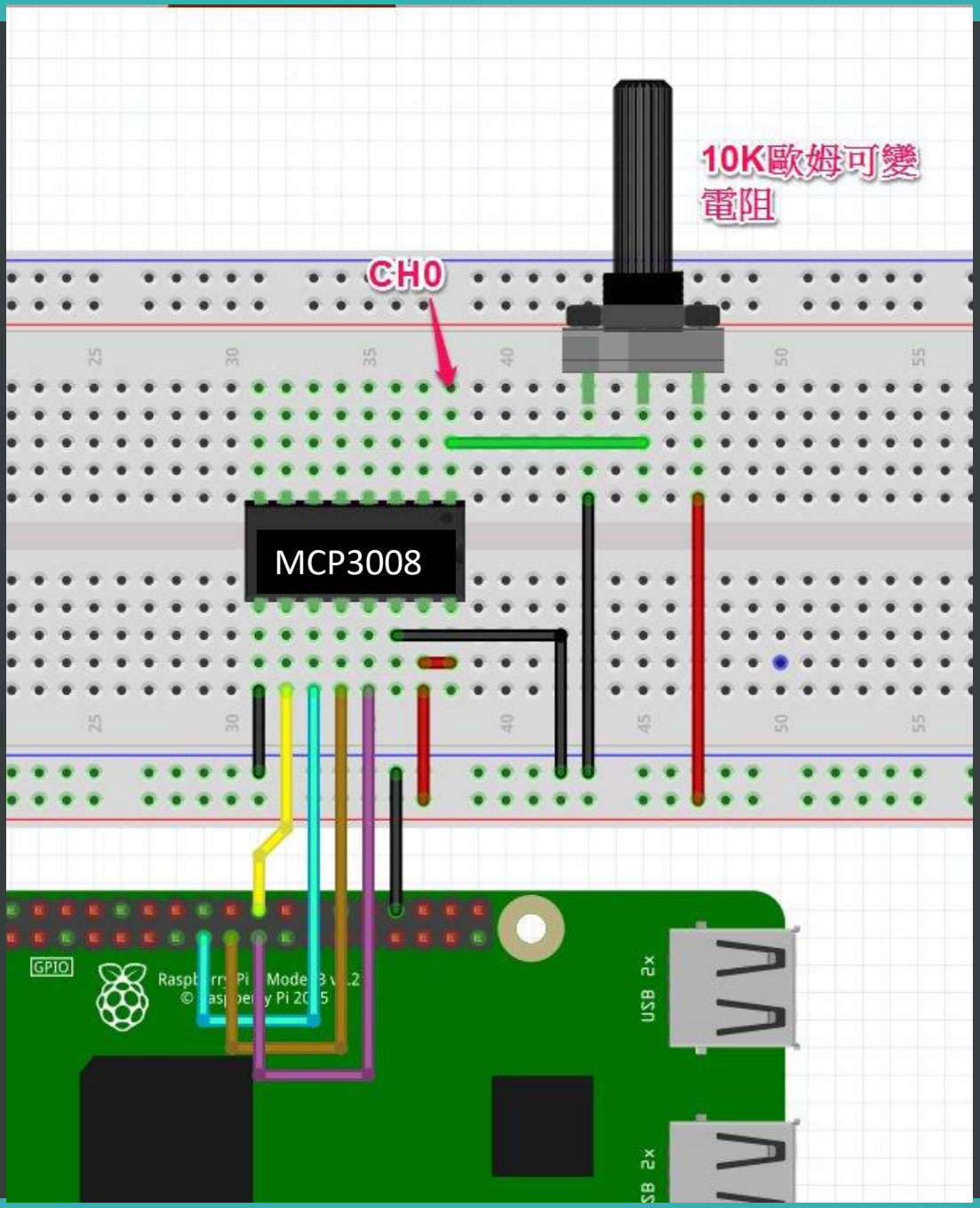


MCP3008接線圖



MCP3008	Raspberry Pi
Pin 9 (GND)	GND
Pin 10 (CS)	GPIO 8 (SPICS0)
Pin 11 (DIN)	GPIO10 (SPIMOSI)
Pin 12 (Dout)	GPIO9 (SPIMISO)
Pin 13 (CLK)	GPIO11 (SPISCLK)
Pin 14 (AGND)	GND
Pin 15 (Verf)	3.3V
Pin 16 (VDD)	3.3V

	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


10K歐姆可變
電阻
CH0
MCP3008
GPIO
USB 2x

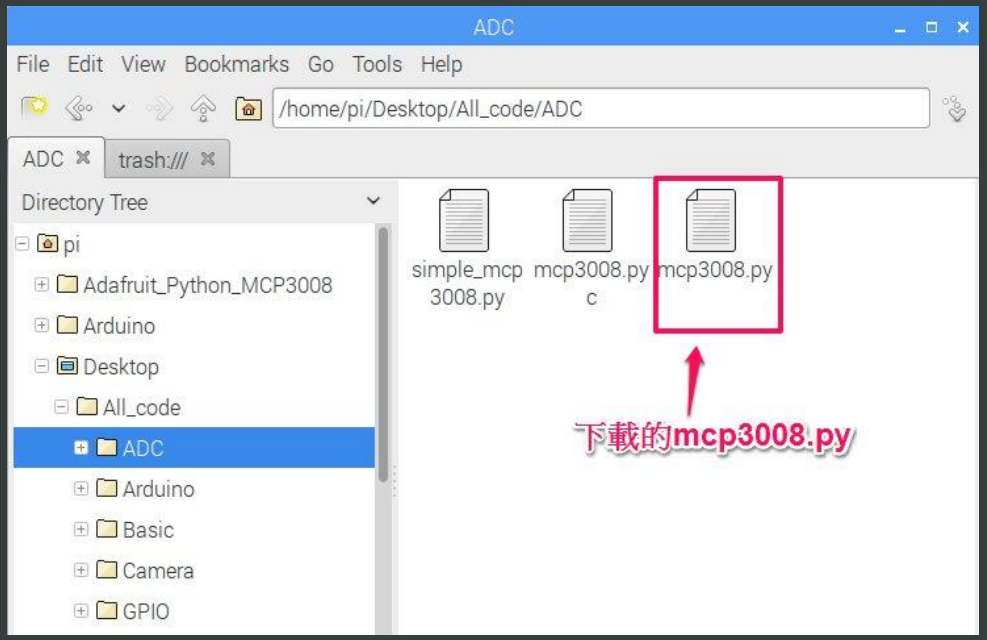
安裝MCP3008程式庫

Step1 – 下載MCP3008程式庫

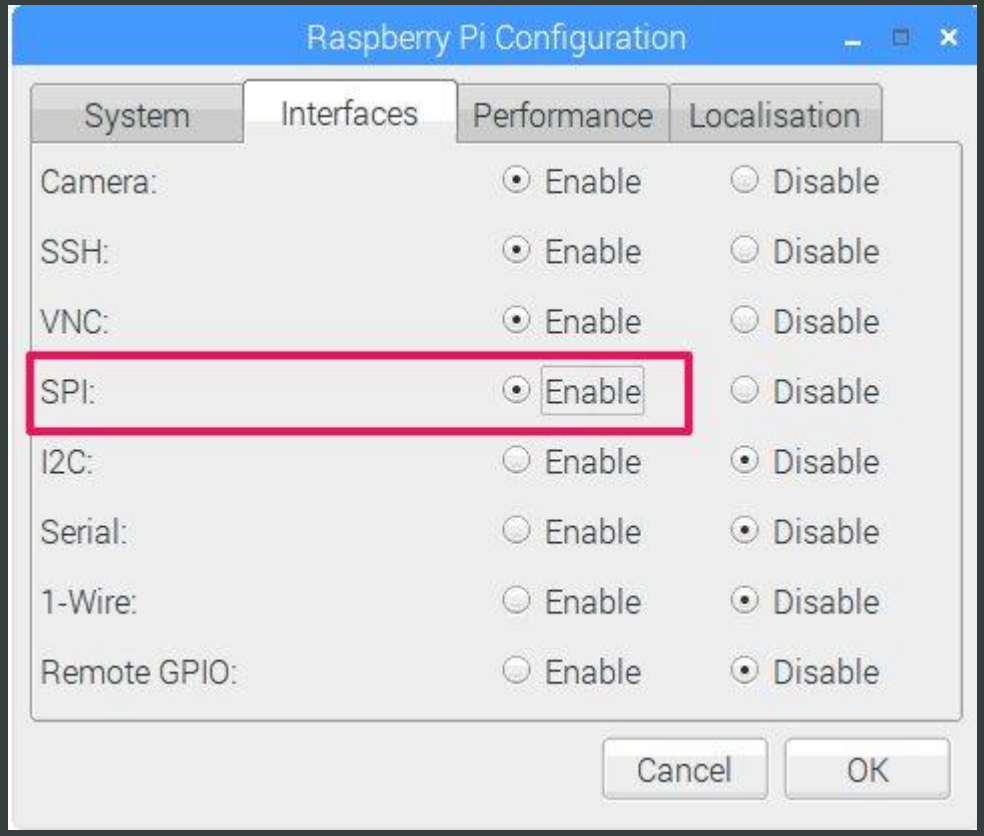
從網站教材下載mcp3008.py

Step2 – 放置MCP3008程式庫到你的資料夾

如同右圖



開啟SPI功能

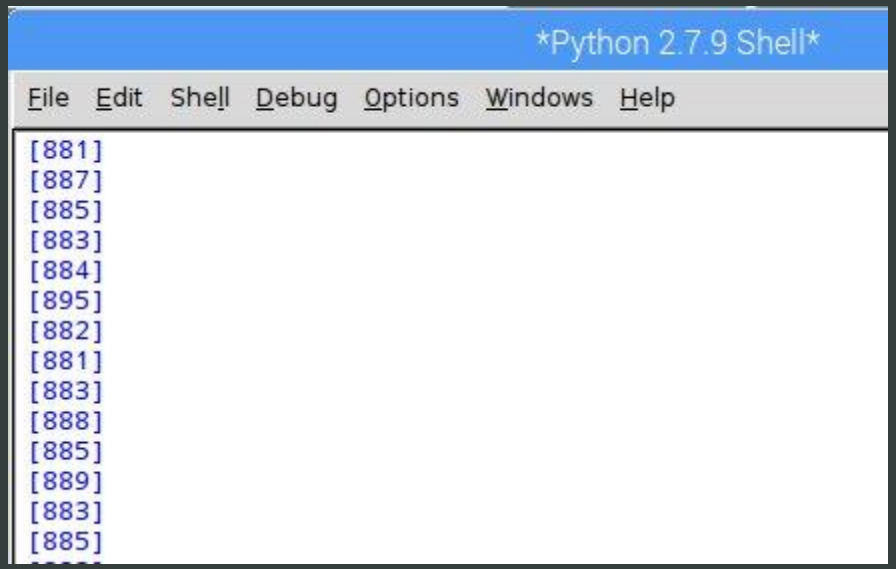

Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera: Enable Disable
SSH: Enable Disable
VNC: Enable Disable
SPI: Enable Disable
I2C: Enable Disable
Serial: Enable Disable
1-Wire: Enable Disable
Remote GPIO: Enable Disable
Cancel
OK

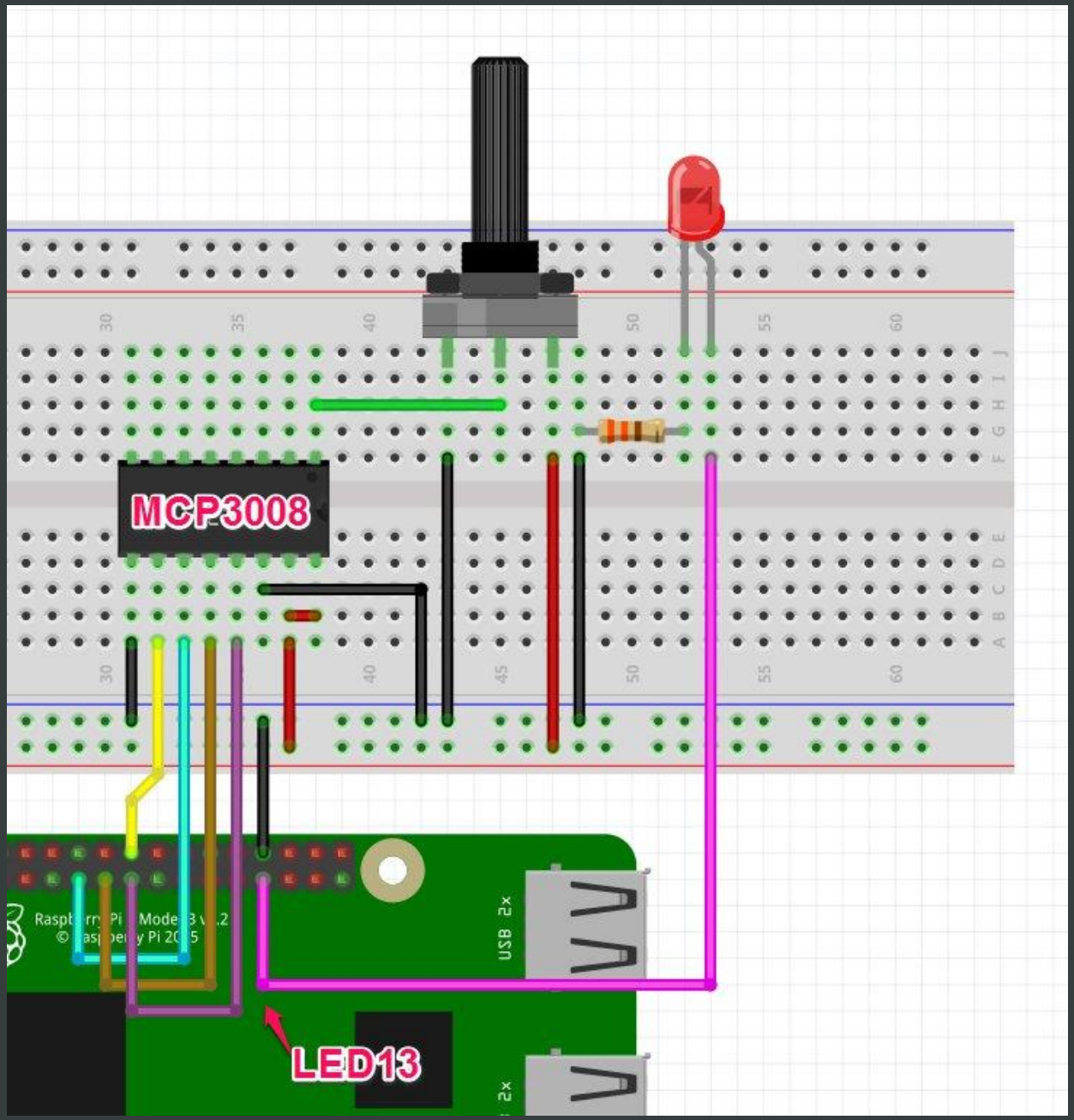
Simple_mcp3008.py

```
import mcp3008
adc = mcp3008.MCP3008()

while True:
    print(adc.read([mcp3008.CH0]))

adc.close()
```

```
#匯入mcp3008模塊
#建立adc物件


#列印出CH0所讀取到的ADC數值

#關閉adc
```

Simple_mcp3008.py



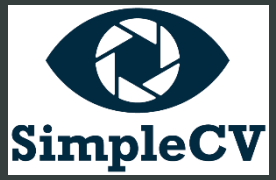

MCP3008
LED13
USB 2x

```
import mcp3008                                  #匯入mcp3008模塊
import RPi.GPIO as GPIO                         #匯入GPIO

led = 13                                        #LED的pin 13
adc = mcp3008.MCP3008()                         #建立adc物件
adcValue = 0                                    #宣告adcValue

GPIO.setmode(GPIO.BCM)
GPIO.setup(led, GPIO.OUT)                       #設定GPIO為輸出

while True:
    adcValue = adc.read([mcp3008.CH0])          #將讀取到CH0的數值給adcValue
    print(adc.read([mcp3008.CH0]))              #印出數值

    if adcValue[0] > 500:                       #判別adcValue[0]元組如果大於500
        GPIO.output(led, True)                  #讓LED亮起
    else:
        GPIO.output(led, False)                 #讓LED熄滅

adc.close()                                     #關閉adc
```

Simple_mcp3008_withLED.py

SimpleCV/OpenCV

SimpleCV / OpenCV



- Simple CV是OpenCV的python簡化版本，原有的OpenCV功能更強大且複雜，OpenCV支援多種程式語言如Java,C++,
- CV即Computer Vision電腦視覺，主要目的是讓機器去［看］目標物並進行識別、追蹤、測量或是圖像處理，以取代人眼觀察並傳送資訊給電腦的一門資訊科學。
- SimpleCV能處理人臉辨識、圖形處理、影像追蹤、編修圖片等功能
- OpenCV能處理複雜的像素與空間轉換，相關資源較多
- 這裡教學的版本為：
  - SimpleCV 1.3.0　/　OpenCV 2.4.9.1

安裝SimpleCV與OpenCV

Step1 – 安裝OpenCV相關套件

```
$ sudo apt-get install ipython python-opencv python-scipy
```

```
pi@raspberrypi_testbed:~ $ sudo apt-get install ipython python-opencv python-scipy
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libopencv-photo2.4 libpython-dev libpython2.7-dev python-decorator python-dev python-imaging python-pexpect python-simplegeneric python2.7-dev
Suggested packages:
  ipython-doc ipython-notebook ipython-qtconsole python-matplotlib python-zmq python-pexpect-doc
The following NEW packages will be installed:
  ipython libopencv-photo2.4 libpython-dev libpython2.7-dev python-decorator python-dev python-imaging python-opencv python-pexpect python-scipy
  python-simplegeneric python2.7-dev
0 upgraded, 12 newly installed, 0 to remove and 151 not upgraded.
Need to get 26.6 MB of archives.
After this operation, 59.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Step2 – 下載安裝SimpleCV套件

```
$ sudo pip install https://github.com/sightmachine/SimpleCV/zipball/develop
```

```
pi@raspberrypi_testbed:~ $ sudo pip install https://github.com/sightmachine/SimpleCV/zipball/develop
Downloading/unpacking https://github.com/sightmachine/SimpleCV/zipball/develop
  Downloading develop (unknown size): 53.7MB downloaded
  Running setup.py (path:/tmp/pip-h3MJIm-build/setup.py) egg_info for package from https://github.com/sightmachine/SimpleCV/zipball/develop

Installing collected packages: SimpleCV
  Running setup.py install for SimpleCV

    Installing simplecv script to /usr/local/bin
Successfully installed SimpleCV
Cleaning up...
pi@raspberrypi_testbed:~ $
```

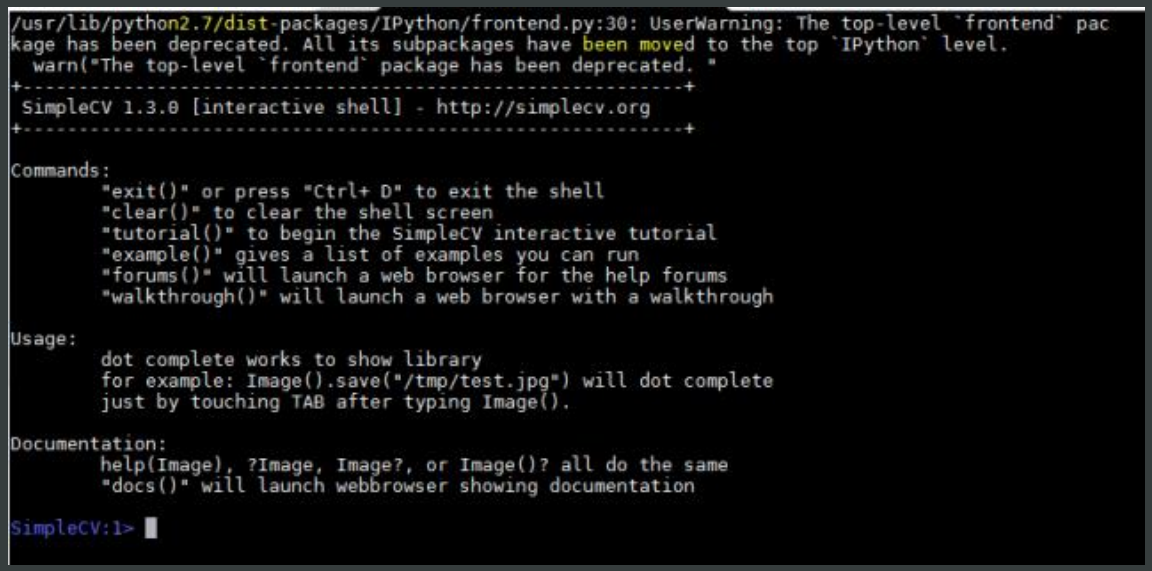
安裝SimpleCV相關套件與測試

Step3 – 安裝缺少的svgwrite套件

```
$ sudo pip install svgwrite
```

Step4 – 執行simplecv

```
$ simplecv
```



測試SimpleCV

Step5 – 測試simplecv 發現缺少/dev/video0這個攝影機裝置

```
SimpleCV:1>cam = Camera()
```

```
SimpleCV:1> cam = Camera()
/bin/sh: 1: lsof: not found
[0]
WARNING: caught exception: SystemError("Cannot identify '/dev/video0': 2, No such file or directory",)
WARNING: SimpleCV can't seem to find a camera on your system, or the drivers do not work with SimpleCV.
```

Step6 – 檢視camera裝置

```
SimpleCV:2>exit

$ ls /dev/
```

```
pi@raspberrypi_testbed:~ $ ls /dev/
autofs           loop2          ram12     tty0   tty31  tty54      vcs
block            loop3          ram13     tty1   tty32  tty55      vcs1
btrfs-control    loop4          ram14     tty10  tty33  tty56      vcs2
bus              loop5          ram15     tty11  tty34  tty57      vcs3
cachefiles       loop6          ram2      tty12  tty35  tty58      vcs4
char             loop7          ram3      tty13  tty36  tty59      vcs5
console          loop-control   ram4      tty14  tty37  tty6       vcs6
cpu_dma_latency  mapper         ram5      tty15  tty38  tty60      vcs7
cuse             mem            ram6      tty16  tty39  tty61      vcsa
disk             memory_bandwidth ram7     tty17  tty4   tty62      vcsa1
fb0              mmcblk0        ram8      tty18  tty40  tty63      vcsa2
fd               mmcblk0p1      ram9      tty19  tty41  tty7       vcsa3
full             mmcblk0p2      random    tty2   tty42  tty8       vcsa4
fuse             mqueue         raw       tty20  tty43  tty9       vcsa5
gpiomem          net            rfkill    tty21  tty44  ttyAMA0    vcsa6
hidraw0          network_latency serial0  tty22  tty45  ttyprintk  vcsa7
hidraw1          network_throughput serial1 tty23 tty46  ttyS0      vcsm
hwrng            null           shm       tty24  tty47  uhid       vhci
i2c-1            ppp            snd       tty25  tty48  uinput     watchdog
initctl          ptmx           spidev0.0 tty26  tty49  urandom    watchdog0
input            pts            spidev0.1 tty27  tty5   usb        xconsole
kmsg             ram0           stderr    tty28  tty50  vc-cma     zero
log              ram1           stdin     tty29  tty51  vchiq
loop0            ram10          stdout    tty3   tty52  vcio
loop1            ram11          tty       tty30  tty53  vc-mem
```

測試v4l2

Step7 – 啟動v4l2

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

Step8 – 檢視camera裝置

```
$ ls /dev/
```

!!確認要有video0 這個裝置!!

```
pi@raspberrypi_testbed:~ $ ls /dev/
autofs           loop2            ram12    tty0   tty31  tty54    vcs
block            loop3            ram13    tty1   tty32  tty55    vcs1
btrfs-control    loop4            ram14    tty10  tty33  tty56    vcs2
bus              loop5            ram15    tty11  tty34  tty57    vcs3
cachefiles       loop6            ram2     tty12  tty35  tty58    vcs4
char             loop7            ram3     tty13  tty36  tty59    vcs5
console          loop-control     ram4     tty14  tty37  tty6     vcs6
cpu_dma_latency  mapper           ram5     tty15  tty38  tty60    vcs7
cuse             mem              ram6     tty16  tty39  tty61    vcsa
disk             memory_bandwidth ram7     tty17  tty4   tty62    vcsa1
fb0              mmcblk0          ram8     tty18  tty40  tty63    vcsa2
fd               mmcblk0p1        ram9     tty19  tty41  tty7     vcsa3
full             mmcblk0p2        random   tty2   tty42  tty8     vcsa4
fuse             mqueue           raw      tty20  tty43  tty9     vcsa5
gpiomem          net              rfkill   tty21  tty44  ttyAMA0  vcsa6
hidraw0          network_latency  serial0  tty22  tty45  ttyprintk vcsa7
hidraw1          network_throughput serial1 tty23 tty46  ttyS0    vcsm
hwrng            null             shm      tty24  tty47  uhid     vhci
i2c-1            ppp              snd      tty25  tty48  uinput   video0
initctl          ptmx             spidev0.0 tty26 tty49  urandom  watchdog
input            pts              spidev0.1 tty27 tty5   usb      watchdog0
kmsg             ram0             stderr   tty28  tty50  vc-cma   xconsole
log              ram1             stdin    tty29  tty51  vchiq    zero
loop0            ram10            stdout   tty3   tty52  vcio
loop1            ram11            tty      tty30  tty53  vc-mem
pi@raspberrypi_testbed:~ $
```

設定bcm2835-v4l2驅動模組到開機自動執行

Step9 – 設置modules
```
$ sudo vim /etc/modules
```

Step10 – 增加modules
進入vim文字編輯器後，在最後一行
新增以下模組名稱
bcm2835-v4l2

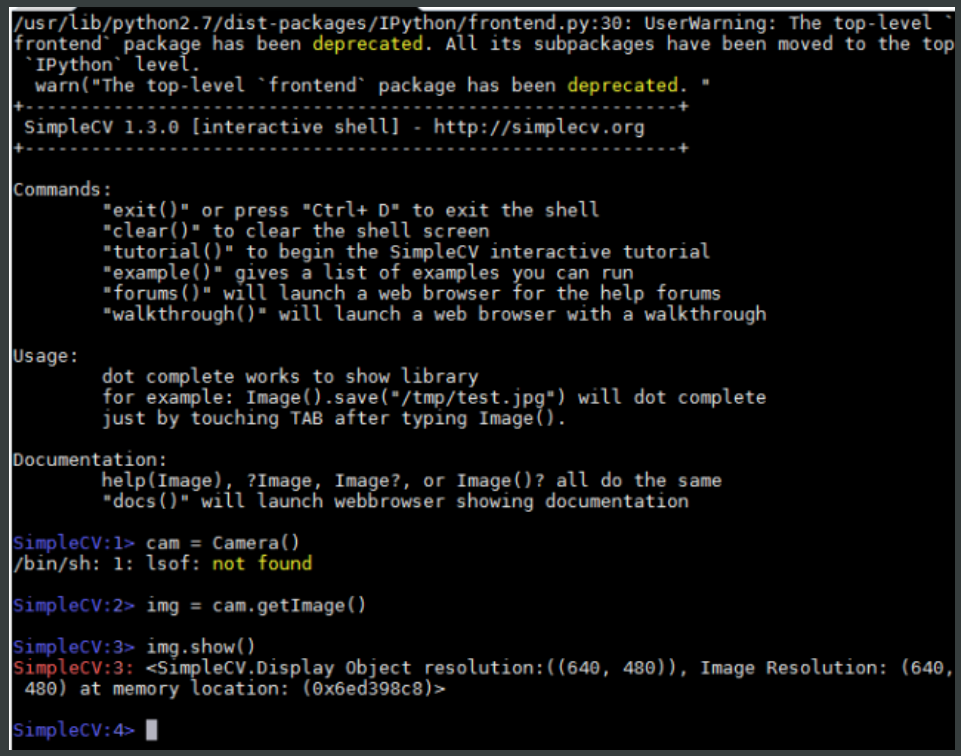
Step11 – 重新開機
```
$ sudo reboot
```

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

i2c-dev
cuse
bcm2835-v4l2
~
```

測試SimpleCV

重新測試simplecv

```
SimpleCV:1>cam = Camera()
SimpleCV:2>img = cam.getImage()
SimpleCV:3>img.show()

SimpleCV:4>exit 離開
```



SimpleCV入門

```
import SimpleCV
cam = SimpleCV.Camera()

while True:
    img = cam.getImage()
    img.show()
```

```
#匯入SimpleCV套件
#建立cam物件


#抓取攝影機畫面
#顯示畫面到視窗
```

simpleCV_basic.py

SimpleCV - Display

```
import SimpleCV

cam = SimpleCV.Camera()
disp = SimpleCV.Display()

while disp.isNotDone():
    img = cam.getImage()
    img.show()
```

simpleCV_display.py

```
#匯入SimpleCV套件

#建立cam物件
#建立一個disp物件

#當display還沒被關閉時
#抓取攝影機畫面
#顯示畫面到視窗
```

SimpleCV – 影像縮放

```
import SimpleCV

cam = SimpleCV.Camera()
disp = SimpleCV.Display()

while disp.isNotDone():
    img = cam.getImage()
    small = img.scale(320,240)
    small.show()
```

simpleCV_scale.py

```
#匯入SimpleCV套件

#建立cam物件
#建立一個disp物件

#當display還沒被關閉時
#抓取攝影機畫面
#將抓取的畫面縮小
#顯示畫面到視窗
```

SimpleCV – 影像儲存

```
import SimpleCV

cam = SimpleCV.Camera()
disp = SimpleCV.Display()

while disp.isNotDone():
    img = cam.getImage()
    img.save('img.jpg')
    img.show()
```

```
#匯入SimpleCV套件

#建立cam物件
#建立一個disp物件

#當display還沒被關閉時
#抓取攝影機畫面
#將抓取的畫面儲存成img.jpg檔
#顯示畫面到視窗
```

simpleCV_save.py

SimpleCV – 攝影機解析度

simpleCV_setCamera.py

```
import SimpleCV

cam = SimpleCV.Camera(0, {"width":320,"height":240})
disp = SimpleCV.Display((320,240))

while disp.isNotDone():
    img = cam.getImage()
    img.show()
```

```
#匯入SimpleCV套件

#建立cam物件，設定大小320,240，第一台相機
#建立一個disp物件，設定視窗大小320,240

#當display還沒被關閉時
#抓取攝影機畫面
#顯示畫面到視窗
```

SimpleCV更多學習資源

- http://simplecv.org/
- http://simplecv.sourceforge.net/doc/cookbook.html
- https://github.com/sightmachine/SimpleCV/tree/master/SimpleCV

OpenCV – Slit Scan

https://www.youtube.com/watch?v=PJZeeKwnTIY

小作業3

- 嘗試用感測器元件或電子元件(按鈕開關、可變電阻、距離感測器、溫度感測器)與PiCamera或SimpleCV進行整合出一個有趣的互動內容。