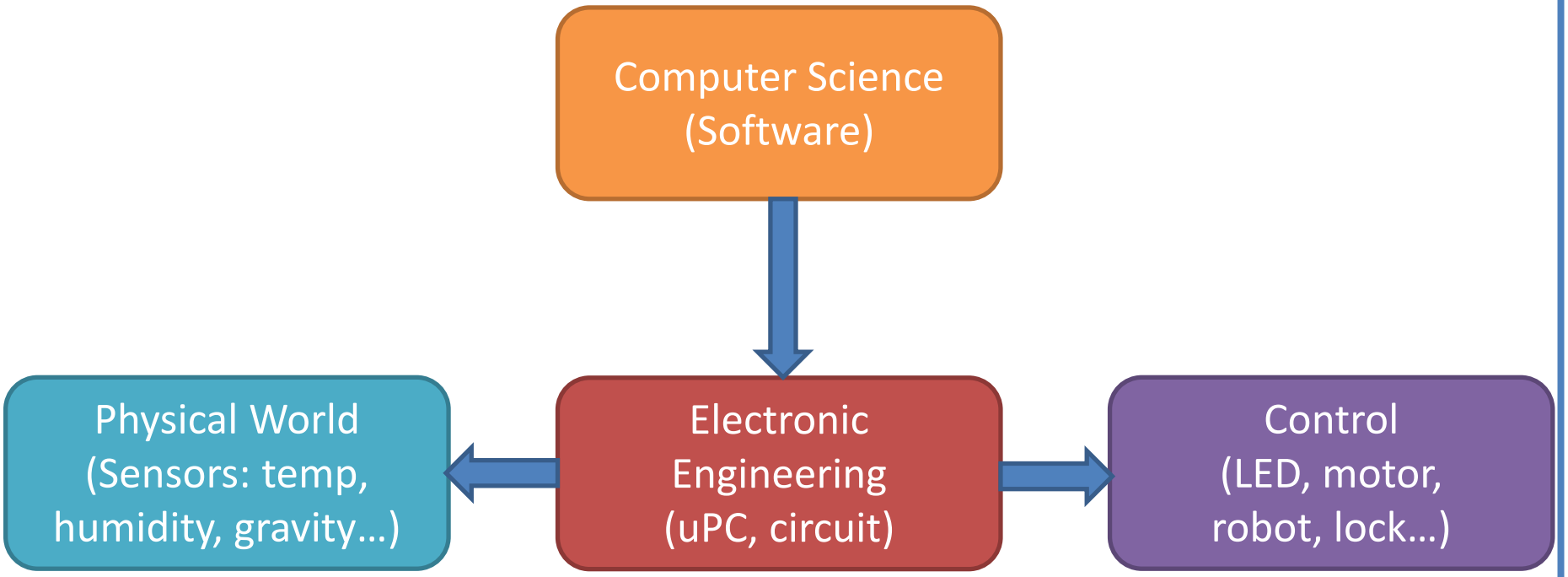
- **本培訓課程相關資訊: https://bit.ly/2CpZW5Z**

- 在開始前, 請先下載Raspbian檔案
  - ubuntu-tw 分流下載
    - http://ftp.ubuntu-tw.org/mirror/raspbian-downloads/raspbian_full/images/raspbian_full-2018-11-15/2018-11-13-raspbian-stretch-full.zip
  - 交大實驗室分流
    - http://140.113.193.13:7788/raspbian.zip
    - http://140.113.144.127/raspbian.zip
- 這些網址在縮網址的文件都有提供!!

物聯網平台Raspberry PI 種子教師培訓

2019.01.16

Aim of This Course



Internet of Things

From connecting devices to human value

1 **Device connection**

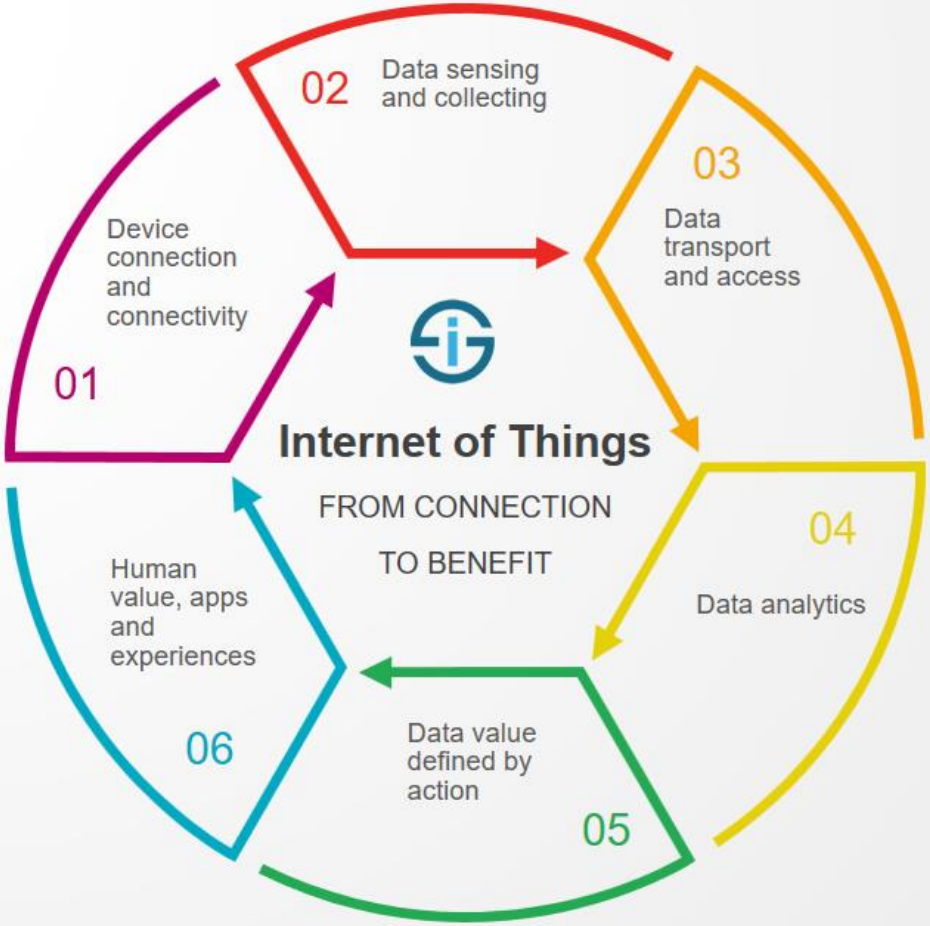
IoT devices
IoT connectivity
Embedded intelligence

2 **Data sensing**

Capture data
Sensors and tags
Storage

3 **Communication**

Focus on access
Networks, cloud, edge
Data transport



4 **Data analytics**

Big data analysis
AI and cognitive
Analyis at the edge

5 **Data value**

Analysis to action
APIs and processes
Actionable intelligence

6 **Human value**

Smart applications
Stakeholder benefits
Tangible benefits

https://www.i-scoop.eu/internet-of-things/

Raspberry PI materials

- Course outline (original)
  - Chapter 1: Introduction
  - Chapter 2: Basic of RaspPI
  - Chapter 3: RaspPI GPIO (2 week)
  - Chapter 4: Python on RaspPI
  - Chapter 5: Output of embedded system
  - Chapter 6: Input of embedded system
  - Chapter 7: RaspPI Networking
  - Chapter 8: Linking with Arduino
  - Chapter 9: Embedded opencv (2 week)
  - Chapter 10: Cloud platform
  - Chapter 11: Final project

https://drive.google.com/file/d/1r3nSUufSA0wO9MLPtAdbwjtJ5fbfT7e9/view?usp=sharing

Online materials

- Slide Share資料 (by 台灣樹莓派)
  - https://www.slideshare.net/raspberrypi-tw/presentations

- Raspberry Pi Camera + Python
- 用Raspberry Pi學GPIO - 自己做遊戲機
- 用Raspberry Pi做手機 - PiPhone
- (COSCUP2014) Raspberry Pi 應用展示與遊戲機演進篇
- (MakerConf2014) Raspberry PI as a Registration System 報到系統
- 用Raspberry Pi學Linux驅動程式
- Raspberry Pi應用與軟硬整合介紹
- 第一次安裝設定Raspberry Pi就上手
- 深入淺出Raspberry Pi GPIO
- 自己動手做無線路由器
- Introduction to Raspberry Pi GPIO
- Introduction to Raspberry Pi 應用篇
- 改造 ReSpeaker 2-MIC HAT
- 小鴨城(Duckietown) 開源無人小車專案介紹
- 樹莓派在物聯網上的應用案例
- 智慧插座(Smart Relay): 從GPIO到繼電器
- 用Raspberry Pi + LoRa實做微型物聯網閘道器
- Raspberry Pi Camera + Python + OpenCV (Day1)
- Raspberry Pi Camera + Python + OpenCV (Day2)
- Raspberry Pi特色相機介紹(熱成像攝影機+微距相機)
- Learn LoRa with Python and Raspberry Pi
- Building a Raspberry Pi LoRa Nano Gateway
- Raspberry Pi IoT無線傳輸技術介紹 - Bluetooth篇
- Raspberry Pi 寵物小車
- Raspberry Pi Zero 網路解決方案
- Introduction to Framebuffer
- Raspberry Pi IoT無線傳輸技術介紹 - ZigBee篇

Outline

- 01/16
  - 10:00 – 12:00: Raspberry PI Introduction
  - 13:30 – 15:00: Raspberry PI GPIO and Python
  - 15:30 – 17:00: I/O of Raspberry PI
- 01/17
  - 10:00 – 12:00: I/O of Raspberry PI
  - 13:30 – 15:00: Communications
  - 15:30 – 17:00: Cloud platform

Update Outline

- **01/16**
  - **10:00 – 12:00**
    - **Raspberry PI Introduction**
    - **Raspberry PI GPIO and Python**
    - **I/O of Raspberry PI – part 1**
  - **13:30 – 15:00, 15:30 – 17:00: 動手操作, Q&A**

- 01/17
  - 10:00 – 12:00
    - I/O of Raspberry PI – part 2
    - Communications
    - Cloud platform
  - 13:30 – 15:00, 15:30 – 17:00: 動手操作, Q&A

補充資料
https://docs.google.com/document/d/16KaALl_yZLpn_HXPV3ZNX7PG8Myg9443ytRqXIgxNEk

1. Raspberry PI Introduction

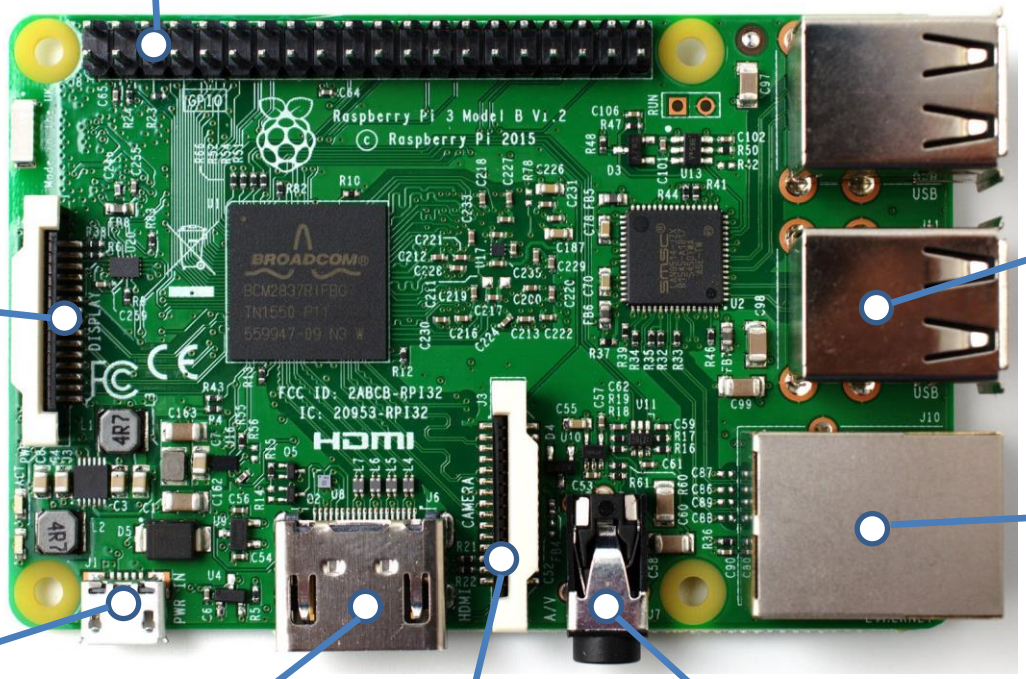
National Chiao Tung University

Introduction

- 信用卡大小般的電腦

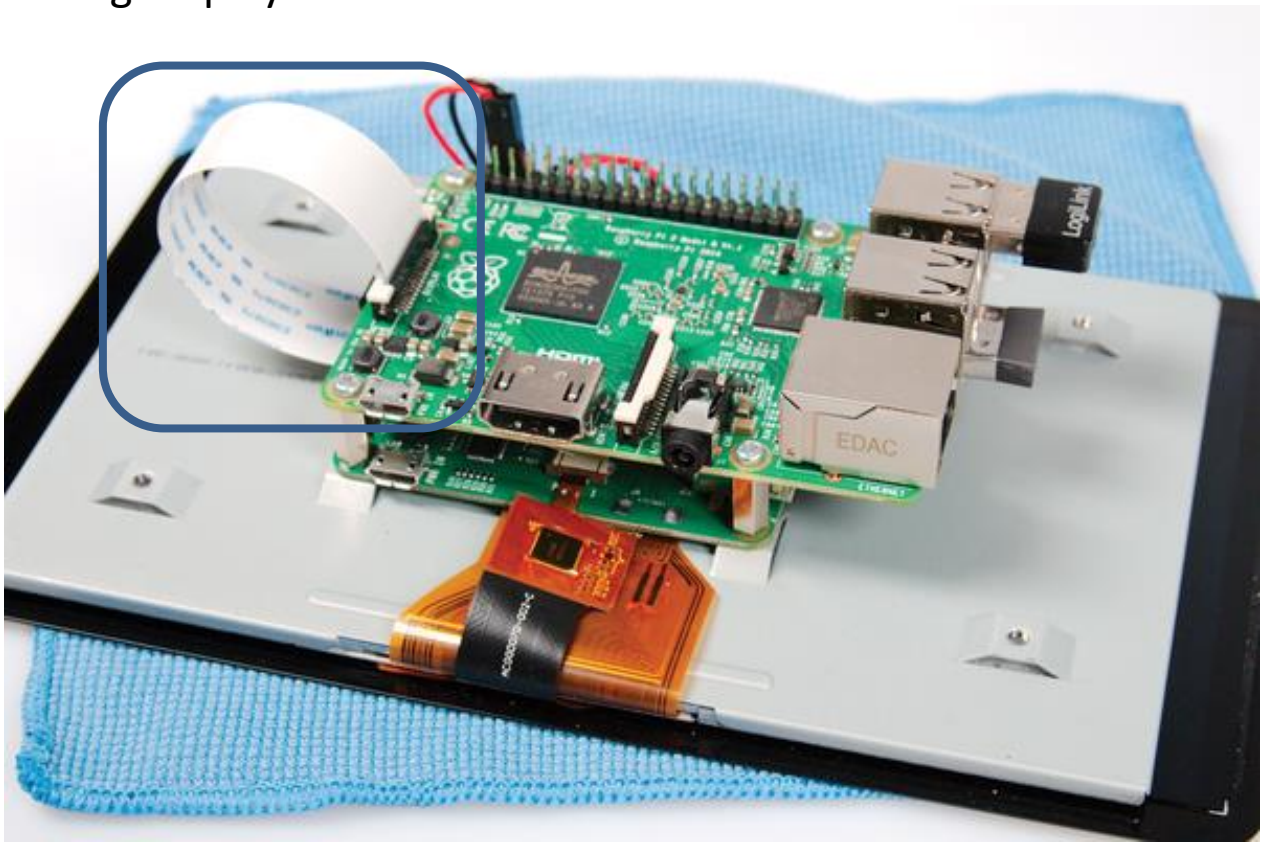
http://www.flickr.com/photos/fotero/7697063016/

Raspberry PI 硬體週邊



Raspberry PI 硬體週邊

Using Display interface



Raspberry PI 硬體週邊

Combined 3.5mm audio jack
and composite video

Camera interface (CSI)

Raspberry PI 3 Model B

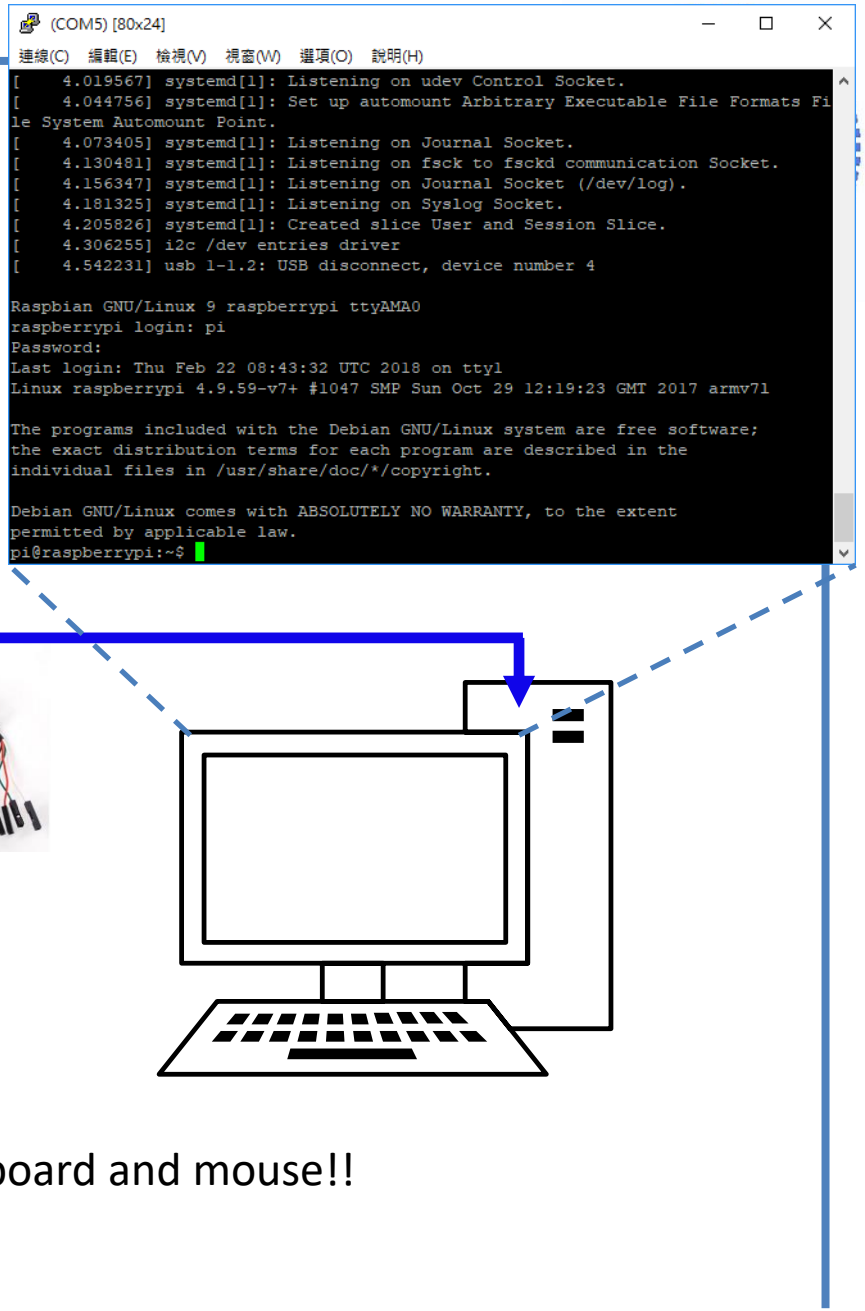
- Raspberry PI 3 model B:
  - 主要有以下功能:
    - 1.) SD 卡: 當成內建的硬碟使用，一般來說最少需要有4GB 的容量，建議用比較穩定的牌子，這樣可以確保讀取資料正常
    - 2.) HDMI 輸出顯示
    - 3.) USB 的輸入端口: 如滑鼠與鍵盤
    - 4.) RJ45網路線端口
    - 5.) Micro USB 電源端口: 僅供電, 無資料傳輸用途
    - 6.) 內建802.11n Wi-Fi 與 藍牙4.1

Raspberry PI 3 Model B

- **Raspberry Pi 3 Model B** 的硬體資訊
  - BCM 2837 SoC (1.2GHz ARM Cortex-A53 64bits)
  - Build-in 802.11n Wi-Fi and Bluetooth 4.1
  - 1GB LPDDR2
  - 4 USB ports
  - 40 GPIO pins
  - Full HDMI port
  - 10M/100M Ethernet port
  - Combined 3.5mm audio jack and composite video
  - Camera interface (CSI)
  - Display interface (DSI)
  - Micro SD card slot
  - VideoCore IV 3D graphics core

Raspberry PI without HDMI?

We will use USB-to-Serial cable to use PI.

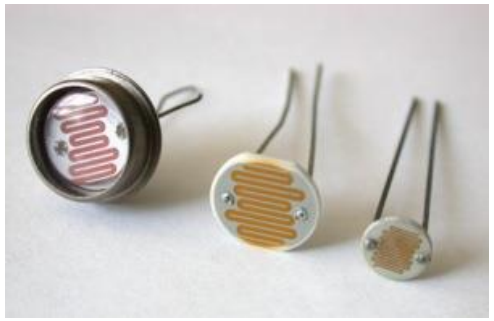


We do not use extra HDMI, keyboard and mouse!!

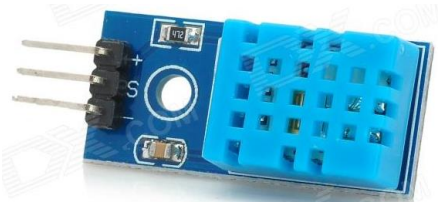
Raspberry Pi 可以用在?

- **IoT應用**
  - 連接特定感測器，讀取數值進行分析
- **桌機 (Raspbian)**
  - FTP, Web, NAS, AP, 自動化控制…等
- **多媒體影音作業系統**
  - OSMC (Open-Source Media Center)
  - OpenELEC (Open Embedded Linux Entertainment Center)
- **遊戲機 (RetroPie、PiPlay)**
- **網站滲透測試 (Kali Linux)**
- **Android系統 (RaspAnd)**
- **超級電腦**
  - 將一堆PI組裝在一起

Sensors



Photocell (light sensor)



Temperature/Humidity

PIR

Ultrasonic

Flame Sensor (火焰感測)

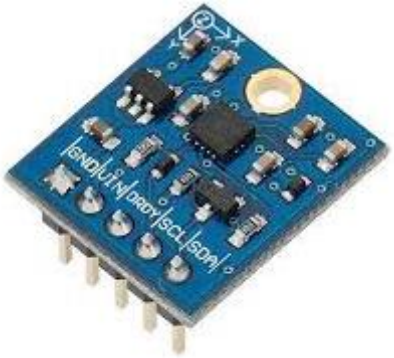
Water Sensor

Sensors

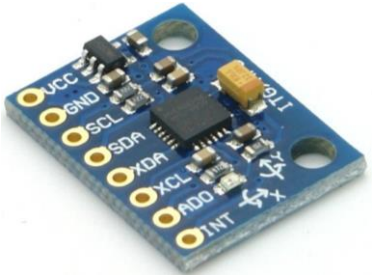
Smoke Detector

Pressure

CO2 Sensor



Compass (電子羅盤)

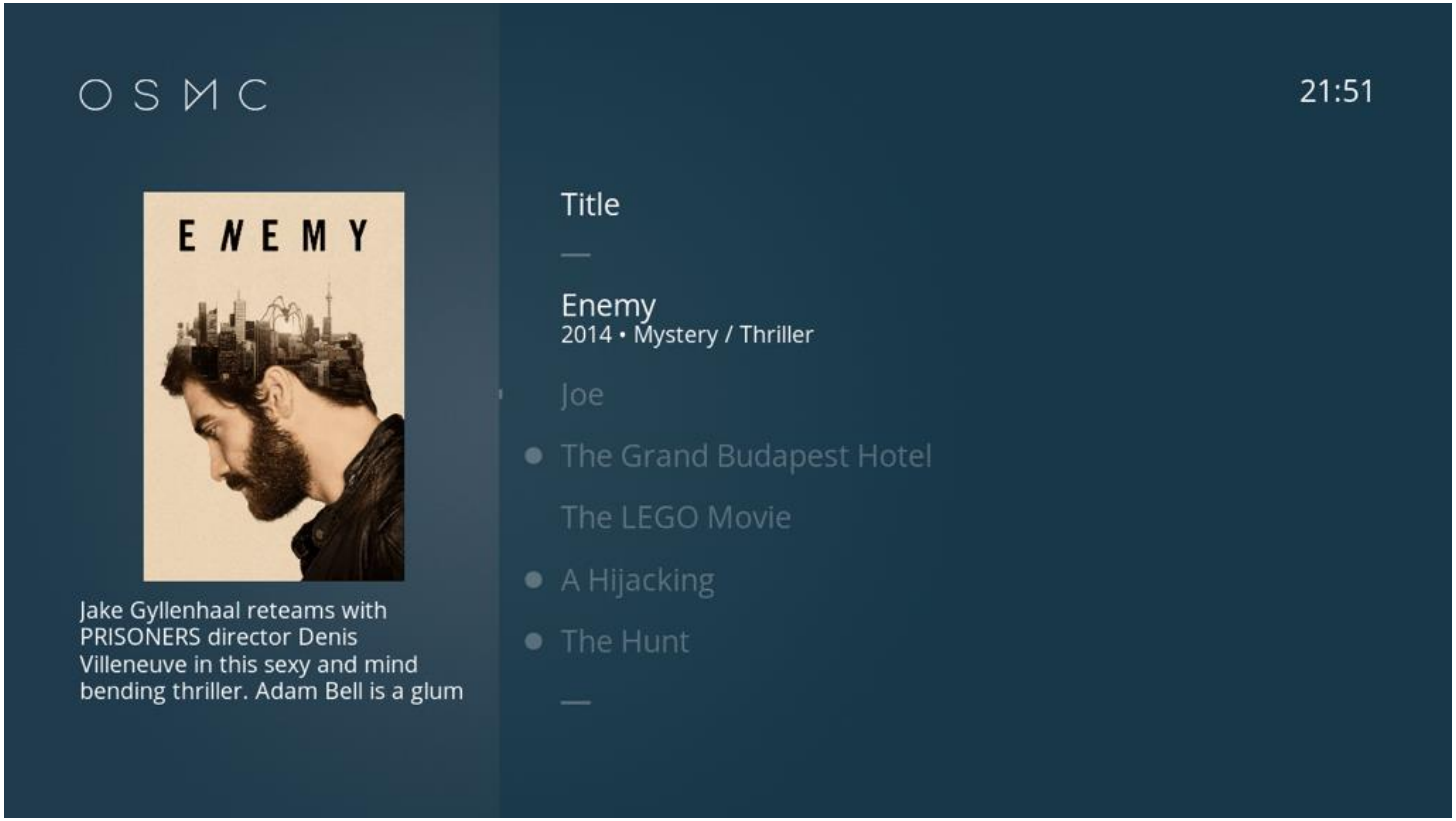


Tri-Axis Accelerometer
(三軸加速度計)

Heartbeat Sensor

Raspberry Pi 可以用在?

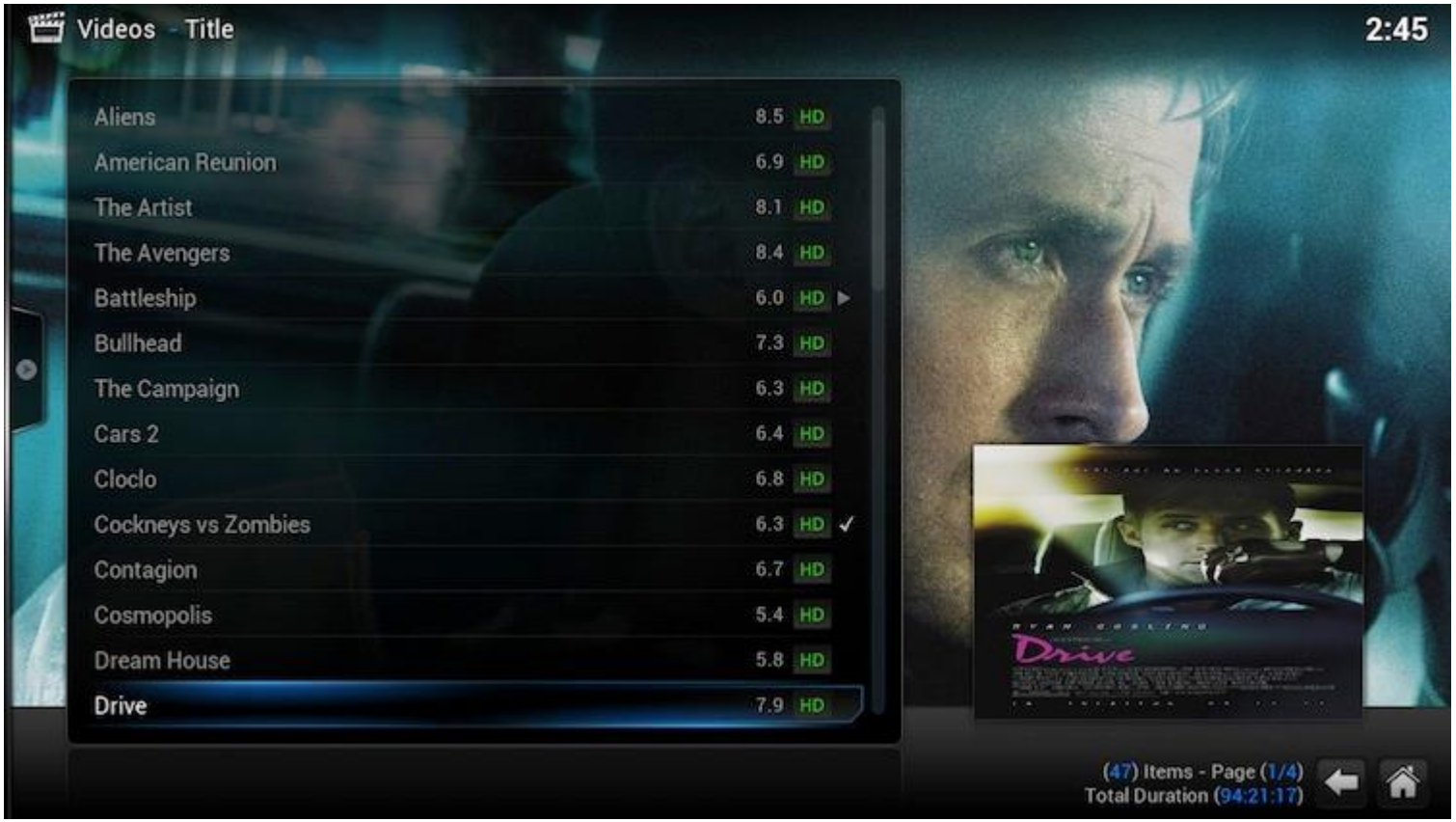
- OSMC



https://osmc.tv/download/

Raspberry Pi 可以用在?

- OpenELEC



http://openelec.tv/get-openelec

Raspberry Pi 可以用在?

- RetroPie

http://makezine.com/projects/build-retro-gaming-console-raspberry-pi/

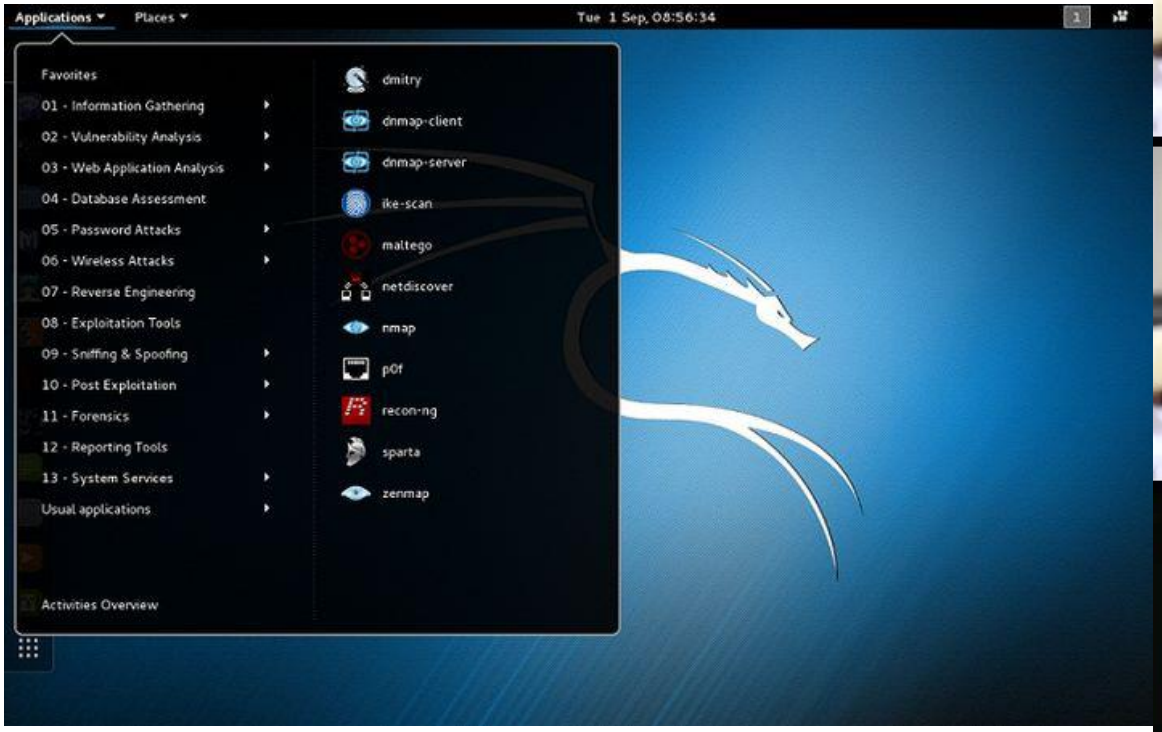
Raspberry Pi 可以用在?

- PiPlay

http://piplay.org/

Raspberry Pi 可以用在?

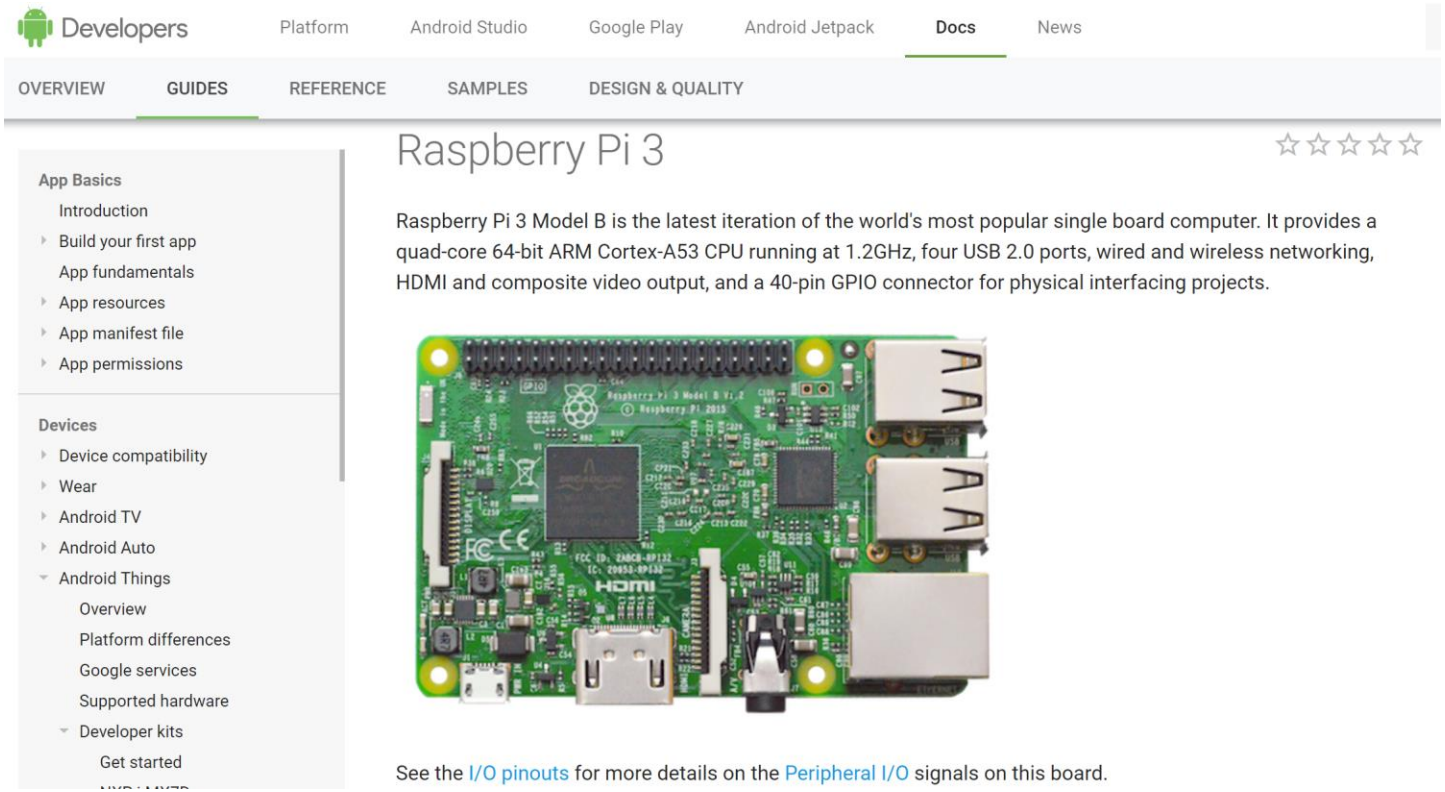
- Kali Linux



https://www.kali.org/

Raspberry Pi 可以用在?

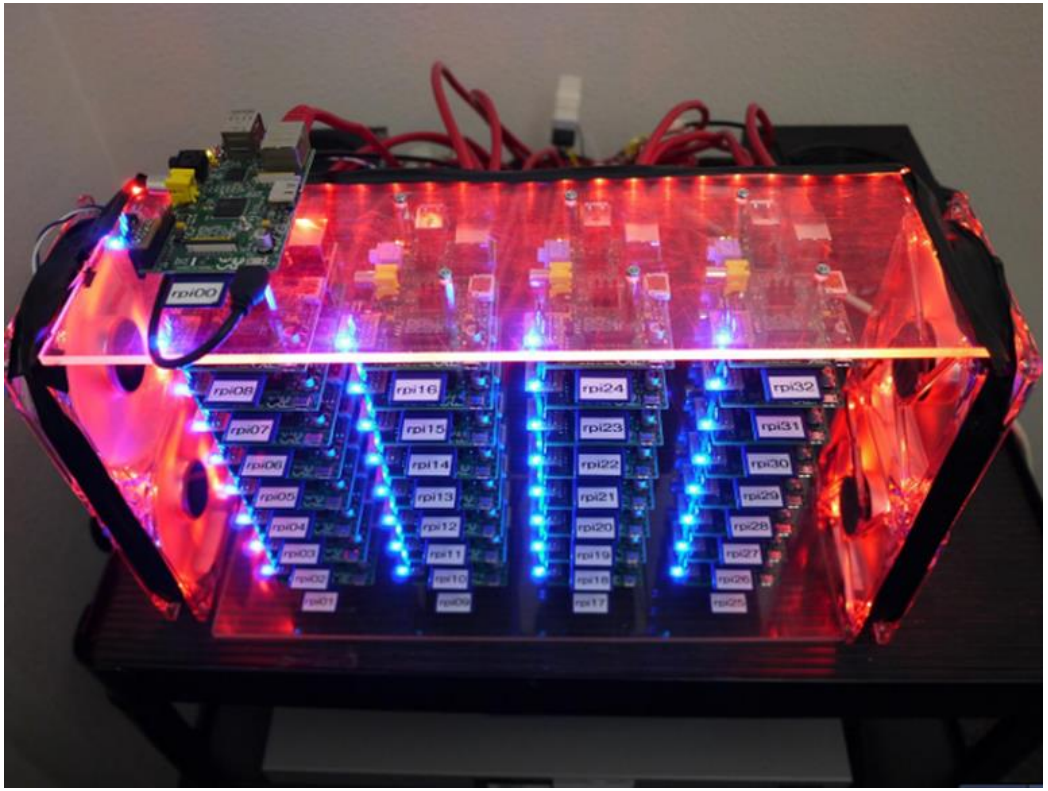
- Android Things



https://developer.android.com/things/hardware/raspberrypi

Raspberry Pi 可以用在?

- 超級電腦

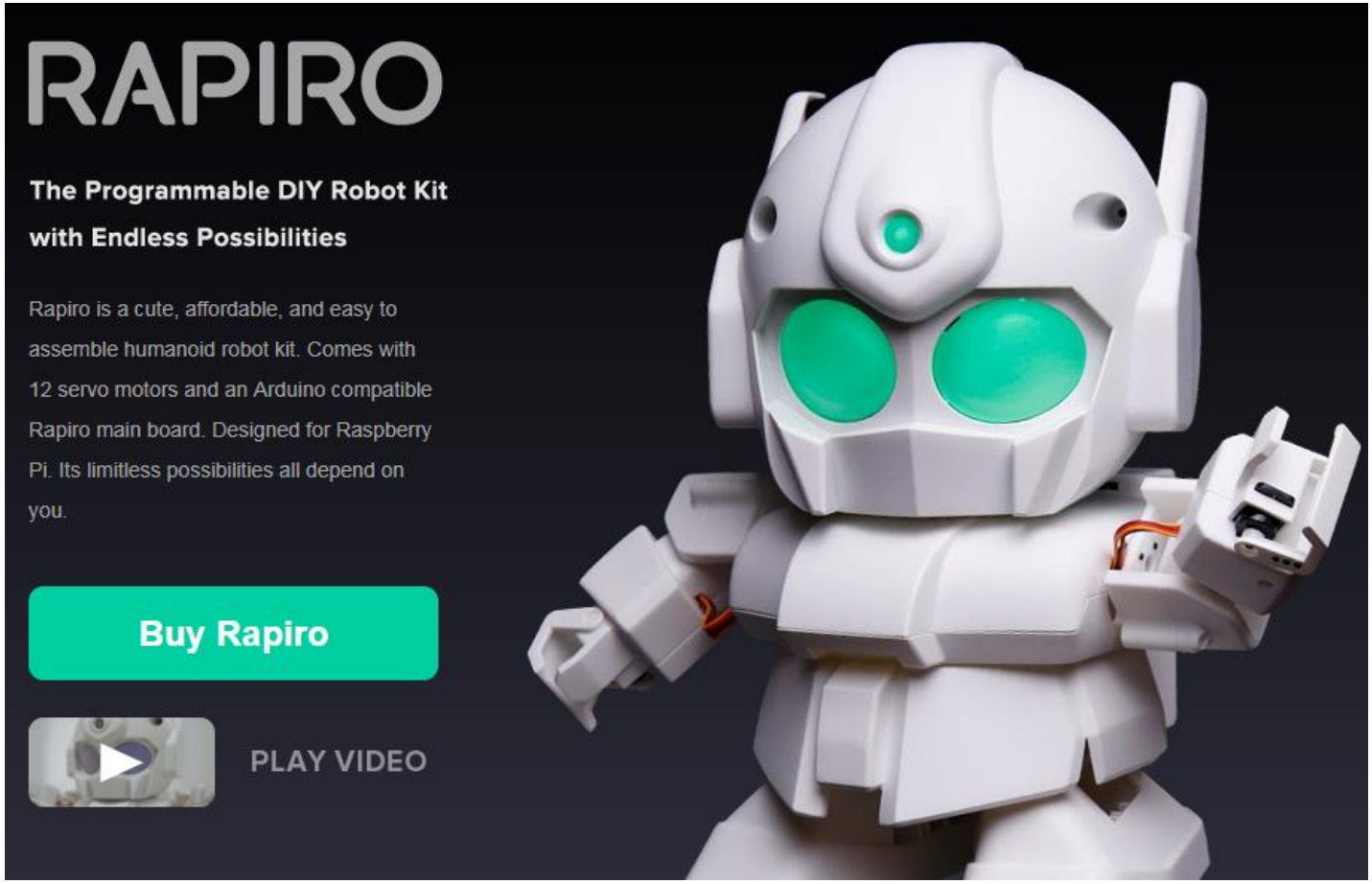


http://www.zdnet.com/article/build-your-own-supercomputer-out-of-raspberry-pi-boards/

Raspberry Pi 可以用在?

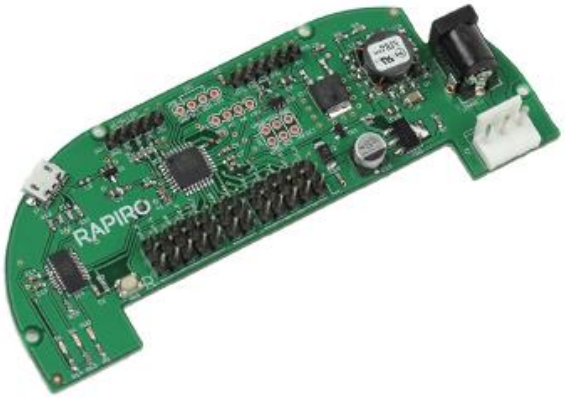
- 空間邏輯測驗(?)

RAPIRO


RAPIRO
The Programmable DIY Robot Kit
with Endless Possibilities
Rapiro is a cute, affordable, and easy to assemble humanoid robot kit. Comes with 12 servo motors and an Arduino compatible Rapiro main board. Designed for Raspberry Pi. Its limitless possibilities all depend on you.
Buy Rapiro
PLAY VIDEO

RAPIRO

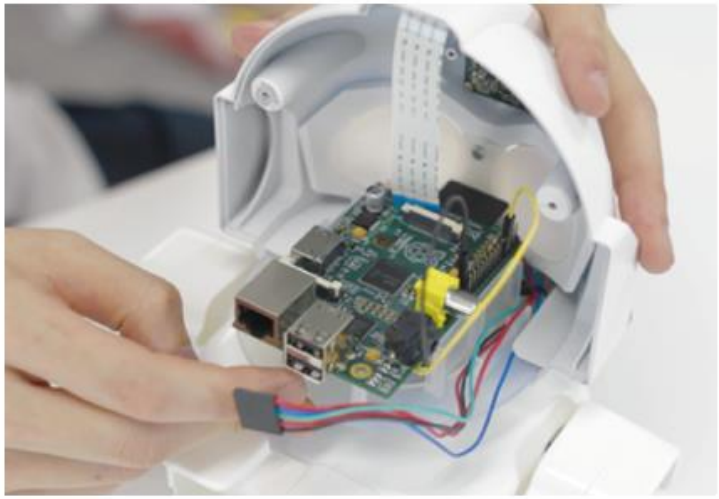
Arduino Compatible Controller



The Rapiro main board works like an Arduino board, and it can be reprogrammed using Arduino IDE. Rapiro is not only for those who have programming skills but also for beginners.

Arduino is a combination of an 8-bit AVR based microcontroller board and C++ like development environment (Arduino IDE).

You can install Raspberry Pi



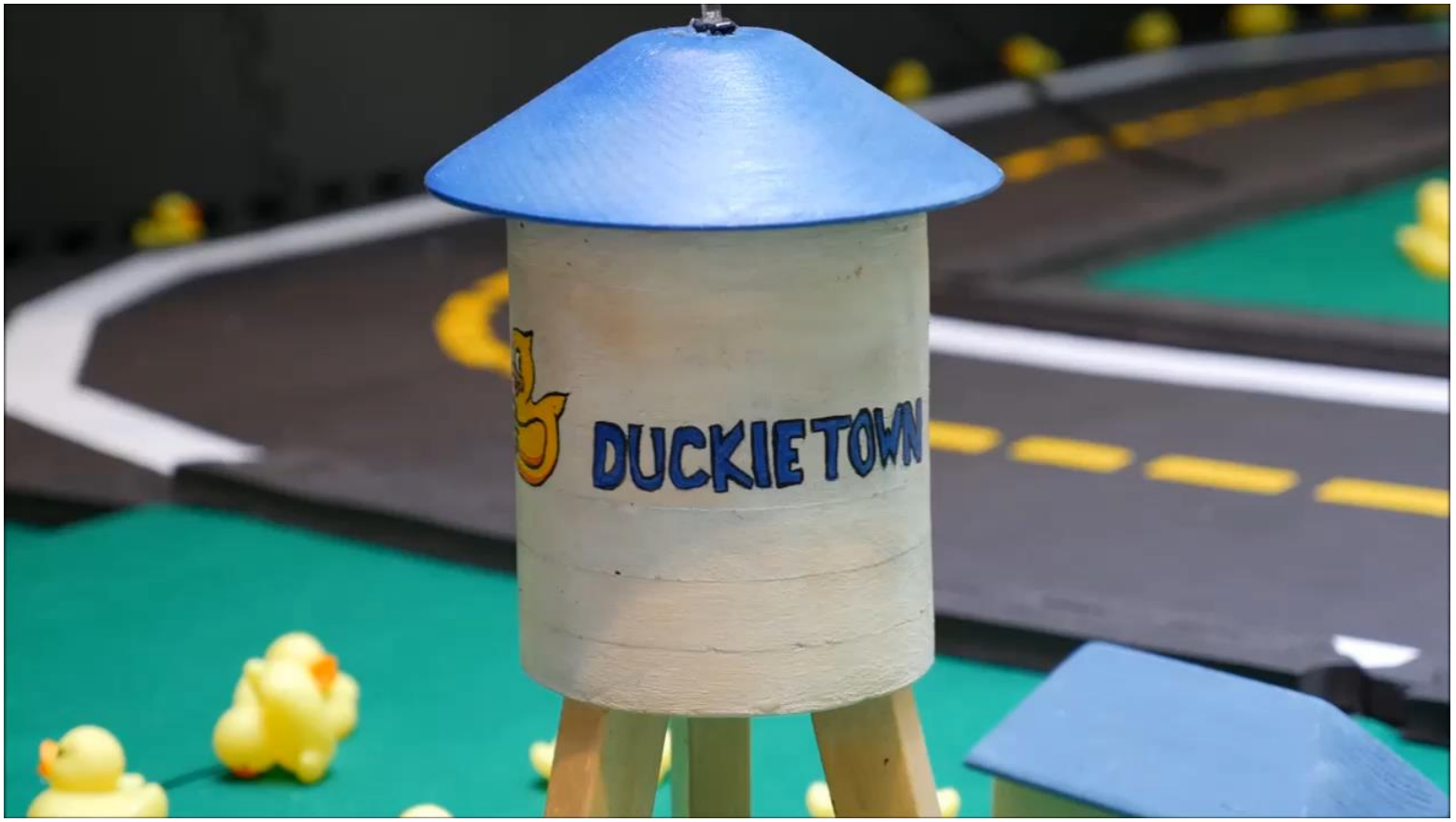
If you want more functions, you could do it by installing a Raspberry Pi (sold seprately) inside Rapiro. Raspberry Pi can be programmed with a wide variety of programming languages for Linux. Depending on your programming, your Rapiro could give you message notifications with Wi-Fi enabled or could protect your home as a security robot using a camera module!

RAPIRO

https://www.youtube.com/watch?v=AQynsVZx5Pk

Duckietown (self-driving robot)

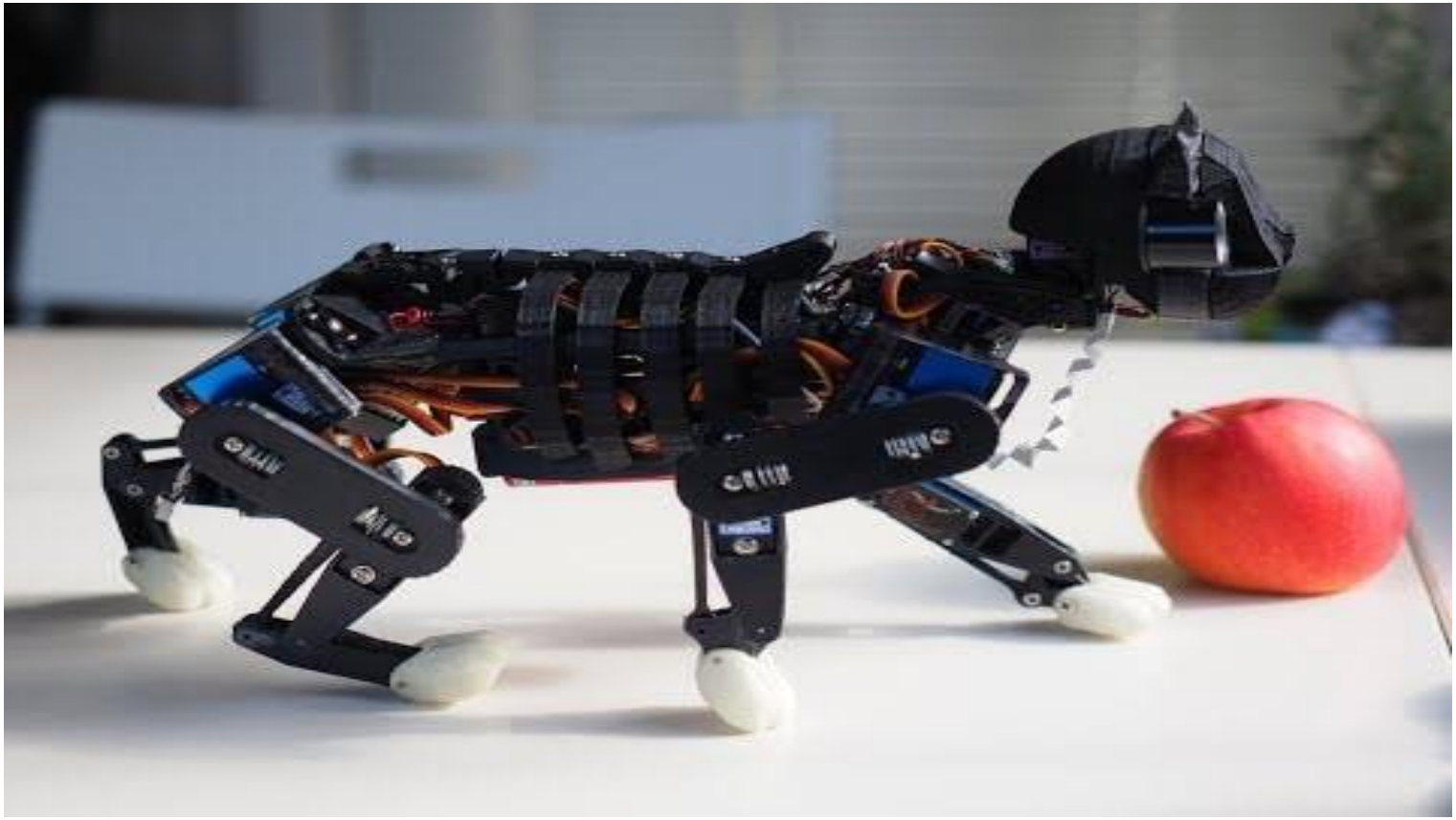
- An Open-Source MIT Class & Computer-Vision Self-Driving Robot for Raspberry Pi



https://blog.adafruit.com/2016/05/25/duckietown-is-an-open-source-mit-class-computer-vision-self-driving-robot-for-raspberrypi/

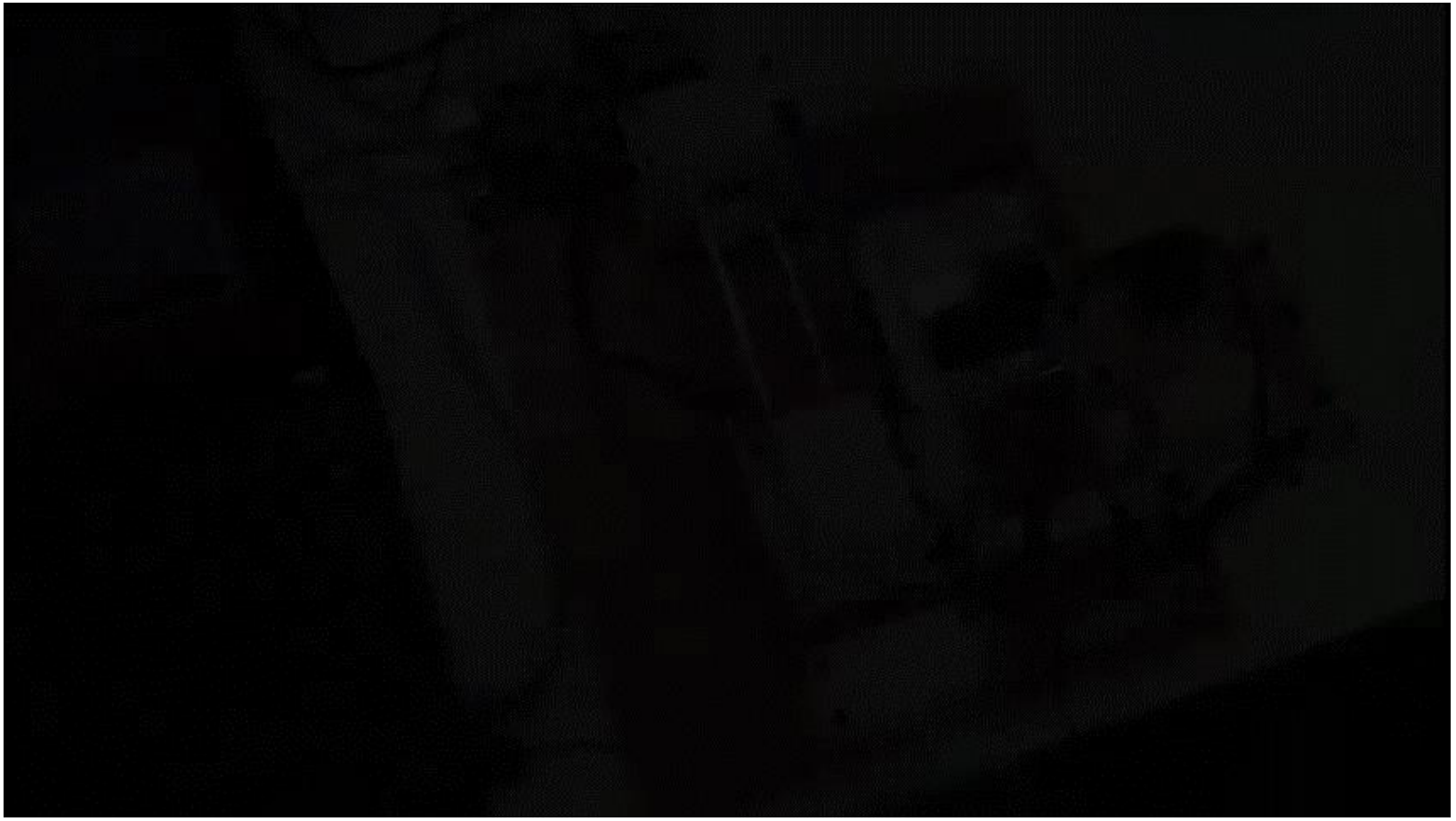
PI cat

- https://www.youtube.com/watch?v=iGEdF5QsDB8



https://www.raspberrypi.org/blog/petoi-a-pi-powered-kitty-cat/

Puzzle and Dragon robot



http://www.nicovideo.jp/watch/sm28160788

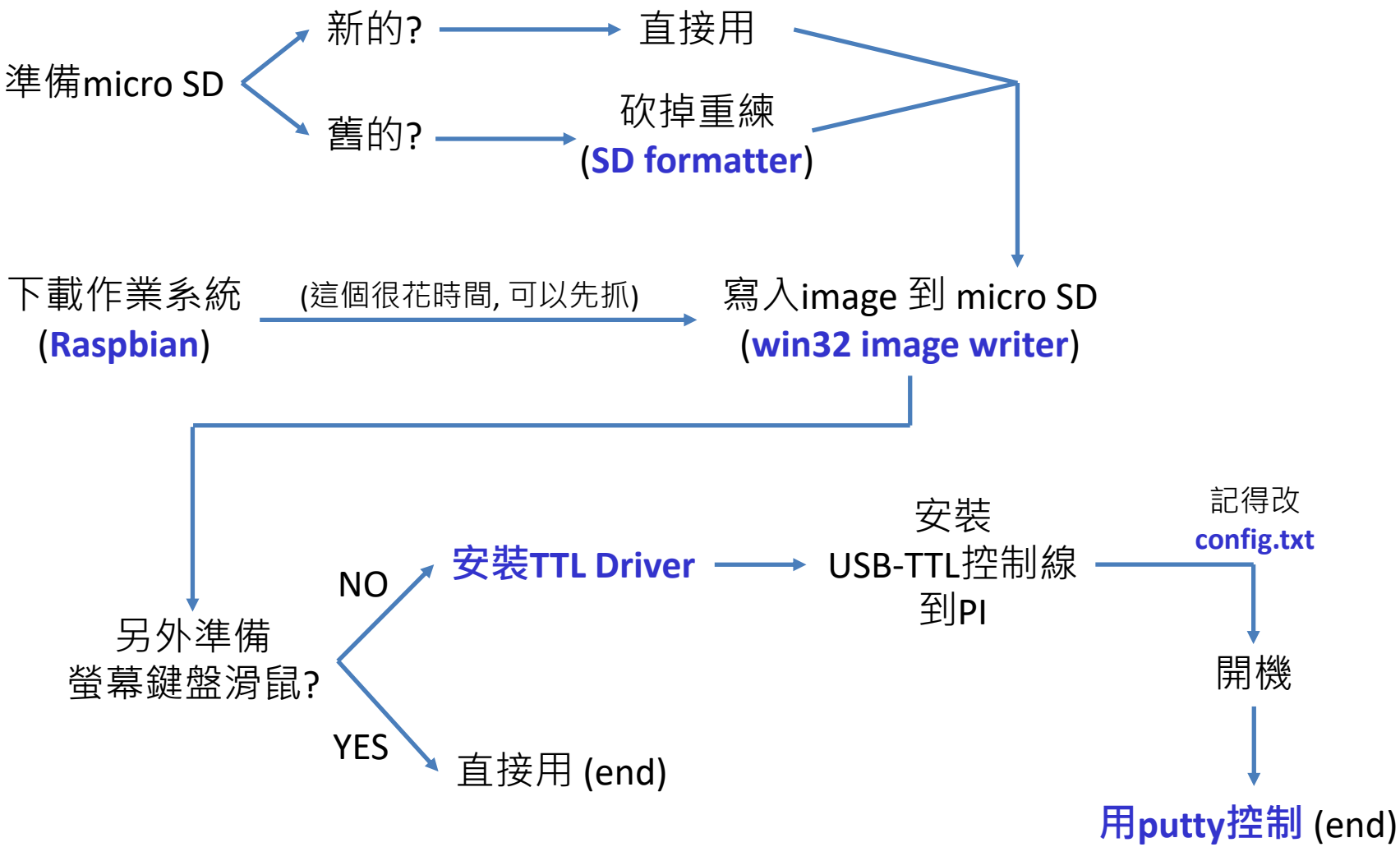
1. Raspberry PI Introduction Installation

National Chiao Tung University

Installation

- **1. 安裝OS (Raspbian)**
- **2. PI的環境設定**
  - 擴充SD卡空間
  - 用command line設定Wi-Fi
  - 使用apt-get安裝程式
- **3. 設定遠端桌面連線**
  - xrdp

準備流程


準備micro SD
新的?
直接用
舊的?
砍掉重練
(SD formatter)
下載作業系統
(Raspbian)
(這個很花時間, 可以先抓)
寫入image 到 micro SD
(win32 image writer)
另外準備
螢幕鍵盤滑鼠?
NO
安裝TTL Driver
安裝
USB-TTL控制線
到PI
記得改
config.txt
開機
YES
直接用 (end)
用putty控制 (end)

準備工具

- Raspbian image
  - https://downloads.raspberrypi.org/raspbian_latest
  - Ubuntu-tw分流連結
- SD Formatter
  - https://www.sdcard.org/cht/downloads/formatter_4/
- Win32 image writer
  - https://sourceforge.net/projects/win32diskimager/
- Notepad ++
  - https://notepad-plus-plus.org/download/v7.3.2.html
- USB TTL driver
  - http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41
- Putty
  - https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

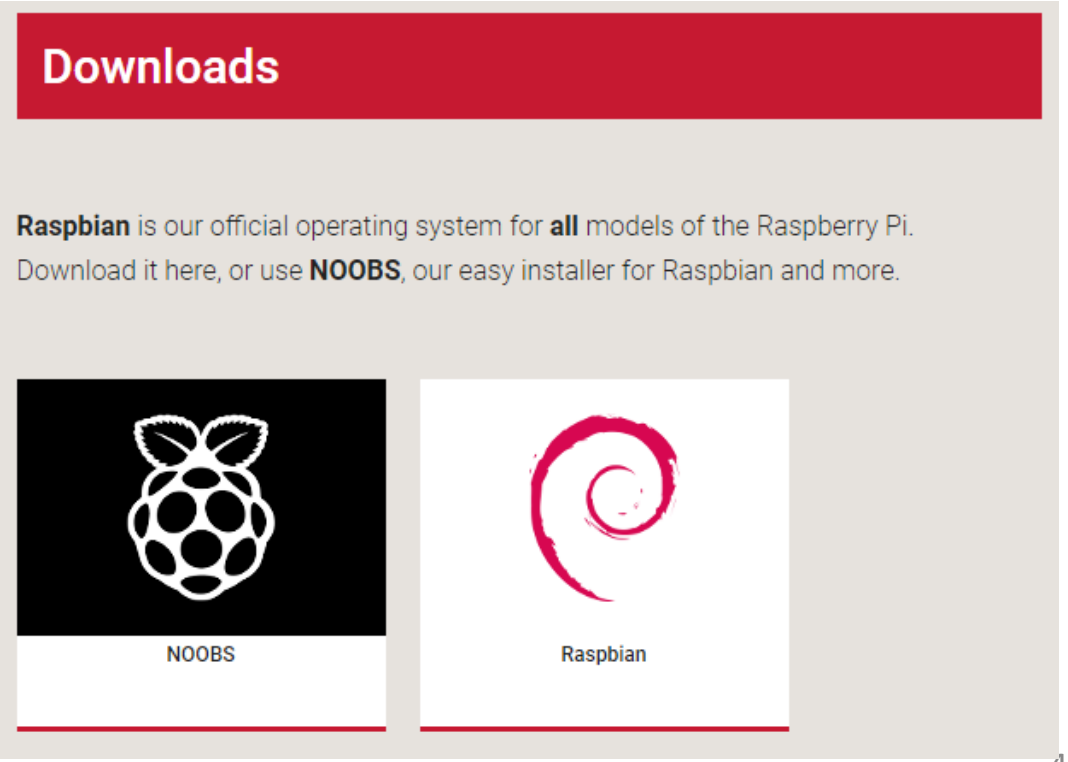
安裝Raspbian

- 步驟1：下載映像檔
  - Raspbian
- 步驟2：將映像檔燒錄至SD卡
  - SD formatter, win32 image writer
- 步驟3：調整開機設定檔
  - config.txt
- 步驟4：電腦端準備 USB 轉 TTL 序列傳輸線
  - 安裝TTL Driver
- 步驟5：將SD卡插到Raspberry PI並開機
  - 用putty控制 (by USB-TTL)

步驟1：下載映像檔

- 官方下載網頁
  - http://www.raspberrypi.org/downloads
  - Ubuntu-tw分流連結 (速度較快)

- 選擇映像檔(image)
  - NOOBS
  - **Raspbian(推薦)**
  - Pidora
  - RISC OS
  - RaspBMC
  - Arch
  - OpenELEC



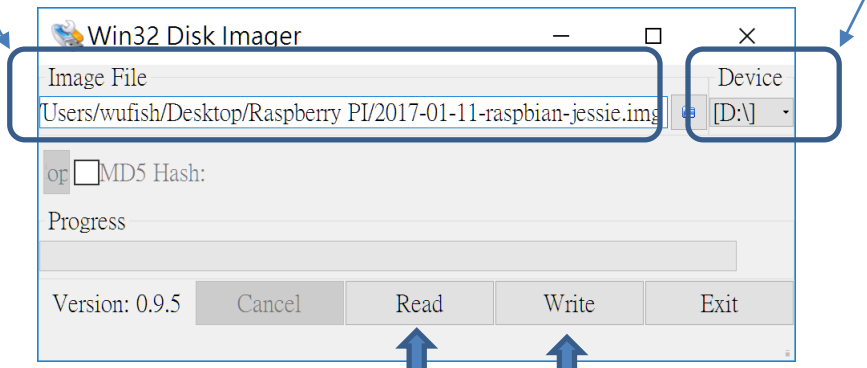
步驟2：將映像檔燒錄至SD卡

- **下載燒錄軟體**
  - Win32 Disk Imager
    - http://sourceforge.net/projects/win32diskimager/

- **燒錄映像檔**
  - 開啟Win32DiskImager
  - 選取映像檔及欲燒入SD卡路徑並開始燒錄

選擇image位置

選擇SD卡位置



Read: 將SD卡儲存至image路徑, 可備份整個系統

Write: 將image寫入至SD卡

步驟3：調整開機設定檔

- 拿到 PI 3, 想使用序列埠連線, 會發現出現亂碼, 該怎麼辦？(使用螢幕鍵盤滑鼠就可以忽略這問題)

因為原本 Pi 3 內建的硬體 UART 被 BCM2837 SoC 拿去給 Bluetooth 晶片組使用，而原本的 UART 輸出腳位(GPIOs 14 & 15)改成用 mini-uart port。意思是原本硬體 UART 有獨立的 clock divisor，因此 baud rate 可以維持在 115200，可是 mini-uart 使用系統核心時脈，實際只能跑到 72000 左右的 baud rate，因此當使用 115200 的 baud rate 連線就會出現亂碼

步驟3：調整開機設定檔

- 修改/boot/config.txt,新增三行
  - dtoverlay=pi3-miniuart-bt
  - core_freq=250
  - enable_uart=1

```
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
dtoverlay=pi3-miniuart-bt
core_freq=250
enable_uart=1
```

新增

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

```
dwc_otg.lpm_enable=0 console=serial0,115200
console=tty1 root=PARTUUID=6b3a87b5-02 rootfstype=ext4
elevator=deadline fsck.repair=yes rootwait quiet
init=/usr/lib/raspi-config/init_resize.sh splash
plymouth.ignore-serial-consoles
```

SD卡插入windows電腦後, 會出現一個磁碟機. 修改磁碟機裡面的config.txt即可!

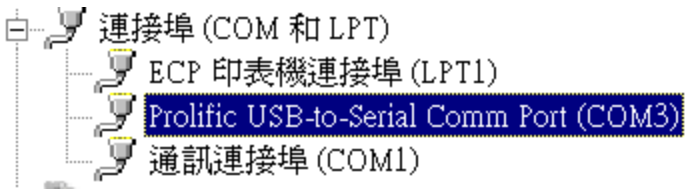
步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

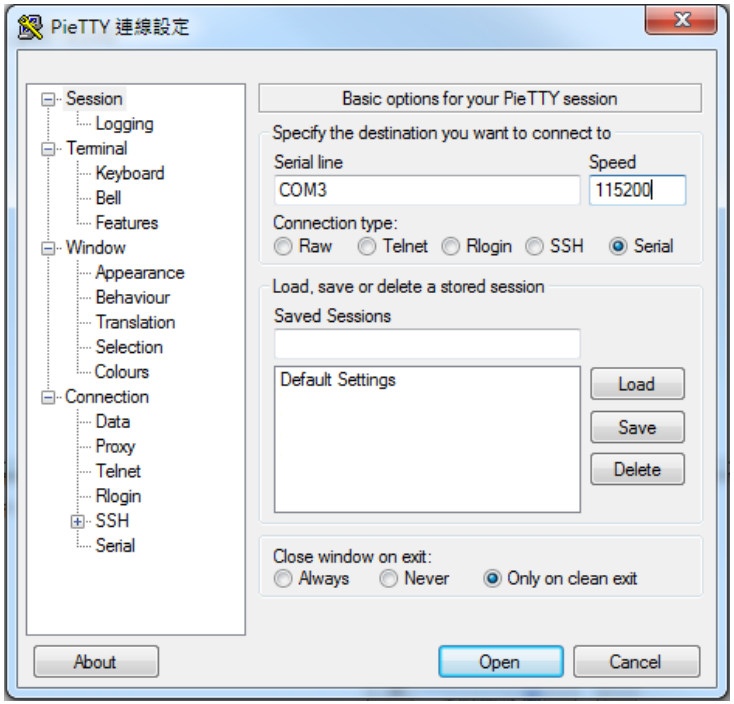
- **從序列埠登入到 Raspberry Pi**
  - 透過 USB 轉 TTL 序列傳輸線，就可以在不需要螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi

- 有兩種晶片組
  - PL2303HX: 支援Windows XP/Vista/7, Linux
  - PL2303HXD:支援Windows 8/10, MAC
    - Driver: http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

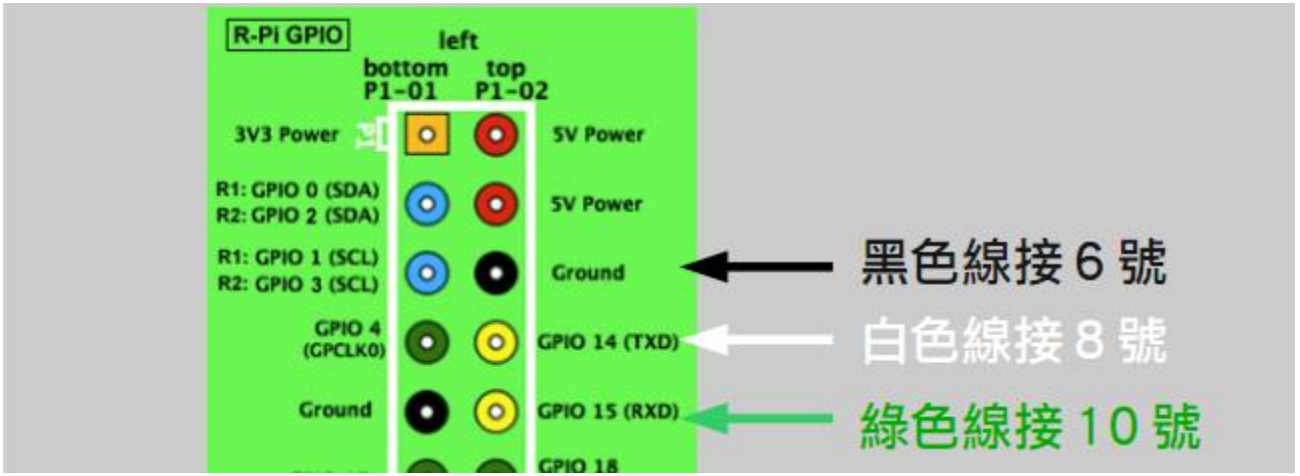
- 安裝完驅動, 在裝置管理員可發現多一個連接阜 (下圖為 COM3)
  - 我的電腦 -> 右鍵-> 內容-> 裝置管理員
- 使用putty連線 (設定Serial port 與 Speed)





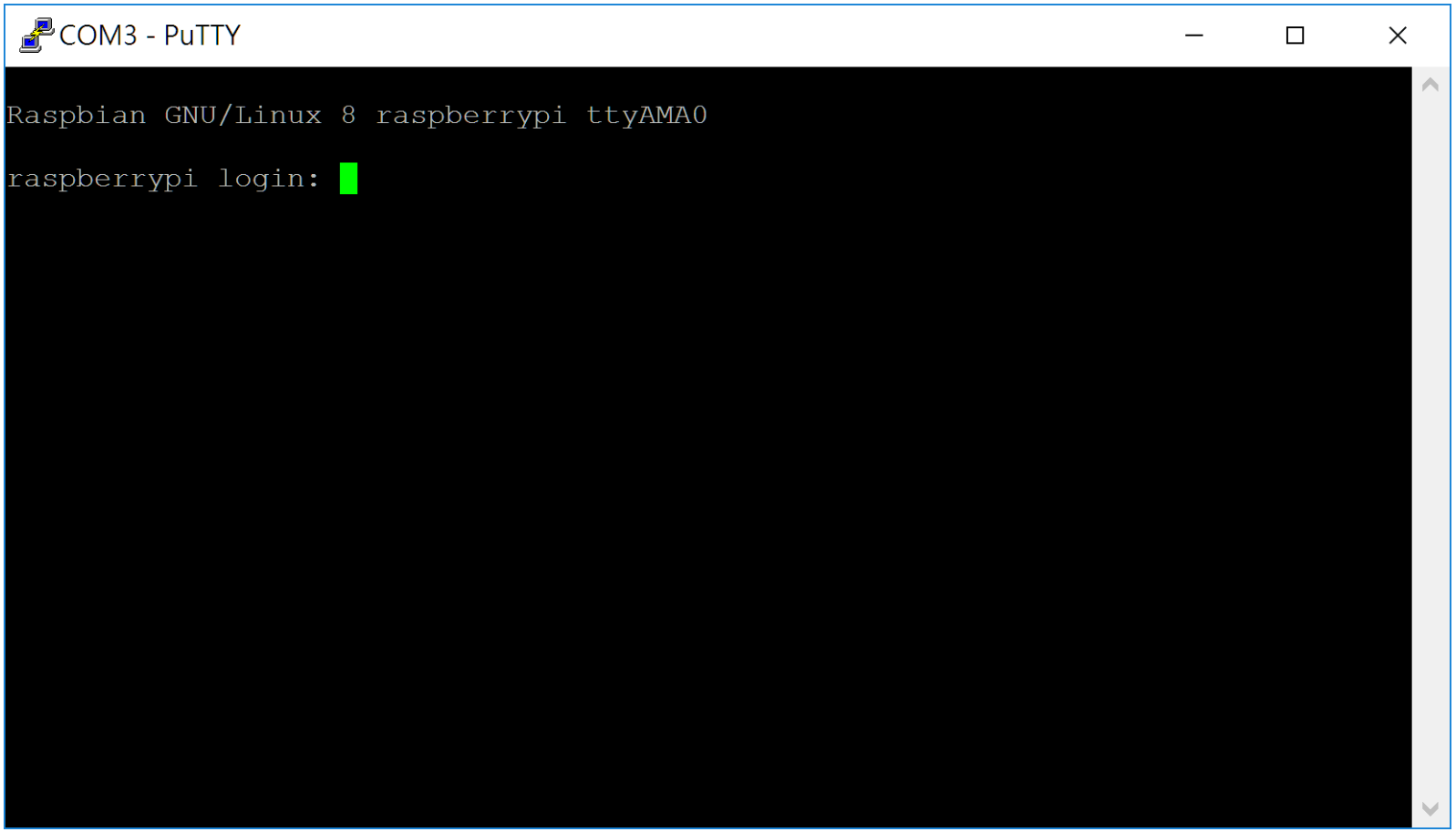
步驟5：將SD卡插到Raspberry PI並開機

- **從序列埠登入到 Raspberry Pi**
  - 透過 USB 轉 TTL 序列傳輸線，就可以在不需要螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi
- **預設登入帳密**
  - ID: pi
  - PW: raspberry



https://www.raspberrypi.com.tw/tag/usb-to-ttl/

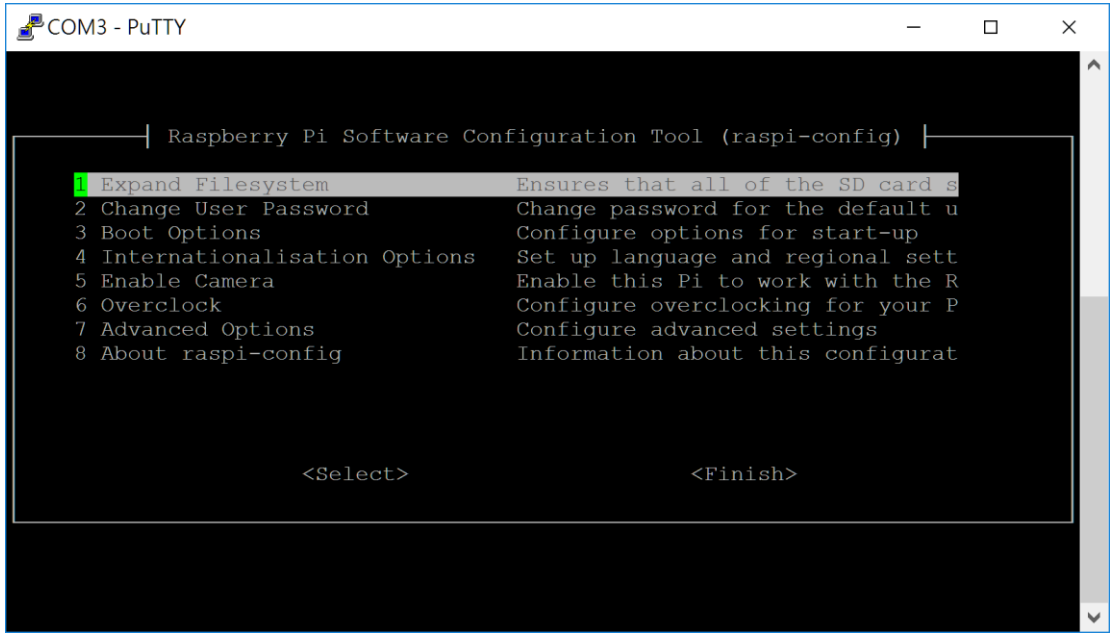
步驟5：將SD卡插到Raspberry PI並開機



還是沒有畫面? -> 按一下Enter鍵 or 電腦重開機試試

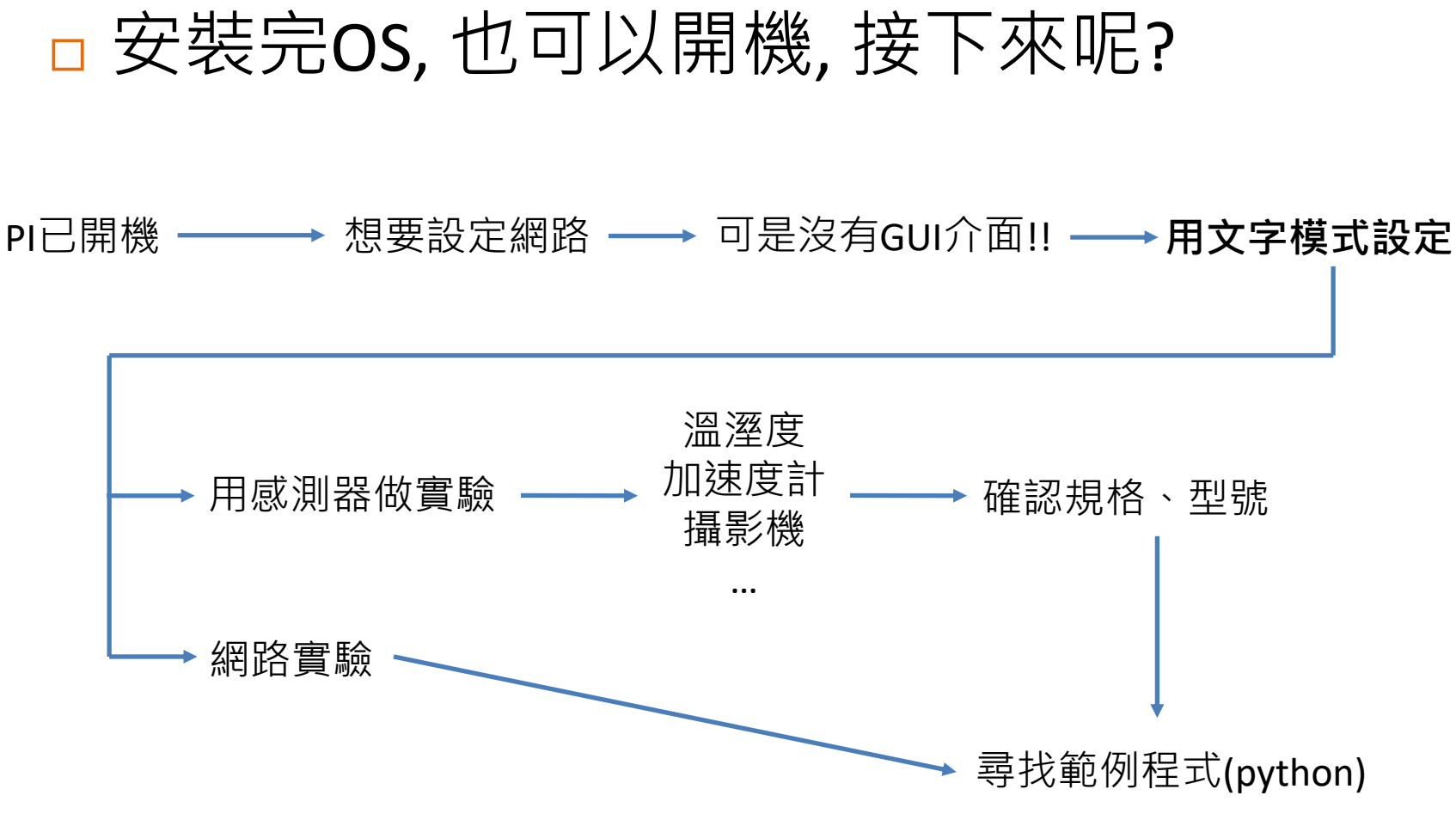
2. PI的環境設定

- 擴充SD卡空間
  - 預設只有使用約2G的空間 -> 可是SD卡有16G -> 浪費了10G的空間
  - 在terminal下輸入: **sudo raspi-config ->** 選擇「 **Expand Filesystem** 」
  - Reboot後系統所能使用的空間就為SD卡所擁有的空間



準備流程

- 安裝完OS, 也可以開機, 接下來呢?



各式各樣的應用

- Slide Share資料 (by 台灣樹莓派)
  - https://www.slideshare.net/raspberrypi-tw/presentations

- Raspberry Pi Camera + Python
- 用Raspberry Pi學GPIO - 自己做遊戲機
- 用Raspberry Pi做手機 - PiPhone
- (COSCUP2014) Raspberry Pi 應用展示與遊戲機演進篇
- (MakerConf2014) Raspberry PI as a Registration System 報到系統
- 用Raspberry Pi學Linux驅動程式
- Raspberry Pi應用與軟硬整合介紹
- 第一次安裝設定Raspberry Pi就上手
- 深入淺出Raspberry Pi GPIO
- 自己動手做無線路由器
- Introduction to Raspberry Pi GPIO
- Introduction to Raspberry Pi 應用篇
- 改造 ReSpeaker 2-MIC HAT
- 小鴨城(Duckietown) 開源無人小車專案介紹
- 樹莓派在物聯網上的應用案例
- 智慧插座(Smart Relay): 從GPIO到繼電器
- 用Raspberry Pi + LoRa實做微型物聯網閘道器
- Raspberry Pi Camera + Python + OpenCV (Day1)
- Raspberry Pi Camera + Python + OpenCV (Day2)
- Raspberry Pi特色相機介紹(熱成像攝影機+微距相機)
- Learn LoRa with Python and Raspberry Pi
- Building a Raspberry Pi LoRa Nano Gateway
- Raspberry Pi IoT無線傳輸技術介紹 - Bluetooth篇
- Raspberry Pi 寵物小車
- Raspberry Pi Zero 網路解決方案
- Introduction to Framebuffer
- Raspberry Pi IoT無線傳輸技術介紹 - ZigBee篇

2. PI的環境設定

- 設定Wi-Fi
  - 使用螢幕鍵盤滑鼠, 可以直接點選Wi-Fi
    -> 在純文字介面要怎麼設定Wi-Fi?
- 用文字編輯器修改設定檔
  1. sudo nano /etc/wpa_supplicant/wpa_supplicant.conf (編輯設定檔)
  2. 填寫SSID與密碼
  3. 重開機
     - sudo reboot
  4. 使用 ifconfig 與 iwconfig 檢查連線狀態

```
network={
        ssid= "your_ap"
        key_mgmt=NONE
} // open system
```

```
network={
        ssid= "your_ap"
        psk= "your_passwd"
} // with password
```

2. PI的環境設定

- 修改設定檔
  - sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

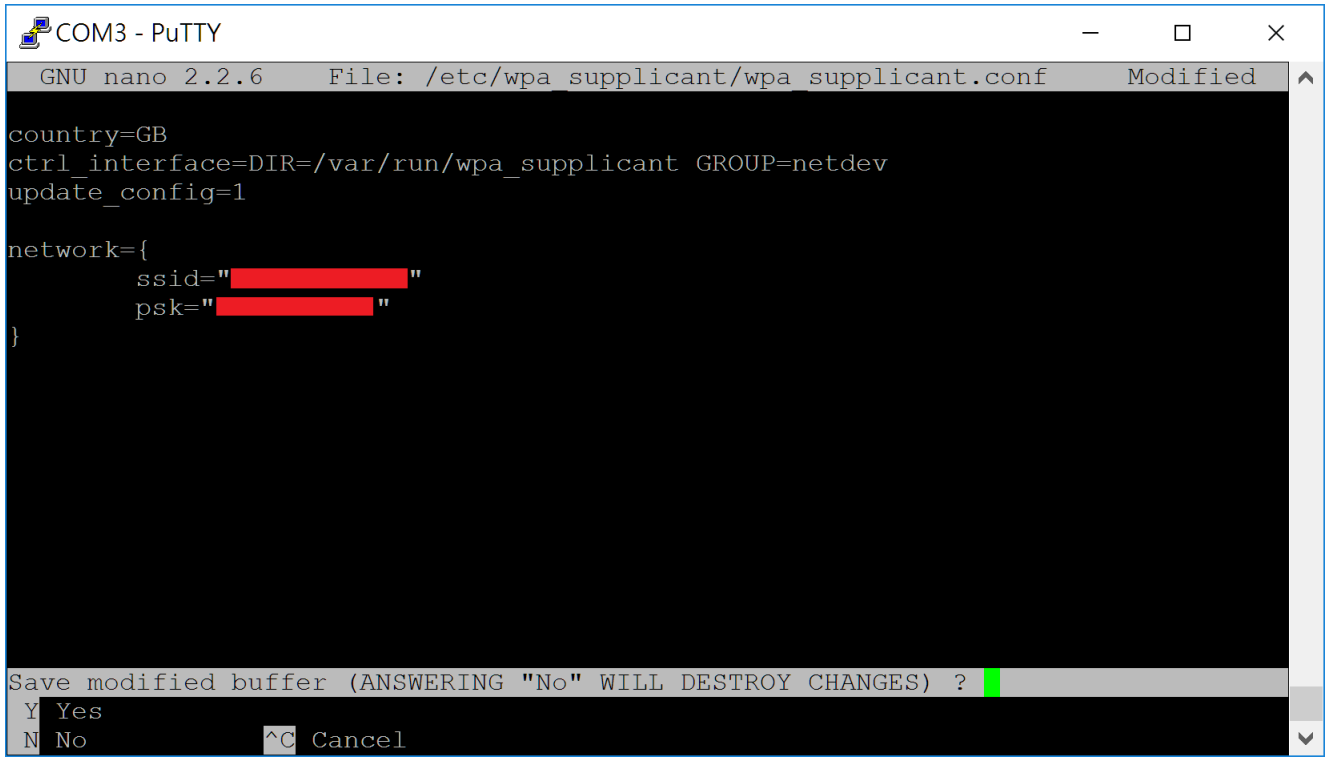
COM3 - PuTTY

```
  GNU nano 2.2.6    File: /etc/wpa_supplicant/wpa_supplicant.conf

country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
        ssid="[redacted]"
        psk="[redacted]"
}

                                  [ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

```
network={
        ssid="SSID"
        psk="your_pw"
}

network={
        ssid="SSID_open"
        key_mgmt=NONE
}
```

2. PI的環境設定

- 文字編輯器 nano
  - 編輯結束按 **ctrl + x** 離開
    - 若有變動, 會問你是否存檔, 輸入 **Y** 即可

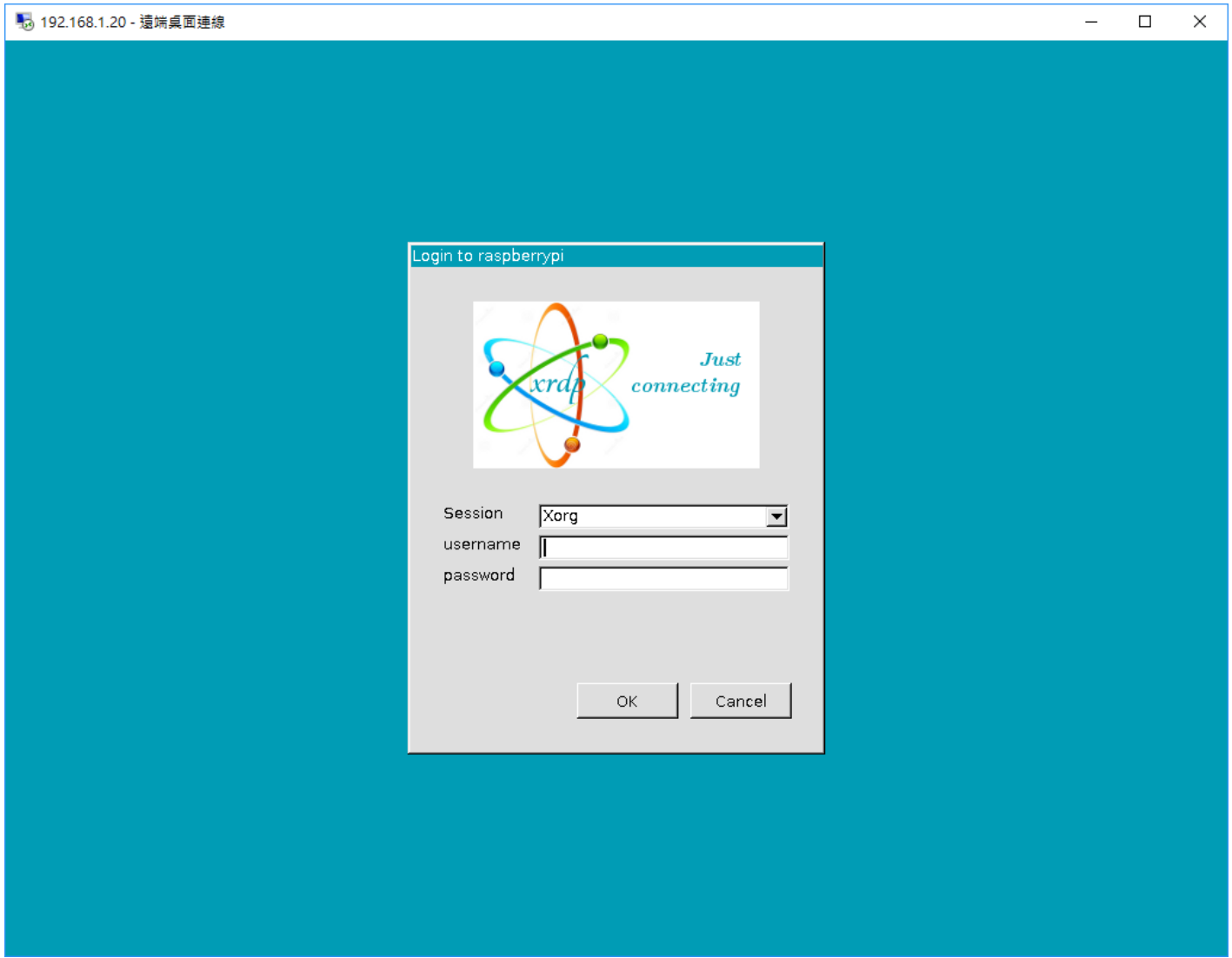


2. PI的環境設定

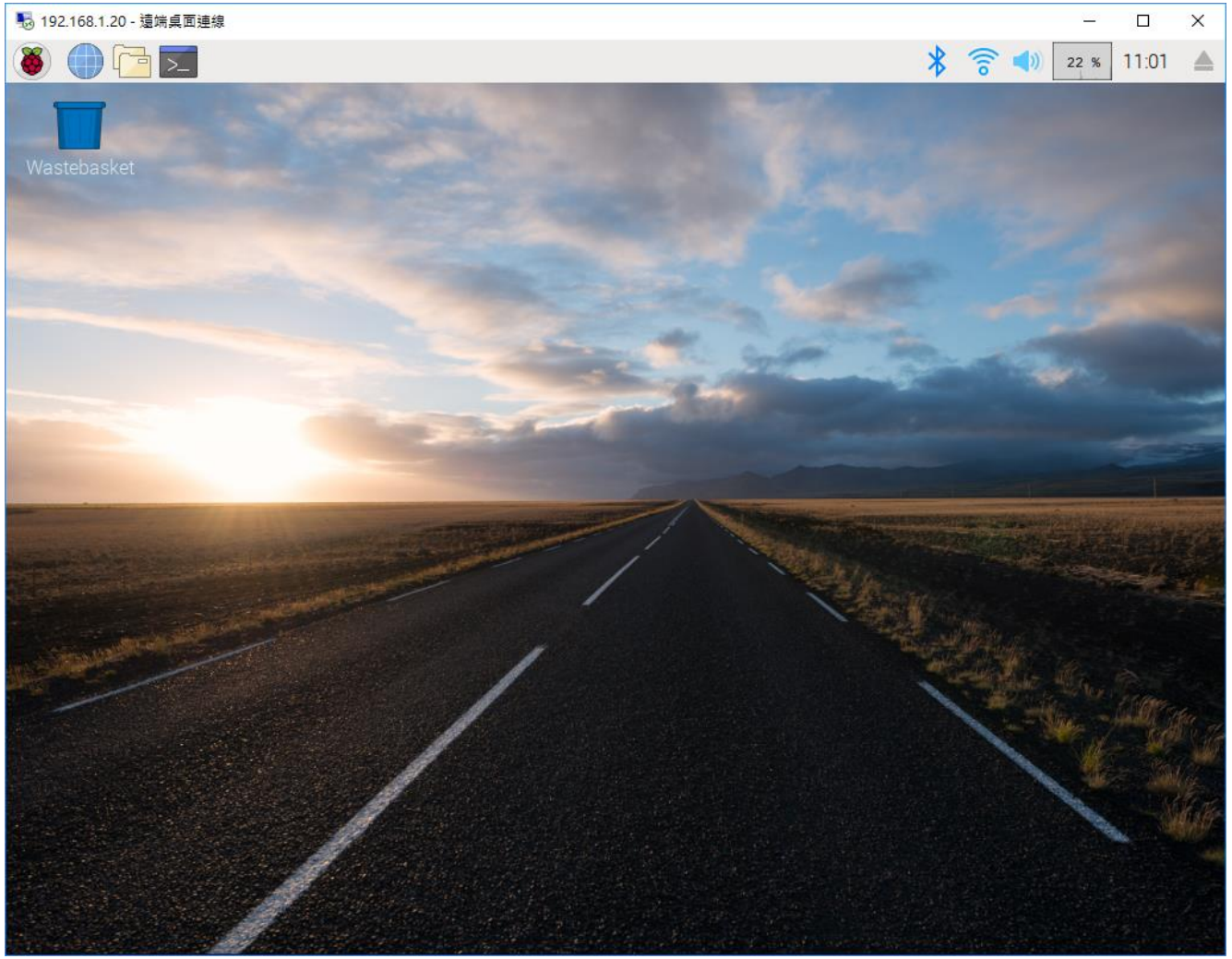
- 網路設定好之後, 可以開始建立想要的服務 ->使用apt-get安裝程式
  - APT = Advanced Packaging Tools
  - 連上網路自動下載程式來安裝
    - (類似Appstore與Google play線上商店的概念)

- Ex: sudo apt-get install xrdp -y
  - 安裝遠端桌面服務
  - 如何得知樹莓派的IP位址? -> 使用 **ifconfig** 指令查詢IP位址

遠端桌面服務(xrdp)


192.168.1.20 - 遠端桌面連線
Login to raspberrypi
xrdp
Just connecting
Session
Xorg
username
password
OK
Cancel

遠端桌面服務(xrdp)



使用apt-get安裝程式

- 更新套件清單: sudo apt-get update
- 安裝套件: sudo apt-get install <pkg_name>

- 搜尋套件: sudo apt-cache search <keyword>
- 移除套件: sudo apt-get remove <pkg_name>
- 升級套件: sudo apt-get upgrade

- 彩蛋: sudo apt-get moo

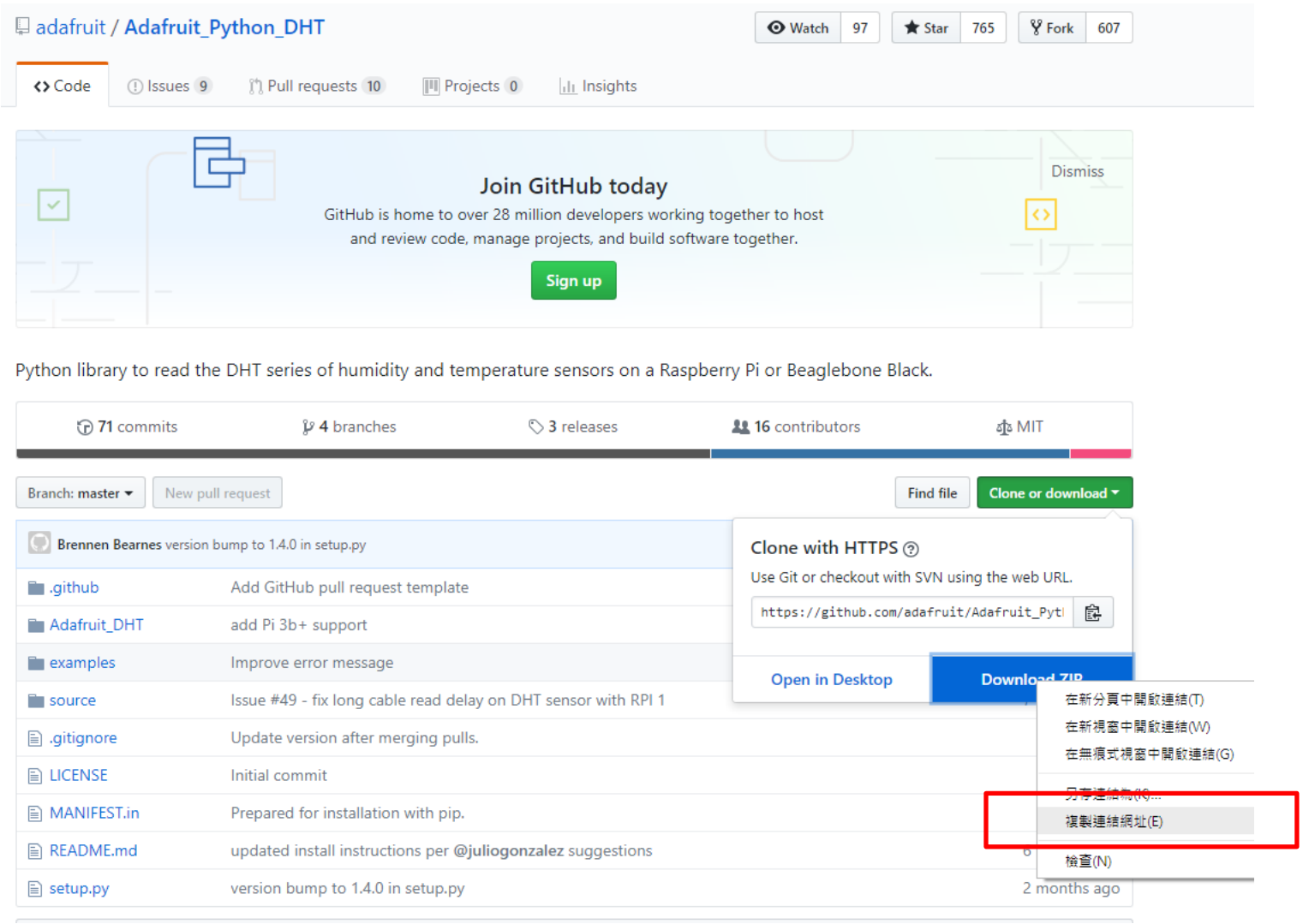
```
pi@raspberrypi:~$ sudo apt-get moo
                 (__)
                 (oo)
           /------\/
          / |    ||
        *  /\---/\
           ~~   ~~
..."Have you mooed today?"...
```

補充: 如何傳輸PI的檔案?

- Ex: 下載溫溼度library
  - https://github.com/adafruit/Adafruit_Python_DHT

- 1. 直接用PI下載檔案
  - wget
- 2.在電腦下載, 透過網路傳輸
  - python, winscp
- 3. 在電腦下載, 用USB隨身碟複製過去
  - Mount

1. 直接用PI下載檔案



複製下載連結

1. 直接用PI下載檔案

- 在終端機執行 wget 指令
  - Ex: wget https://github.com/adafruit/Adafruit_Python_DHT/archive/master.zip

(COM8) [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
pi@raspberrypi:~$ wget https://github.com/adafruit/Adafruit_Python_DHT/archive/m
aster.zip
--2019-01-14 07:57:02--  https://github.com/adafruit/Adafruit_Python_DHT/archive
/master.zip
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/adafruit/Adafruit_Python_DHT/zip/master [f
ollowing]
--2019-01-14 07:57:04--  https://codeload.github.com/adafruit/Adafruit_Python_DH
T/zip/master
Resolving codeload.github.com (codeload.github.com)... 192.30.253.121, 192.30.25
3.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ? 黄aster.zip? ?

master.zip              [   <=>              ]  54.30K   125KB/s    in 0.4s

2019-01-14 07:57:05 (125 KB/s) - ? 黄aster.zip? ? saved [55607]

pi@raspberrypi:~$
```

1. 直接用PI下載檔案

- 下載完畢

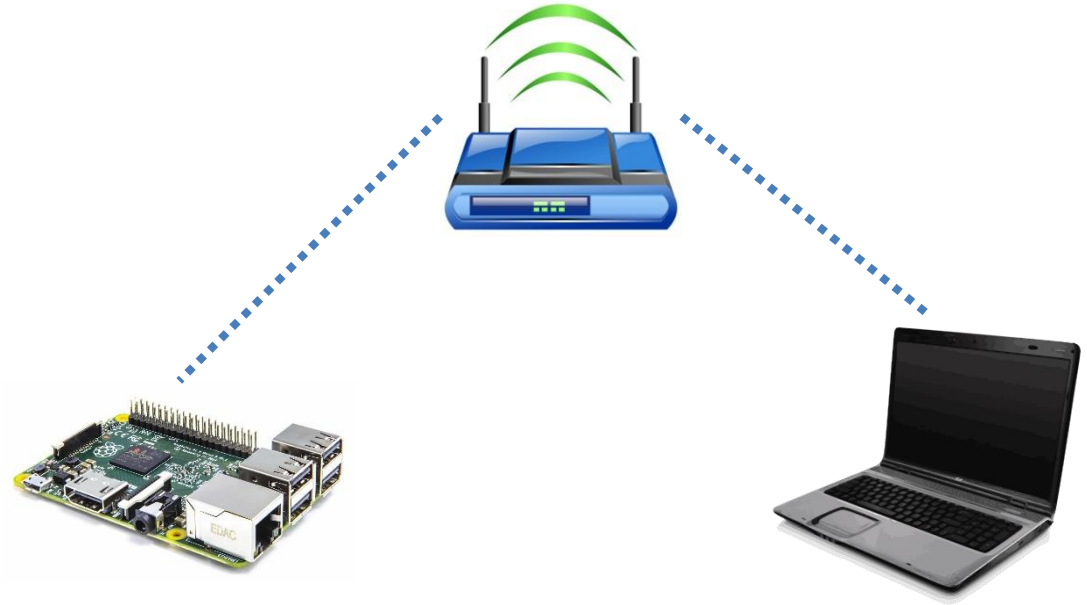
(COM8) [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ? 黃aster.zip? ?

master.zip              [    <=>              ]  54.30K   125KB/s    in 0.4s

2019-01-14 07:57:05 (125 KB/s) - ? 黃aster.zip? ? saved [55607]

pi@raspberrypi:~$ ls -l
total 96
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Desktop
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Documents
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Downloads
drwxr-xr-x 2 pi pi  4096 Nov 13 13:45 MagPi
-rw-r--r-- 1 pi pi 55607 Jan 14 07:57 master.zip
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Music
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Pictures
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Public
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Templates
drwxr-xr-t 2 pi pi  4096 Jan 10 11:00 thinclient_drives
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Videos
pi@raspberrypi:~$
```

2.在電腦下載, 透過網路傳輸

- 網路架構:
  - Case 1: PI 有 Public IP address (通常不可能)
  - Case 2: 電腦與PI在同一個子網路 (較常見)



2.在電腦下載, 透過網路傳輸

- 1. python有內建simple http server
  - 電腦端可以直接透過網頁下載PI裡面的檔案

(COM8) [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
pi@raspberrypi:~$ python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.1.80 - - [14/Jan/2019 08:01:15] "GET / HTTP/1.1" 200 -
192.168.1.80 - - [14/Jan/2019 08:01:16] code 404, message File not found
192.168.1.80 - - [14/Jan/2019 08:01:16] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.80 - - [14/Jan/2019 08:01:26] "GET /Desktop/ HTTP/1.1" 200 -
192.168.1.80 - - [14/Jan/2019 08:01:26] "GET /Desktop/ HTTP/1.1" 200 -
----------------------------------------
Exception happened during processing of request from ('192.168.1.80', 64505)
Traceback (most recent call last):
  File "/usr/lib/python2.7/SocketServer.py", line 290, in _handle_request_nobloc
k
    self.process_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 318, in process_request
    self.finish_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 331, in finish_request
    self.RequestHandlerClass(request, client_address, self)
  File "/usr/lib/python2.7/SocketServer.py", line 654, in __init__
    self.finish()
  File "/usr/lib/python2.7/SocketServer.py", line 713, in finish
    self.wfile.close()
  File "/usr/lib/python2.7/socket.py", line 283, in close
    self.flush()
  File "/usr/lib/python2.7/socket.py", line 307, in flush
```

不安全 | 192.168.1.48:8000

Directory listing for /

- .bash_history
- .bash_logout
- .bashrc
- .cache/
- .config/
- .gnupg/
- .local/
- .nano/
- .profile
- .wget-hsts
- .Xauthority
- .xsession-errors
- .xsession-errors.old
- Desktop/
- Documents/
- Downloads/
- MagPi/
- master.zip
- Music/
- Pictures/
- Public/
- Templates/
- thinclient_drives/
- Videos/

2.在電腦下載, 透過網路傳輸

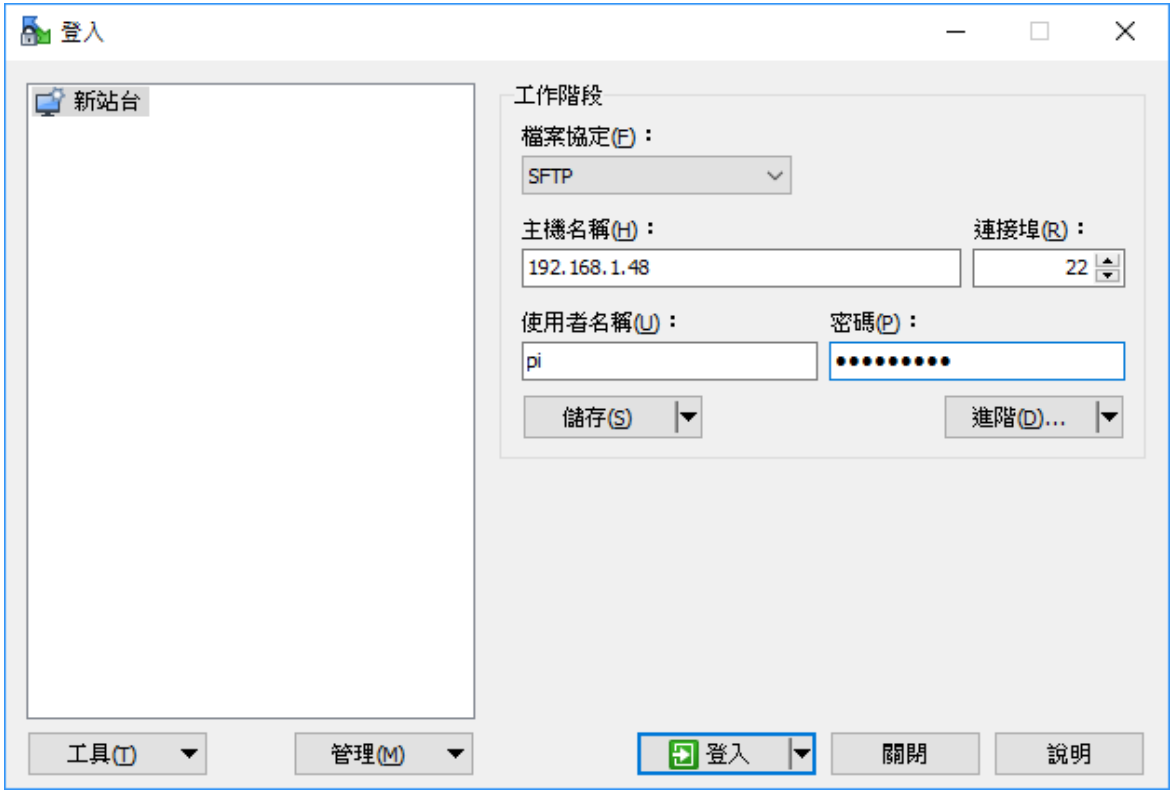
- 2. 在電腦上安裝winscp, 輸入IP的ip address與帳密, 即可連線傳輸資料 (如同ftp介面)
  - 需先啟用ssh login功能
    - 直接在 /boot/ 裡面新增空白檔案, 檔名為ssh
  - 設定指令: sudo touch /boot/ssh

For headless setup, SSH can be enabled by placing a file named ssh, without any extension, onto the boot partition of the SD card from another computer. When the Pi boots, it looks for the ssh file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.

https://www.raspberrypi.org/documentation/remote-access/ssh/

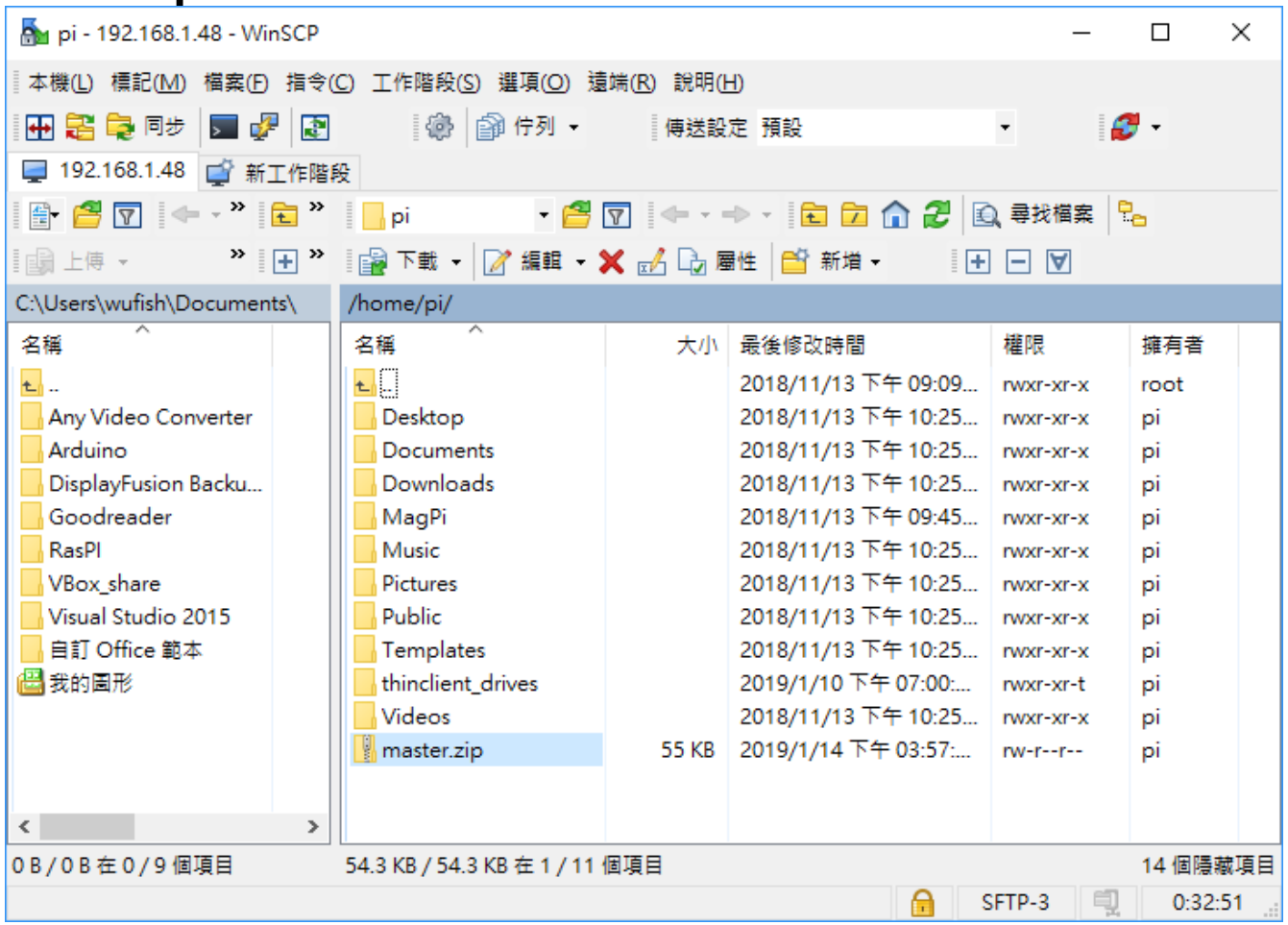
2.在電腦下載, 透過網路傳輸

- 2. winscp: 輸入IP的ip address與帳密, 即可連線傳輸資料 (如同ftp介面)



2.在電腦下載, 透過網路傳輸

- 2. winscp 使用畫面



3. 在電腦下載, 用USB隨身碟複製

- 指令: mount
  - 較複雜, 適用於有linux操作經驗的使用者
  - 優點: 沒有網路也可複製

(COM8) [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
Disk /dev/mmcblk0: 14.9 GiB, 15931539456 bytes, 31116288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf292dfld

Device         Boot Start      End  Sectors  Size Id Type
/dev/mmcblk0p1       8192    98045    89854 43.9M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      98304 31116287 31017984 14.8G 83 Linux


Disk /dev/sda: 3.6 GiB, 3880452096 bytes, 7579008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device     Boot Start     End Sectors  Size Id Type
/dev/sda1  *     8064 7579007 7570944  3.6G  c W95 FAT32 (LBA)
pi@raspberrypi:~$
```

3. 在電腦下載, 用USB隨身碟複製

- 掛載: sudo mount /dev/sda1 /mnt
- 卸載: sudo umount /mnt

(COM8) [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
Device          Boot Start      End  Sectors  Size Id Type
/dev/mmcblk0p1       8192    98045    89854 43.9M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      98304 31116287 31017984 14.8G 83 Linux


Disk /dev/sda: 3.6 GiB, 3880452096 bytes, 7579008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device     Boot Start     End Sectors  Size Id Type
/dev/sda1  *     8064 7579007 7570944  3.6G  c W95 FAT32 (LBA)
pi@raspberrypi:~$ sudo mount /dev/sda1 /mnt/
pi@raspberrypi:~$ ls /mnt/
autorun.inf  isolinux     preseed                    ubnkern
boot         ldlinux.sys  README.diskdefines         ubnpathl.txt
casper       md5sum.txt   syslinux.cfg               wubi.exe
dists        menu.c32     System Volume Information
EFI          pics         ubnfilel.txt
install      pool         ubninit
pi@raspberrypi:~$
```

Summary

- 1. 練習安裝步驟, 使用TTL控制樹莓派

- 額外練習:
  - A. 若有需求, 可嘗試建立遠端桌面連線
  - B. 可嘗試傳輸PI裡面的檔案

2. Raspberry PI GPIO and Python

National Chiao Tung University

Outline

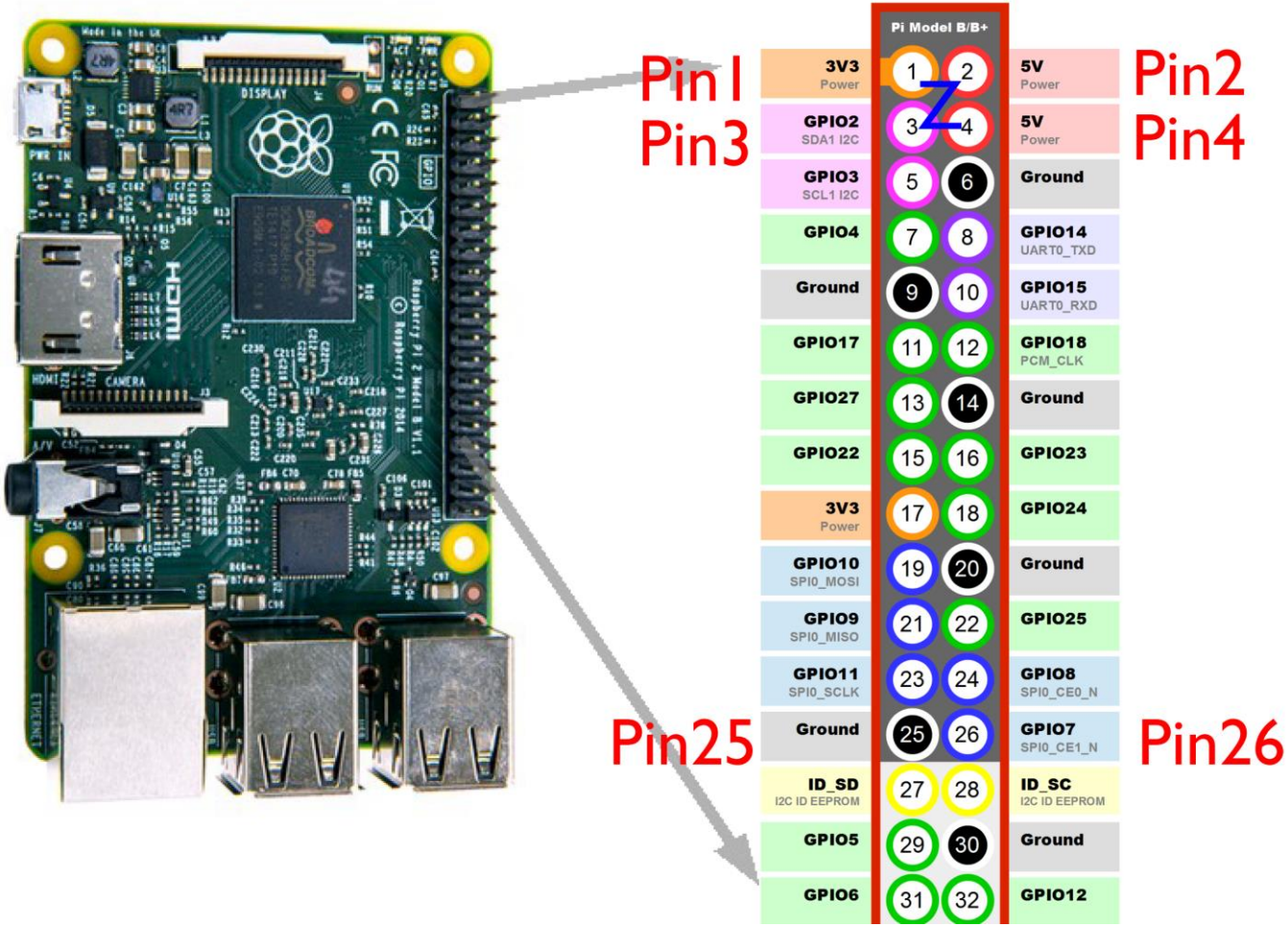
- 0. GPIO介紹, 注意事項
- 1. 安裝LED燈, 電阻的使用方式
- 2. 溫溼度sensor
- 3. 超音波sensor

0. GPIO on Raspberry PI

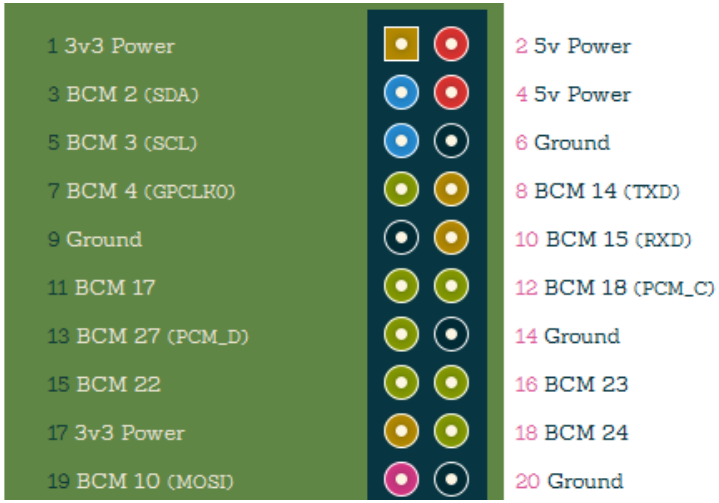
- General-purpose input/output (GPIO)
  - 通用型之輸入輸出(General Purpose I/O)
  - PIN腳可設為輸入(GPI), 輸出(GPO), 輸入與輸出(GPIO)
    - 輸出：寫值到某根腳位; 輸入：從某根腳位讀值
  - 可用軟體控制的數位訊號

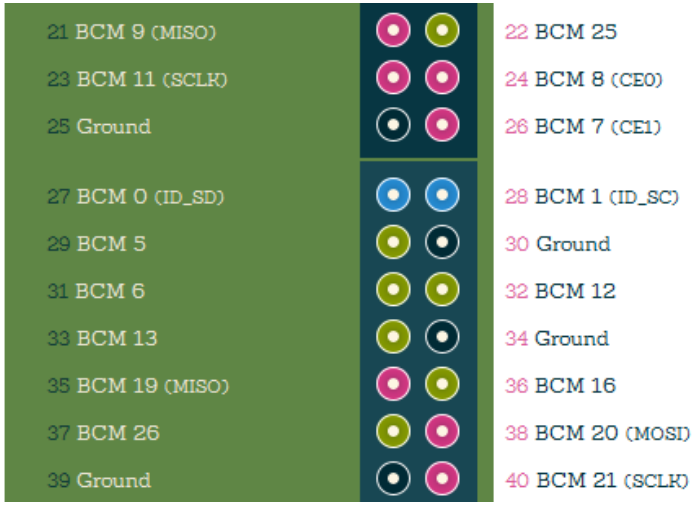
0. GPIO on Raspberry PI

- Z 字型的腳位編號



0. GPIO on Raspberry PI

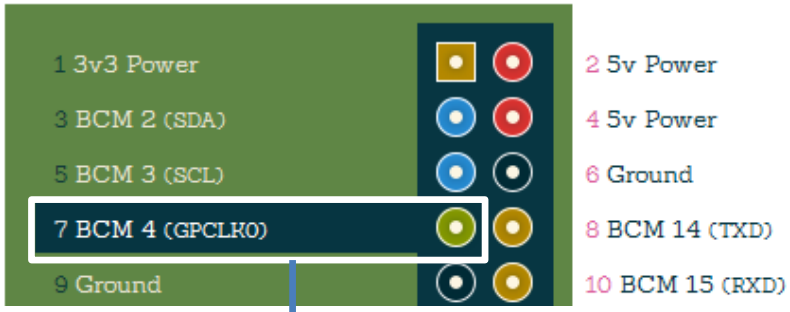




http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/
http://pi.gadgetoid.com/pinout

0. GPIO的腳位與應用

- 注意腳位的編號!!
  - Pin number != GPIO number
    - Physical numbering vs. GPIO numbering
  - Ex: Pin 7 = GPIO 4



BCM 4:
- Physical pin 7
- BCM pin 4
- Wiring Pi pin 7

0. GPIO的注意事項

- 在GPIO pin上不可以輸入超過3.3V
- GPIO PIN的電流上限是 3mA
- 不要拿金屬物體接觸GPIO PIN (會短路)
  - 使用杜邦線, 針腳不要碰到板子
- 用GPIO PIN啟動PI時, 電壓不可以超過5V
  - 不太建議這樣用, 因為容易短路, 也不適合接太多周邊
- 供電腳位(3.3V)不可以超過50mA
- 供電腳位(5V)不可以超過250mA

0. 控制 PI 的 GPIO

- 寫程式
  - C
  - C + wiringPi
  - C#
  - Ruby
  - Perl
  - **Python** ← 我們用這個
  - Scratch
  - Java Pi4J Library
  - Shell script

0. Python 簡介

- 程式架構可大致分為這幾類
  1. 變數 , 物件 , 註解
  2. 模組
  3. 縮排
  4. 迴圈
  5. 條件判斷
  6. 函式

- PS. PI預設使用Python 2

(1). 變數，物件，註解

- 動態型別 (dynamic typing)

```
# 井號是註解

i = 3                    # 變數 i 指到數字物件 3
print i                  # 印出 i

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[0]               # 印出字串中第一個元素
```

(2). 模組

- 模組

```
# import MODULE
import RPi.GPIO

# import MODULE as ALIAS
import RPi.GPIO as GPIO
```

可以使用該模組的函式

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
```

(3). 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

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

前面有 4 個空白鍵

(4). 迴圈

- 自動迭代 (iterator); for loop

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


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

前面有 4 個空白鍵

https://docs.python.org/2/tutorial/controlflow.html

(5). 條件判斷

- If statement

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

```
if value > 20:
    print("lalala")
else:
    print(“XD")
```

https://docs.python.org/2/tutorial/controlflow.html

(6). 函式

- 自訂函式 (User-Defined Functions)

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

```
def f(x):
    return x**2 + 1
```

f(4) 可以得到 17

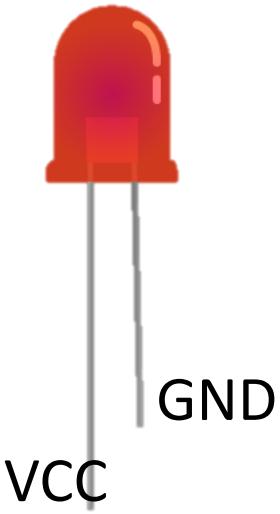
http://www.pitt.edu/~naraehan/python2/user_defined_functions.html

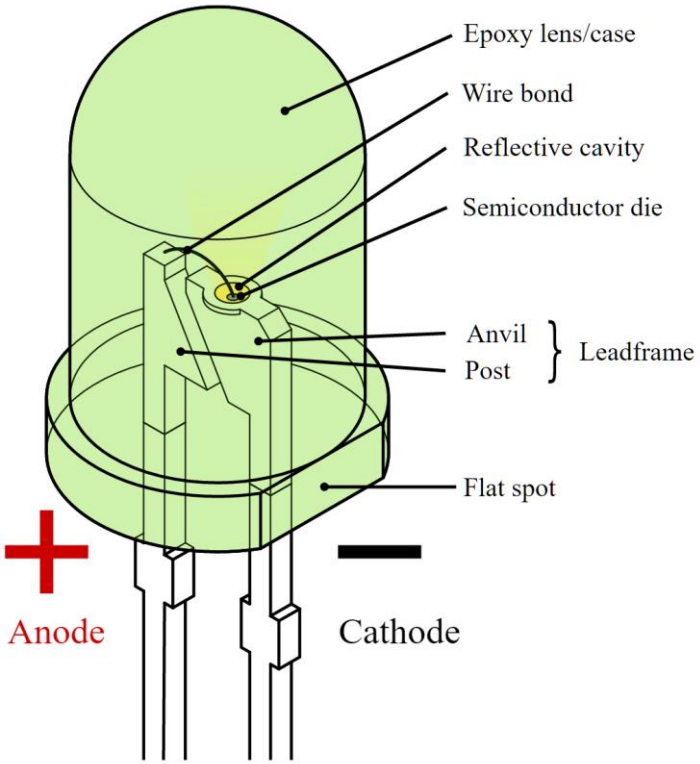
準備python環境

- 先安裝程式&相關函式庫
  - sudo apt-get update
  - sudo apt-get install -y python-dev python-pip python-rpi.gpio

1. LED燈

- 發光二極體 (Light-emitting diode)
  - 長腳接正極 VCC
  - 短腳接負極 GND
  - LED的規格&特性!?





https://zh.wikipedia.org/wiki/發光二極管

1. LED燈

- LED的電流最好不要超過20mA
  - 超過也可以, 只是容易壞掉...

Electrical / Optical Characteristics at TA=25°C

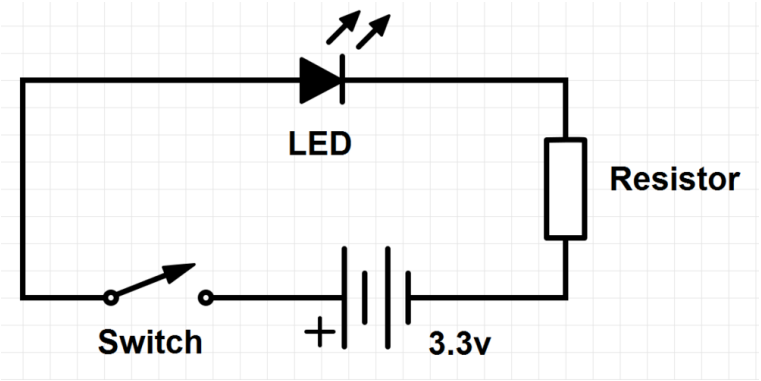
Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Super Bright Red	660		nm	I_F=20mA
λD [1]	Dominant Wavelength	Super Bright Red	640		nm	I_F=20mA
Δλ1/2	Spectral Line Half-width	Super Bright Red	20		nm	I_F=20mA
C	Capacitance	Super Bright Red	45		pF	V_F=0V;f=1MHz
V_F [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	I_F=20mA
I_R	Reverse Current	Super Bright Red		10	uA	V_R = 5V

Notes:
1.Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

LED的順向電壓是1.85V, 電流需要20mA

1. LED燈

- 根據前面的參數
  - PI的GPIO腳位可以提供3.3V
  - LED的順向電壓是1.85V
  - LED的電流需要20mA



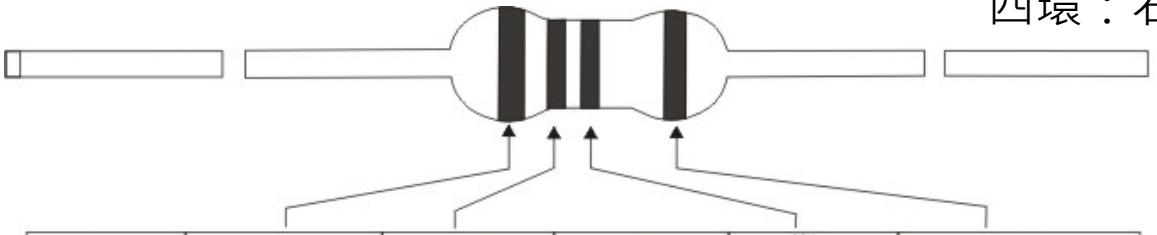
- 電阻公式: R=V/I
  - R =(3.3-1.85)/0.02=72.5 歐姆
  - 最少須接 72.5 歐姆的電阻，才可避免 LED燒毀

https://www.raspberrypi.org/documentation/usage/gpio/

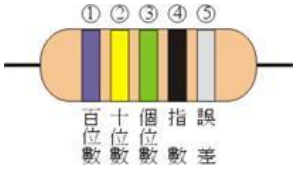
1. LED燈

- 電阻的判斷方式?

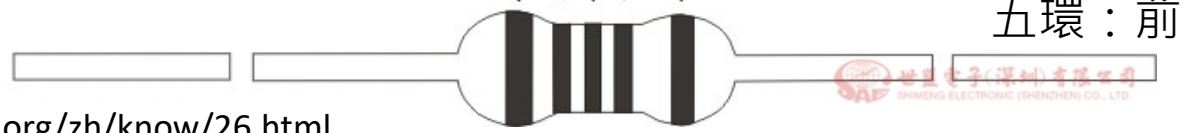
四環：右側為金或銀



顏色 Color	第1道 1nd Band	第2道 2nd Band	第3道 3nd Band	倍率 Multiple	誤差值% Tolerance
黑 Black	0	0	0	10^0	
棕 Brown	1	1	1	10^1	F(±1%)
紅 Red	2	2	2	10^2	G(±2%)
橙 Orange	3	3	3	10^3	
黃 Yellow	4	4	4	10^4	
綠 Green	5	5	5	10^5	D(±0.5%)
藍 Blue	6	6	6	10^6	C(±0.25%)
紫 Violet	7	7	7	10^7	B(±0.10%)
灰 Gray	8	8	8	10^8	
白 White	9	9	9	10^8	
金 Gold				10^{-1}	J(±5%)
銀 Silver				10^{-2}	K(±10%)



五環：前四環很靠近

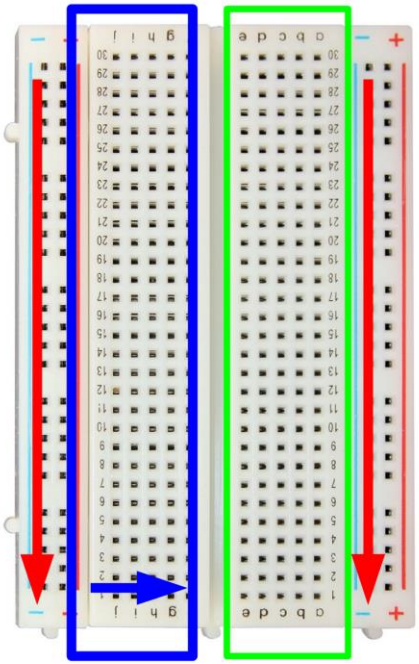


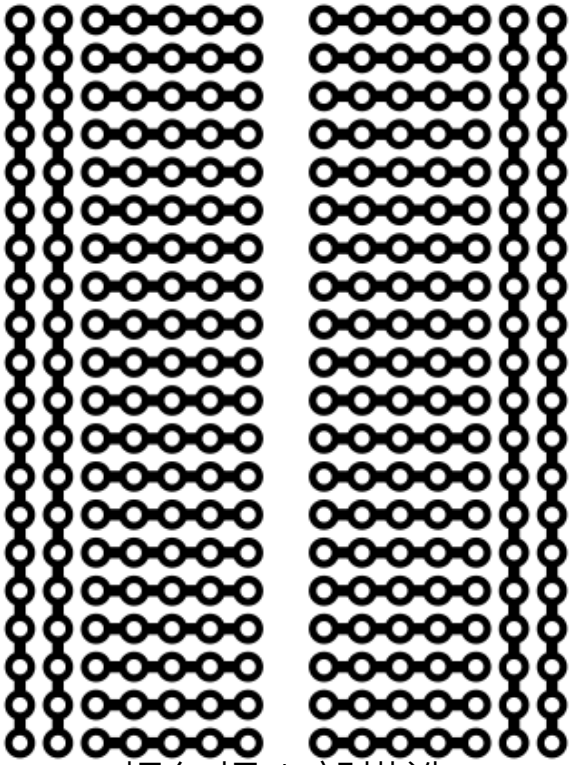
http://www.shimeng.org/zh/know/26.html

1. LED燈

- 麵包板的使用

1. 藍色和綠色兩塊不通
2. 藍色水平相通
3. 紅色垂直相通



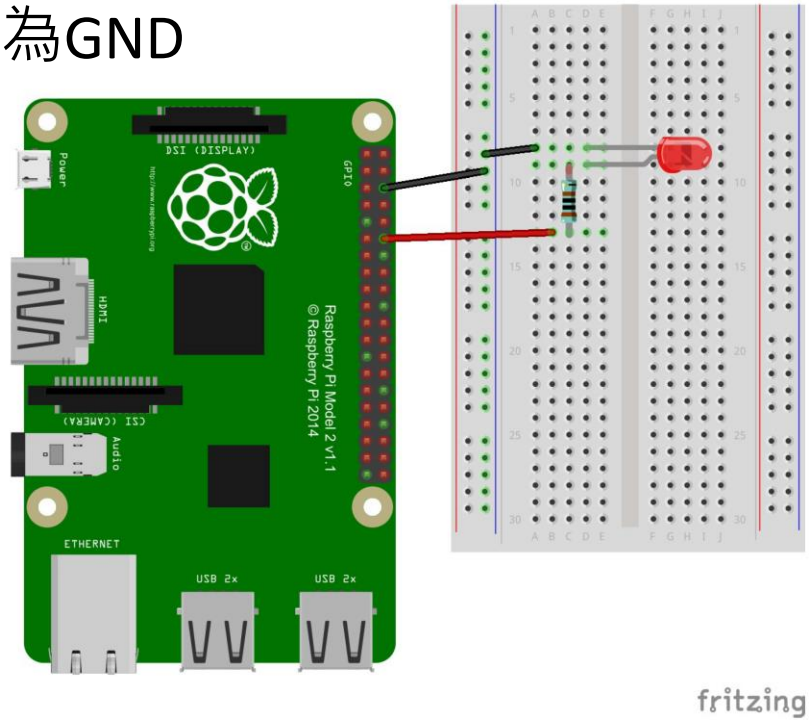


麵包板內部構造

1. LED燈

- 連接LED燈
  - LED的長腳位為VCC, 短腳位為GND
  - VCC 接至 PI 的 Pin 12
  - GND 接至 PI 的 Pin 6

  - 使用1k Ω電阻
  - Q: 1k 電阻長什麼樣子?



1. 控制LED燈

- Python code (LED閃爍)

```python
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
LED_PIN = 12
GPIO.setup(LED_PIN, GPIO.OUT)

while True:
    print("LED is on")
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(1)
    print("LED is off")
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(1)

GPIO.cleanup()
```

載入函式庫

GPIO.BOARD:
按照腳位的順序編號

使用NO. 12的pin腳

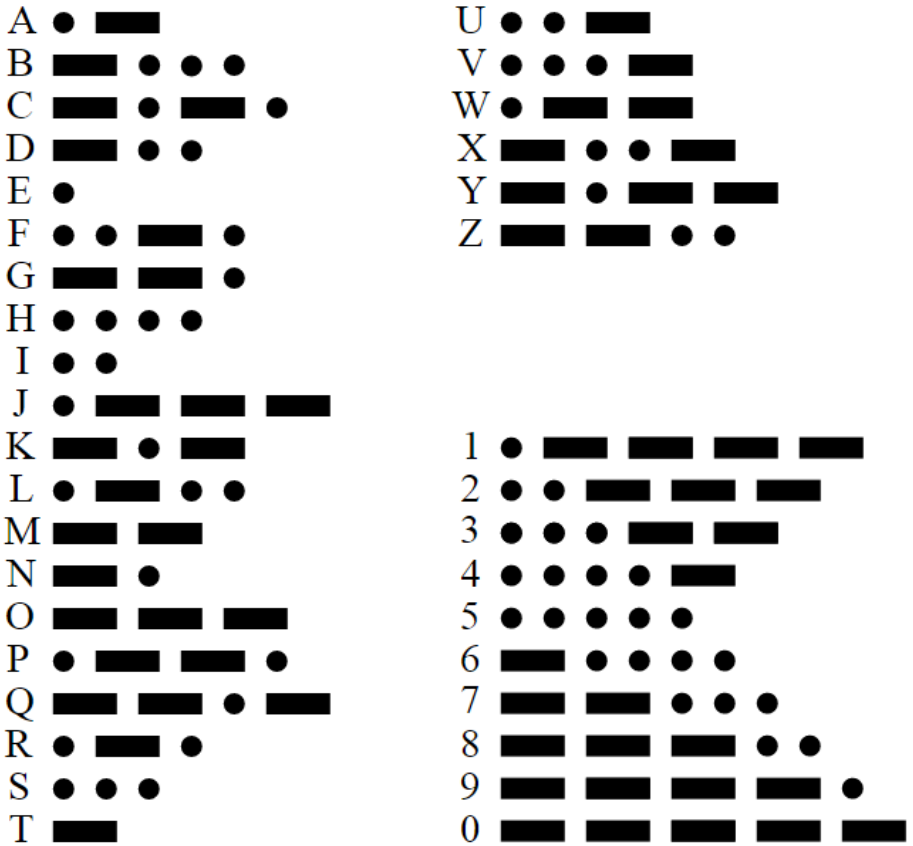
載入函式庫

1. 控制LED燈

★ □ Quiz1: 用LED產生SOS的摩斯密碼

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

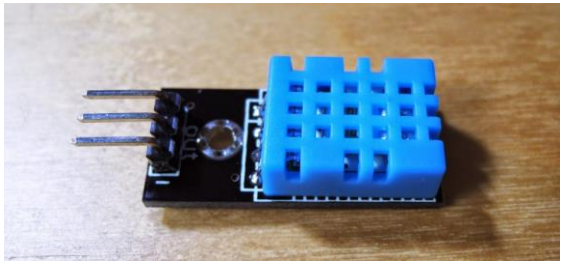
PS. 可設定 one unit = 1 sec

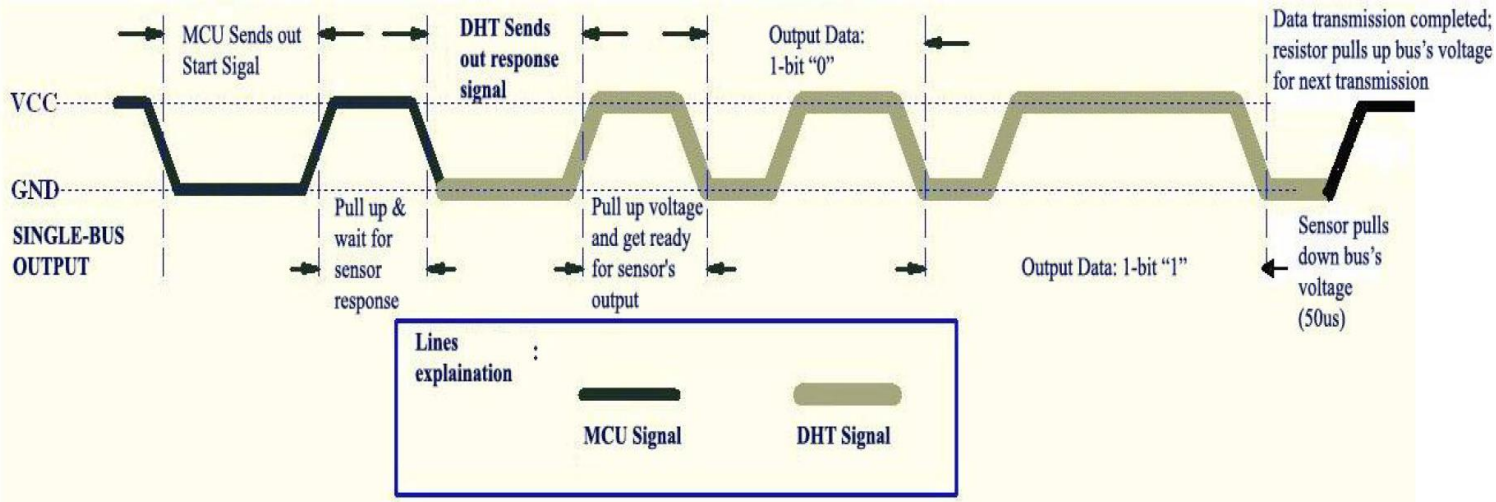


https://en.wikipedia.org/wiki/Morse_code

2. 溫濕度sensor

- DHT11
  - 溫濕度感應器
  - 溫度: 0~50℃, 誤差±2 ℃
  - 濕度: 20~90%, 誤差±5%
  - 使用三個腳位: VCC, Data , GND (+, out, -)

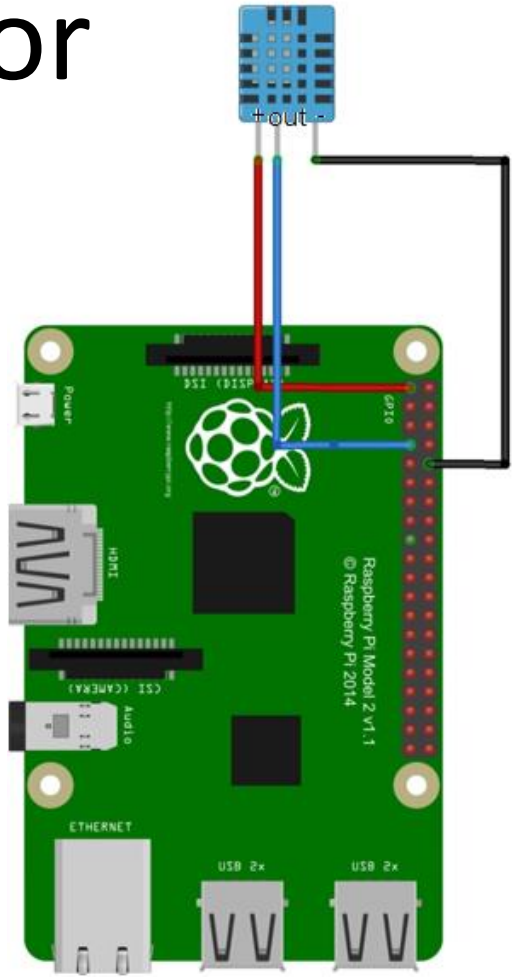




https://cdn-shop.adafruit.com/datasheets/DHT11-chinese.pdf
http://www.micropik.com/PDF/dht11.pdf

2. 溫濕度sensor

- 安裝Adafruit的模組與範例程式
  - Source code: https://github.com/adafruit/Adafruit_Python_DHT
- 先安裝編譯程式
  - sudo apt-get update
  - sudo apt-get install git-core build-essential python-dev



2. 溫濕度sensor

- 下載source code
  - 在終端機輸入下列指令
  - git clone https://github.com/adafruit/Adafruit_Python_DHT.git
- 安裝Adafruit_Python_DHT程式
  - cd Adafruit_Python_DHT
  - sudo python setup.py install
- 補充: DHT11/22驅動程式學習套件補充資訊
  - https://hackpad.com/DHT1122-6YNZZMytoLD

2. 擷取溫濕度資訊

- 執行程式
  - $ cd examples
  - $ sudo ./AdafruitDHT.py 11 4
    - 使用 DHT 11 感應器 (另有DHT 22, DHT 2302)
    - 連接 GPIO 4 (也就是Pin 7)
    - 得到資訊: Temp=26.0*C Humidity=37.0%

Code第一行是環境變數, 需保留:
#!/usr/bin/python

```
pi@raspberrypi ~ $ cd Adafruit_Python_DHT/examples/
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 11 4
Temp=26.0*  Humidity=37.0%
```

- 調整 溫度 & 濕度 的顯示方式
  - Method 1: 修改AdafruitDHT.py, 改為自己想要的格式
  - Method 2: 利用現有的輸出格式, 去抓取想要的範圍 (ex: shell script)

2. 擷取溫濕度資訊

- Python code:

```
import sys
import Adafruit_DHT

sensor_args = { '11': Adafruit_DHT.DHT11,
                '22': Adafruit_DHT.DHT22,
                '2302': Adafruit_DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
    sensor = sensor_args[sys.argv[1]]
    pin = sys.argv[2]
else:
    print('usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#')
    print('example: sudo ./Adafruit_DHT.py 2302 4 - Read from an AM2302 connected to GPIO #4')
    sys.exit(1)

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

if humidity is not None and temperature is not None:
    print('Temp={0:0.1f}*  Humidity={1:0.1f}%'.format(temperature, humidity))
else:
    print('Failed to get reading. Try again!')
    sys.exit(1)
```

載入函式庫

讀取.py後的參數

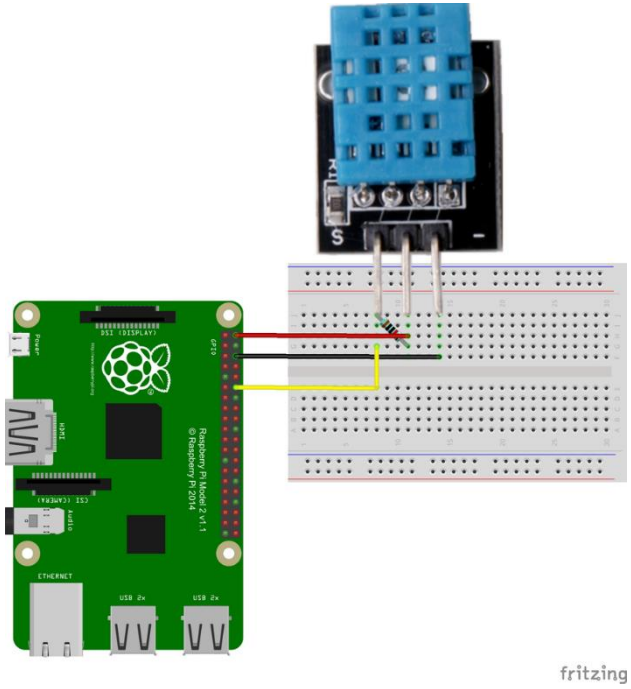
本程式的執行方式 (後面要接2個參數)

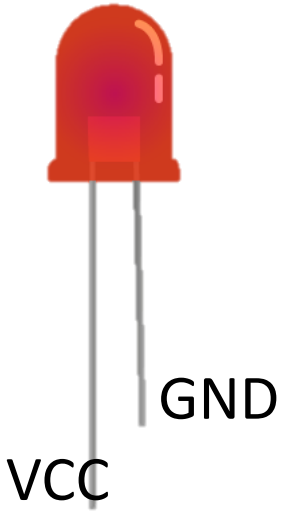
導入.py後的2個輸入參數
(Adafruit_DHT/common.py)

回傳數值

2. 擷取溫濕度資訊

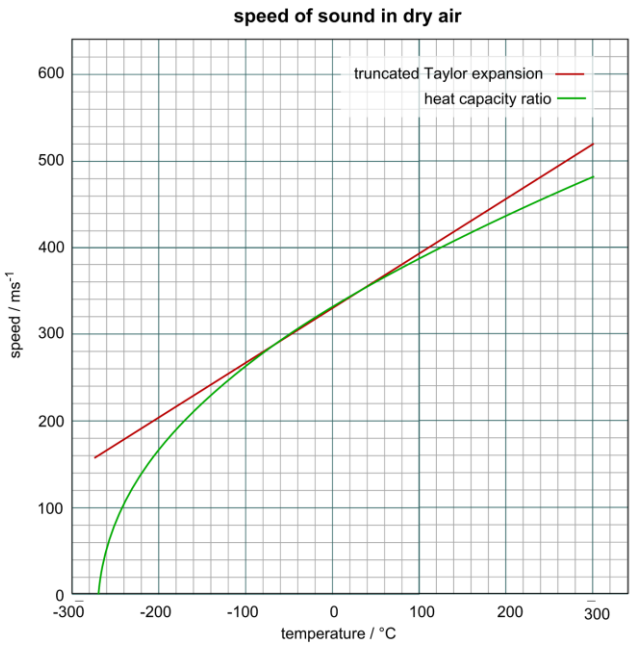
- Quiz2: 設計一個溫度警示燈
  - 溫度大於特定值(ex: 25度), 開啟LED燈號





3. HC-SR04 超音波感測器

- 測距離原理
  - 聲音在 20°C (68°F) 的速度是 343 公尺 / 秒
  - 溫度每增加 1°, 速度增加 0.6 公尺 / 秒



$$Speed = \frac{Distance}{Time}$$

$$34300 = \frac{Distance}{Time/2}$$

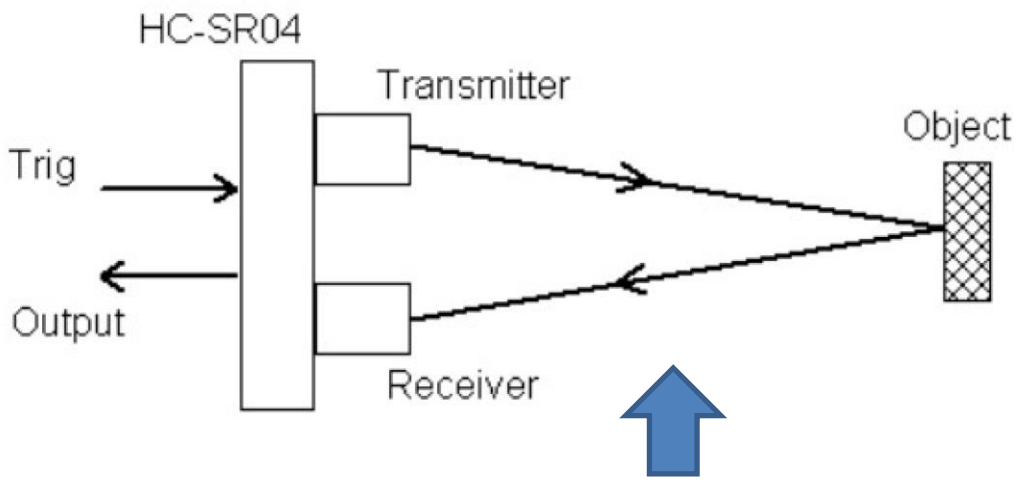
$$17150 = \frac{Distance}{Time}$$

$$17150 \times Time = Distance$$

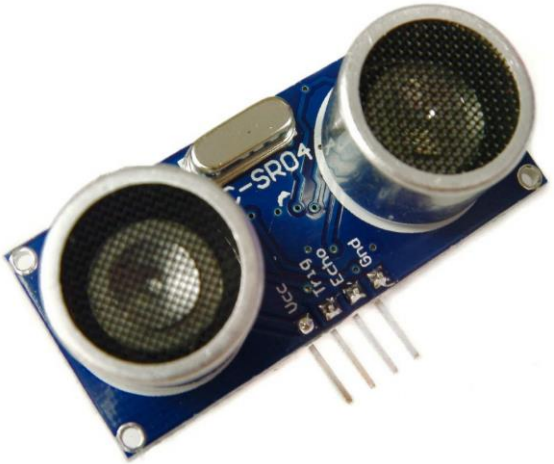
http://en.wikipedia.org/wiki/Speed_of_sound
https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

3. HC-SR04 超音波感測器

- 內建發射 (40kHz) 與接收電路
- 根據發射與接收的時間差計算距離
- 特殊功能 :US-020(長距離) 、 US-100(溫度補償)



可得到時間差



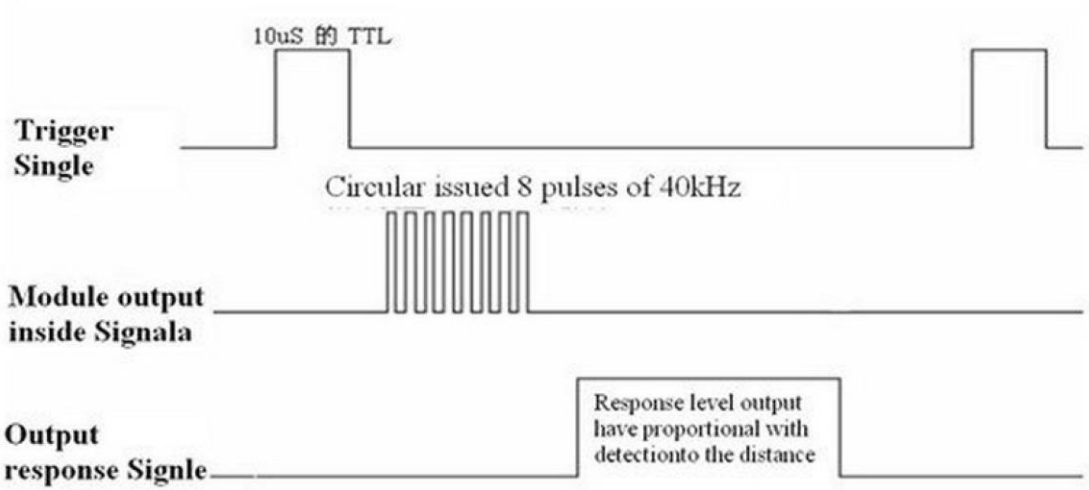
VCC, Trigger, Echo, GND

3. HC-SR04 超音波感測器

- 規格與時序圖

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

4、Ultrasound timing diagram

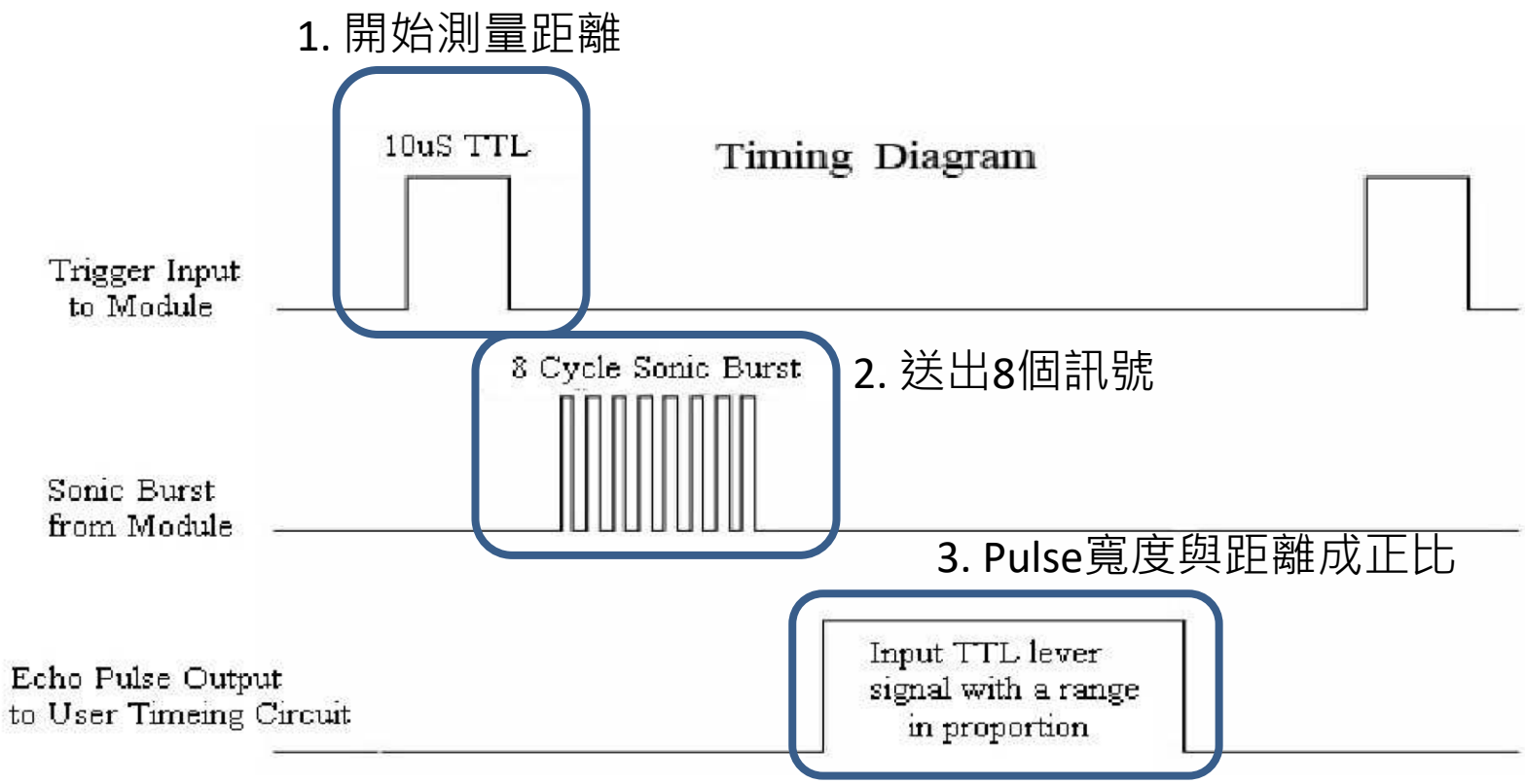


建議測量間隔 >60ms

http://coopermaa2nd.blogspot.tw/2012/09/hc-sr04.html

3. HC-SR04 超音波感測器

- 時序圖(Timing diagram)



http://www.micropik.com/PDF/HCSR04.pdf

3. HC-SR04 超音波感測器

- TRIG 腳位收到高電位 (3.3V) 後發送超聲波
- ECHO 腳位維持低電位 (0V), 收到回應後拉到高電位 (5V)
- Raspberry Pi 腳位的容忍電位為 3.3V
  - => 將 ECHO 腳位的 5V 降壓為 3.3V 左右

$$\frac{3.3}{5} = \frac{R2}{1000 + R2}$$

$$0.66 = \frac{R2}{1000 + R2}$$

$$0.66(1000 + R2) = R2$$

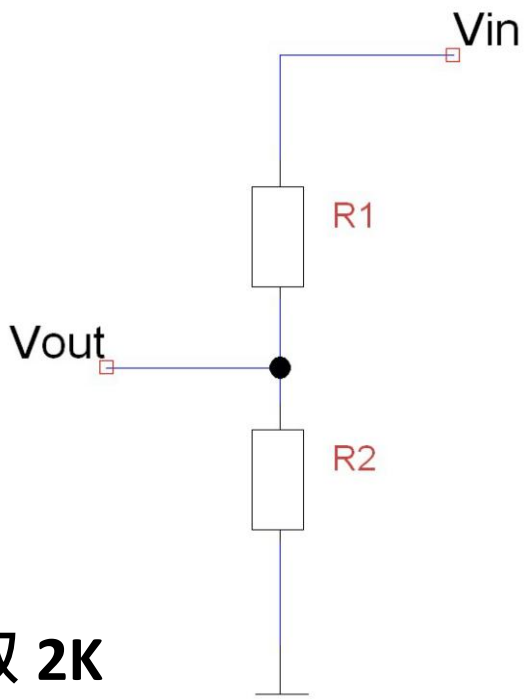
$$660 + 0.66R2 = R2$$

$$660 = 0.34R2$$

$$1941 = R2$$

$$Vout = Vin \times \frac{R2}{R1 + R2}$$

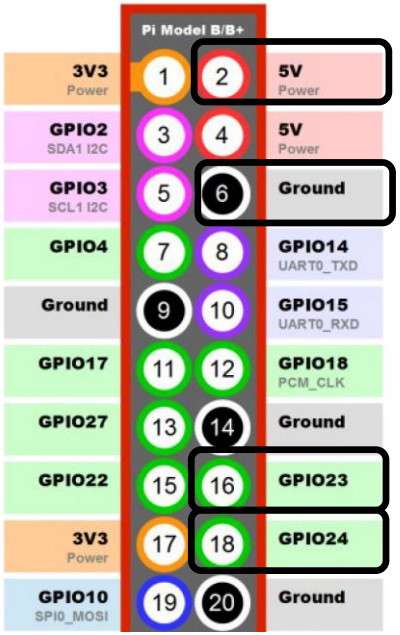
$$\frac{Vout}{Vin} = \frac{R2}{R1 + R2}$$

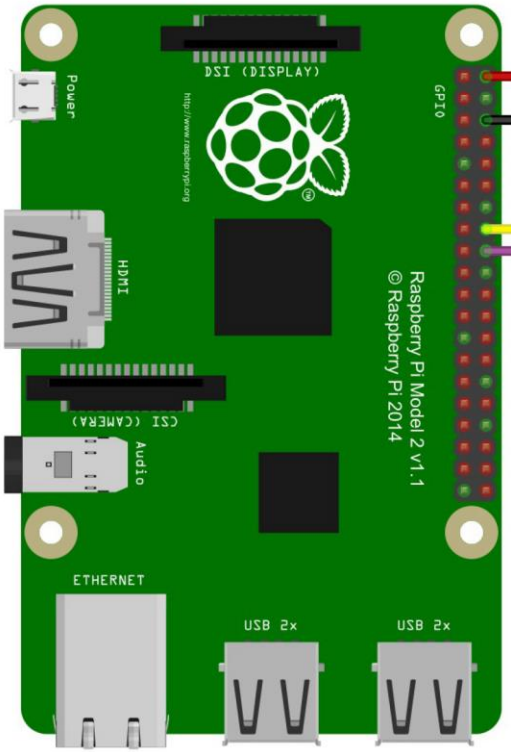


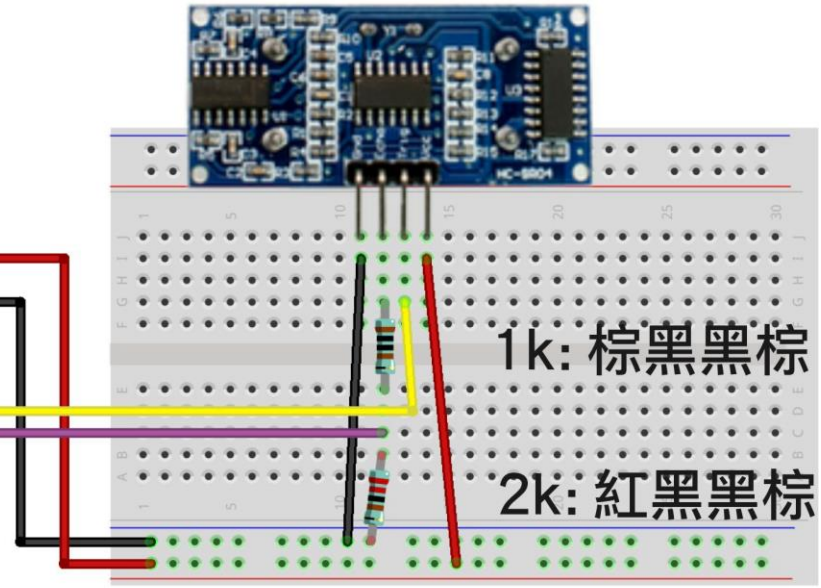
計算結果: R1=1K, R2 取 2K

https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

3. HC-SR04 超音波感測器







ULTRASONIC	RPi	
Vcc(RED)	Pin2	(5V)
Trig(YELLOW)	Pin16	(GPIO23)
Echo(PURPLE)	Pin18	(GPIO24)
Grnd(BLACK)	Pin6	(Ground)

3.超音波(Python)

```python
#!/usr/sbin/env python
import RPi.GPIO as GPIO
import time

v = 343          # (331 + 0.6*20)
TRIGGER_PIN = 16
ECHO_PIN = 18
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIGGER_PIN,GPIO.OUT)
GPIO.setup(ECHO_PIN,GPIO.IN)

def measure() :
    GPIO.output(TRIGGER_PIN, GPIO.HIGH)
    time.sleep(0.00001)      # 10uS
    GPIO.output(TRIGGER_PIN, GPIO.LOW)

    while GPIO.input(ECHO_PIN) == GPIO.LOW:
        pulse_start = time.time()
    while GPIO.input(ECHO_PIN) == GPIO.HIGH:
        pulse_end = time.time()
    t = pulse_end - pulse_start
    d = t * v
    d = d/2
    return d*100

print measure()
GPIO.cleanup()
```

載入函式庫

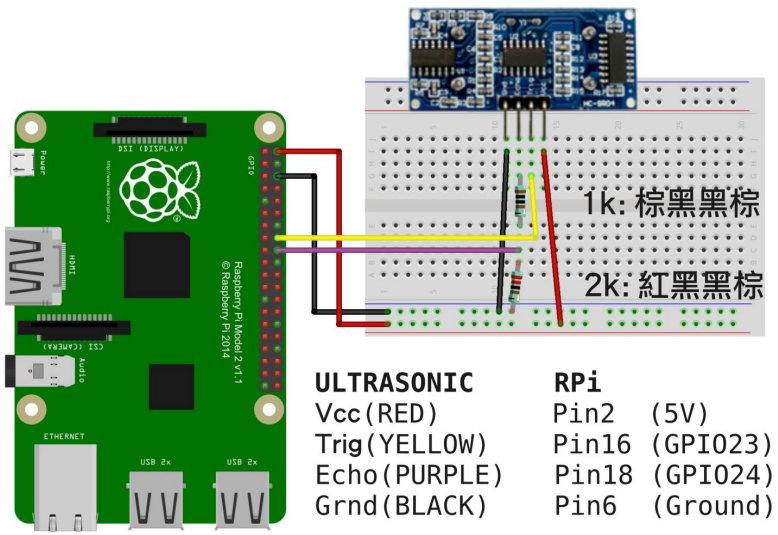
第一行是環境變數
很重要, 不可以刪掉!!

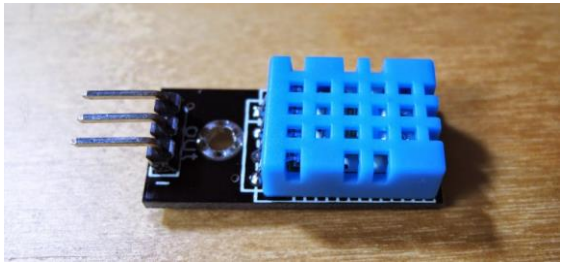
在Trigger上傳送10us的訊號

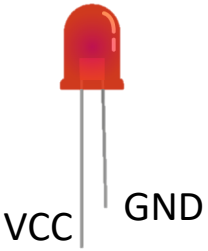
在Echo上偵測Pulse寬度

3. 測量距離

- Quiz3: DIY倒車雷達
  - 結合LED+溫濕度+超音波
  - 距離需搭配溫度校正
  - 距離30cm以上, LED沒反應
  - 小於30cm, LED開始閃爍
  - 小於15cm, LED的閃爍變快







Summary

- Practice Lab (LED, DHT11, ultrasonic)
- Write down the answer for question
  - Q: 1k 電阻長什麼樣子?

- Write code for Quiz 1 - 3, then demonstrate it to TAs
  - Quiz1: 用LED產生SOS的摩斯密碼
  - Quiz2: 設計一個溫度警示燈
  - Quiz3: DIY倒車雷達

3. I/O of Raspberry PI – part 1

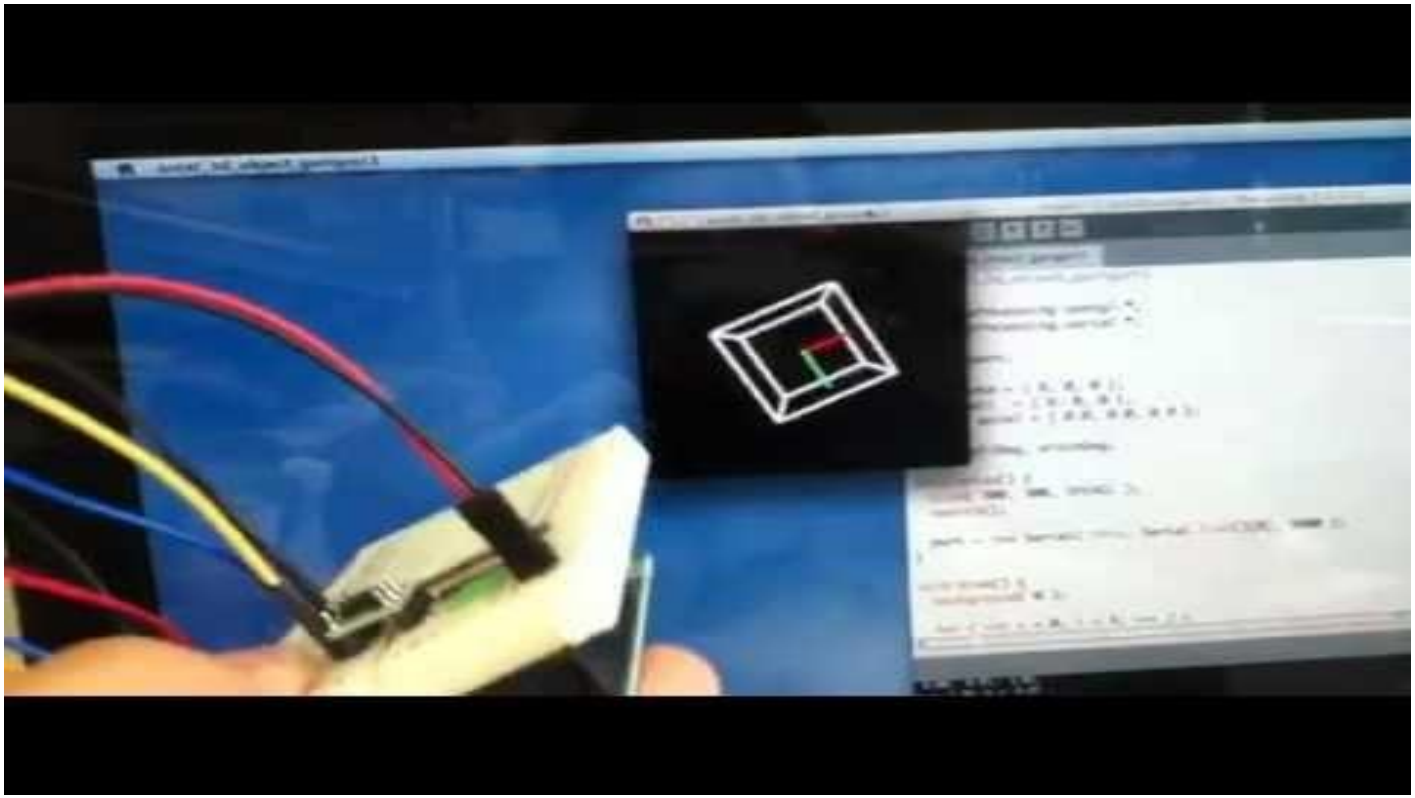
National Chiao Tung University

Demo video

- Open Source IMU and AHRS Algorithm with x-IMU
  - https://www.youtube.com/watch?v=BXsGWoOMtmU

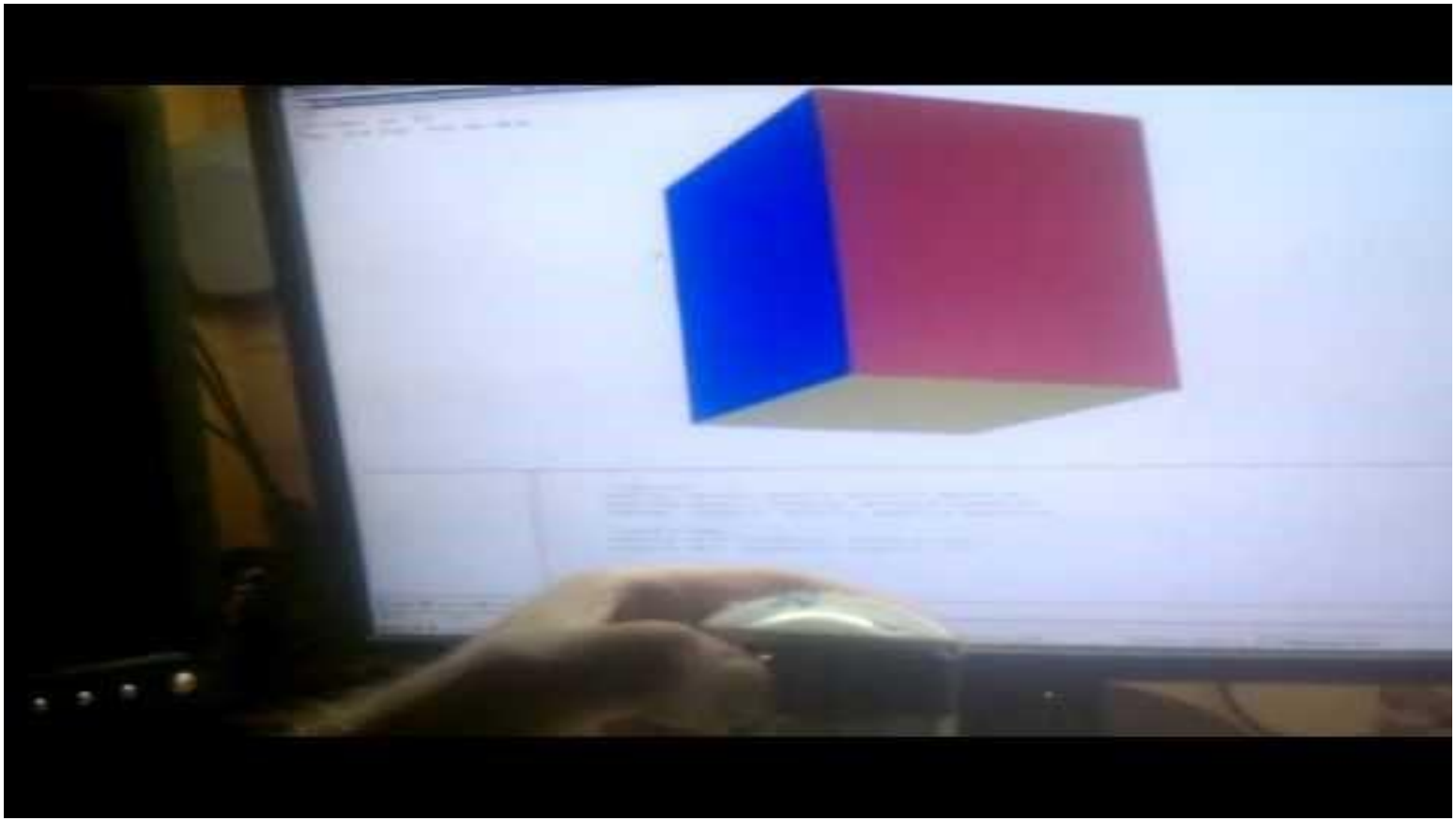
Demo video

- 3D Box on screen rotated around its roll and pitch axes through an acceleration sensor
  - https://www.youtube.com/watch?v=85rl3KVAEGU



Demo video

- 9DOF IMU Test
  - https://www.youtube.com/watch?v=C7DVH3goMUA



真實產品

龍珠雷達 XD

移動方式：
利用**方位感測器**與**三軸加速度**，
配合移動來接近光源。

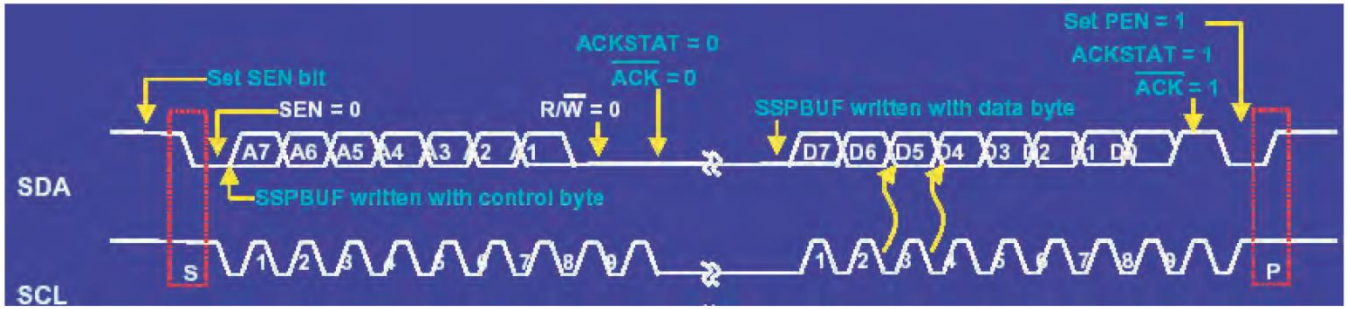
Outline

- 0. 啟用I2C介面
- 1. 加速度計
- 2. 陀螺儀
- 3. 電子羅盤

這個念square喔!!

I2C

- I2C全名為Inter-IC，它是一種半雙工同步多組設備匯流排，只需要兩條信號線：串列資料線 (SDA) 及串列時脈線 (SCL)。

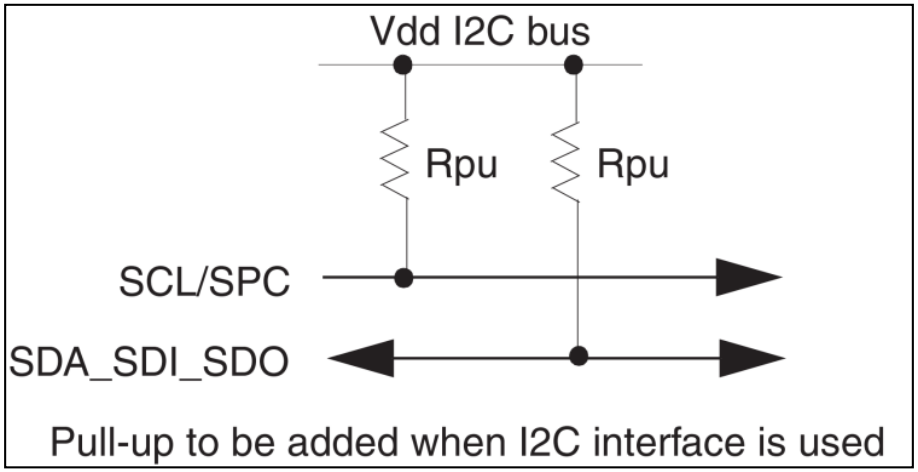


- 在傳送資料過程中共有三種類型信號，分別是：開始信號、結束信號和應答信號。
  - 開始信號：SCL為高電位時，SDA由高電位降為低電位，開始傳送資料。
  - 結束信號：SCL為高電位時，SDA由低電位升為高電位，結束傳送資料。
  - 應答信號：收到 8bit 資料後，向發送資料的IC發出特定的低電位脈衝
- 資料讀取方式：
  - 當 SCL 由低電位升為高電位時，讀取 SDA 的資料。

http://www.leaptronix.com/PDF/introducing_of_I2C.pdf

I2C

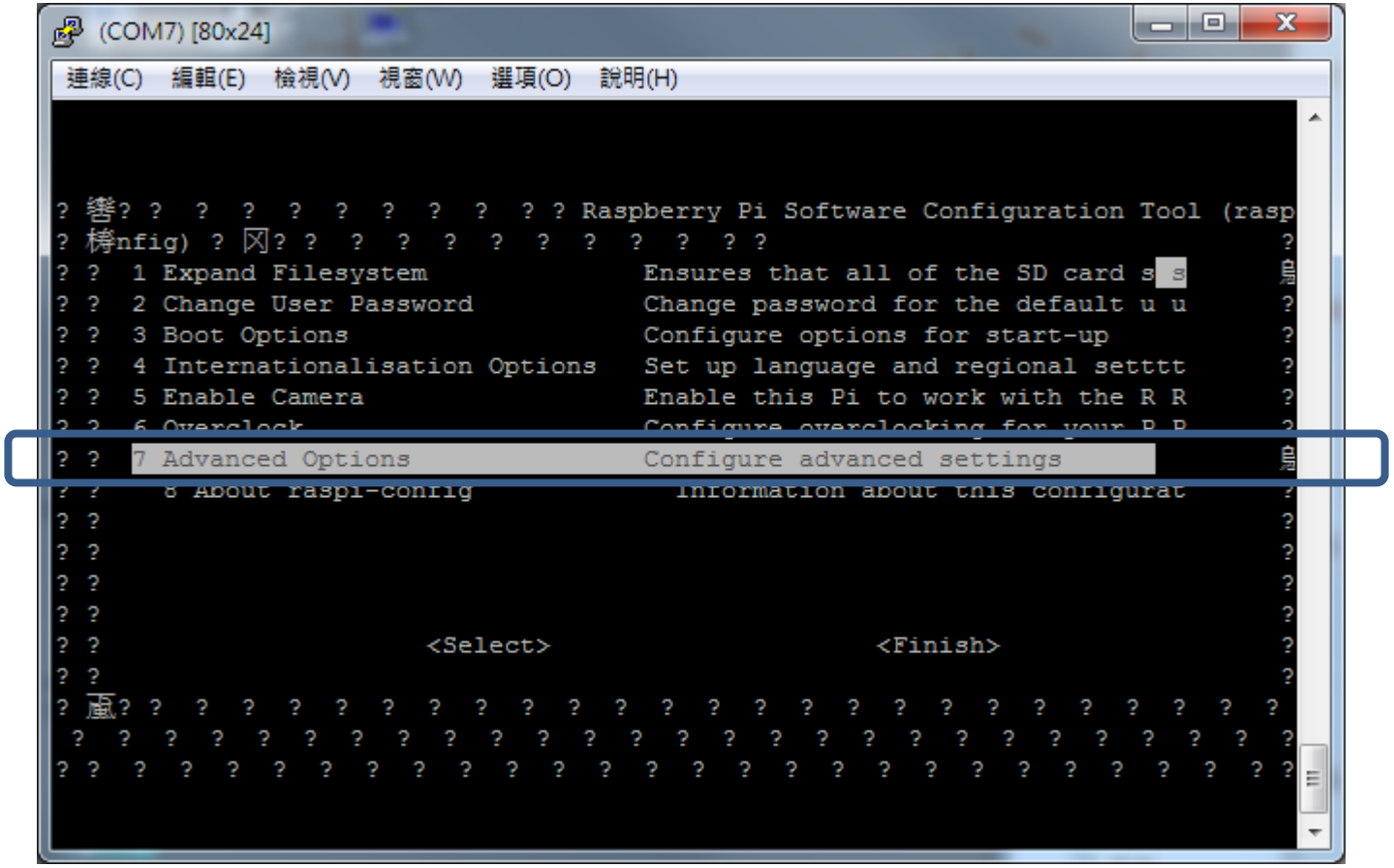
- 當 SCL 由低電位升為高電位時，讀取 SDA 的資料。
- 在 SDA、SCL 加入上拉電阻(Pull-up resister)，使未導通狀態下線路會處於邏輯高準位(High)狀態，告訴晶片是走 I2C 的協定
  - 上拉電阻：當某輸入埠未連接設備或處於高阻抗的情況下，上拉電阻可用於保證輸入訊號為預期邏輯電平的電阻元件。他們通常在不同的邏輯器件之間工作，提供一定的電壓訊號。



https://zh.wikipedia.org/wiki/上拉电阻

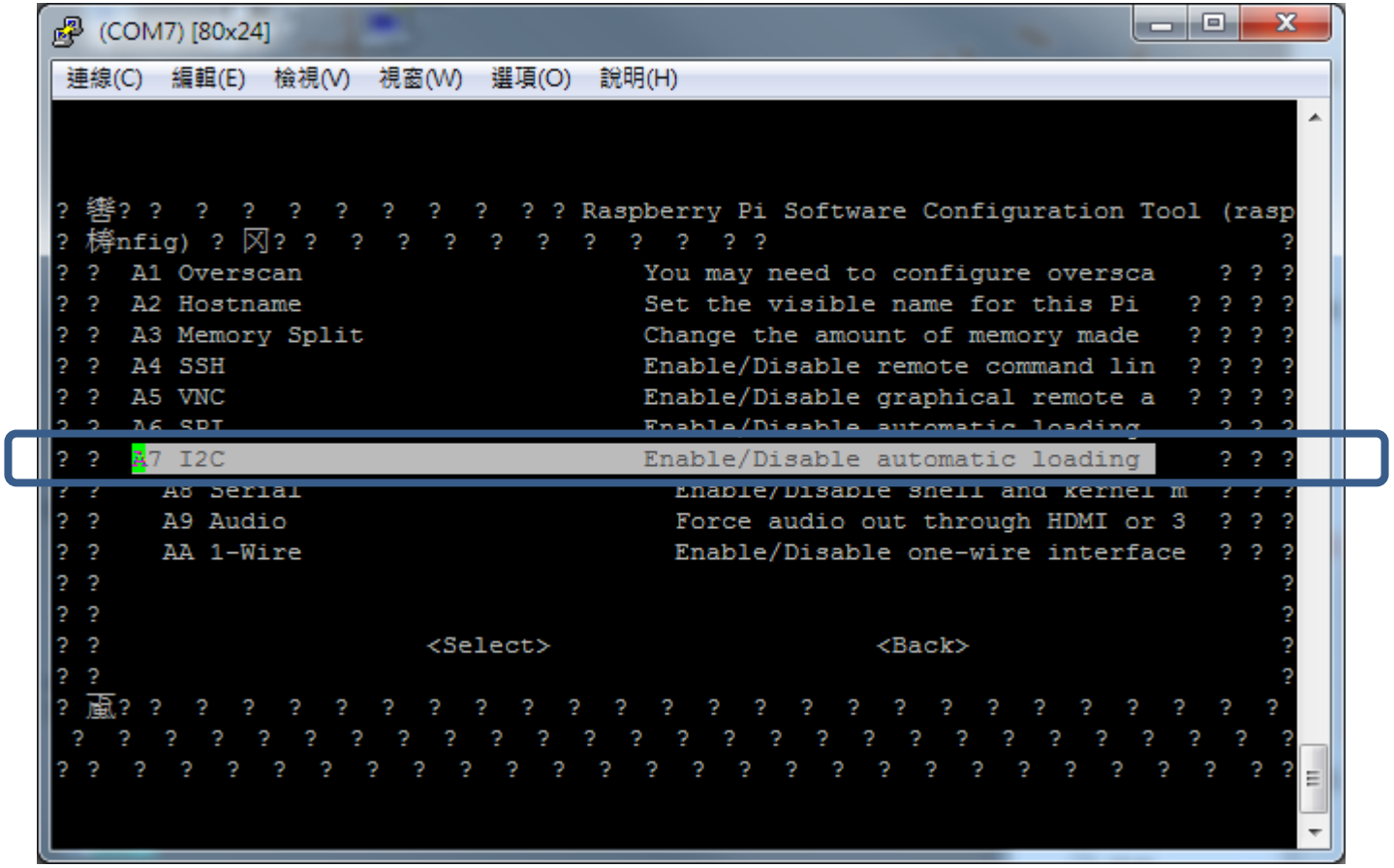
啟用 I2C功能

- 在終端機執行：sudo raspi-config
  - 選擇7 Advanced Options



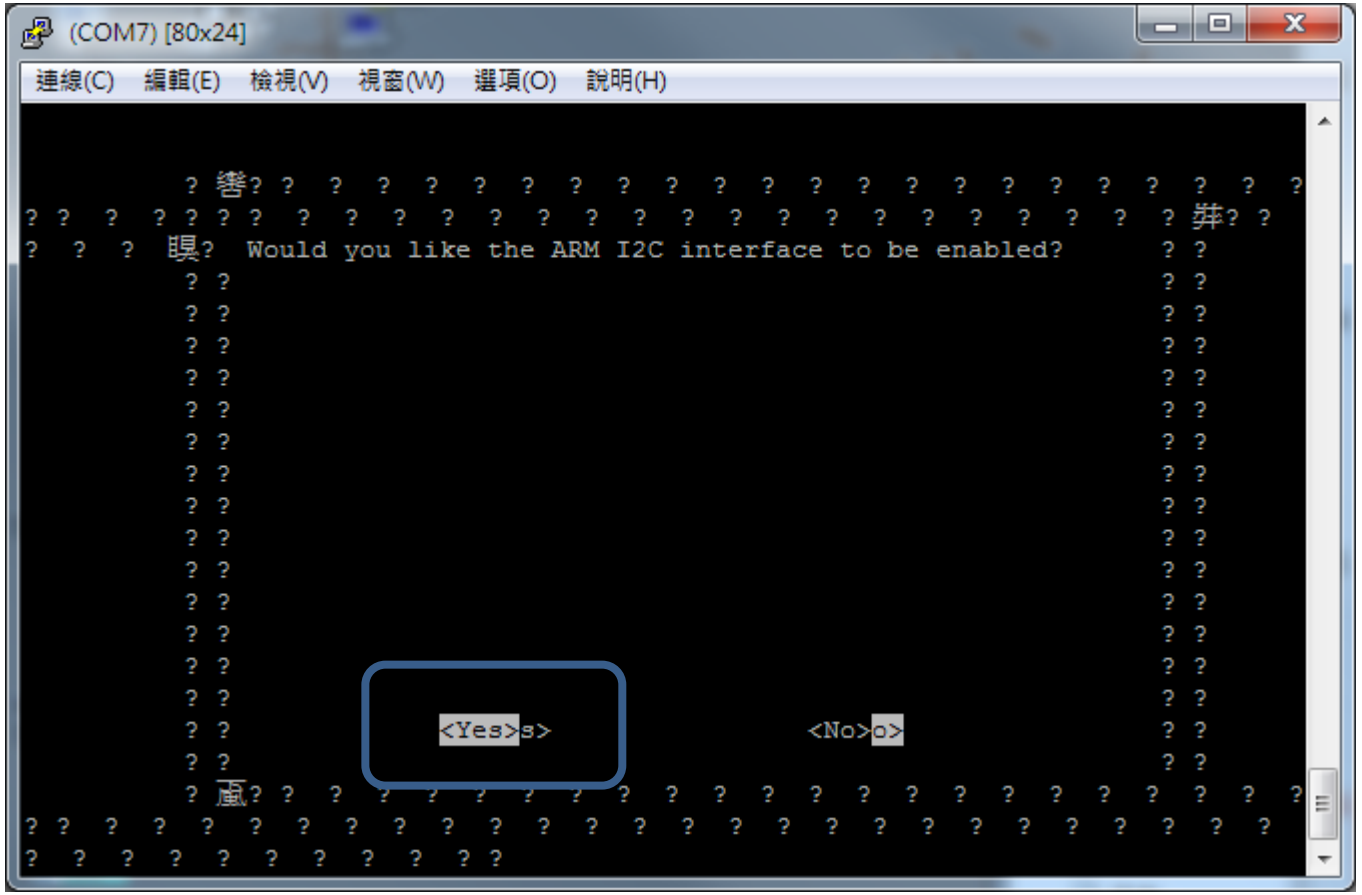
啟用 I2C

- 接著選擇A7 I2C



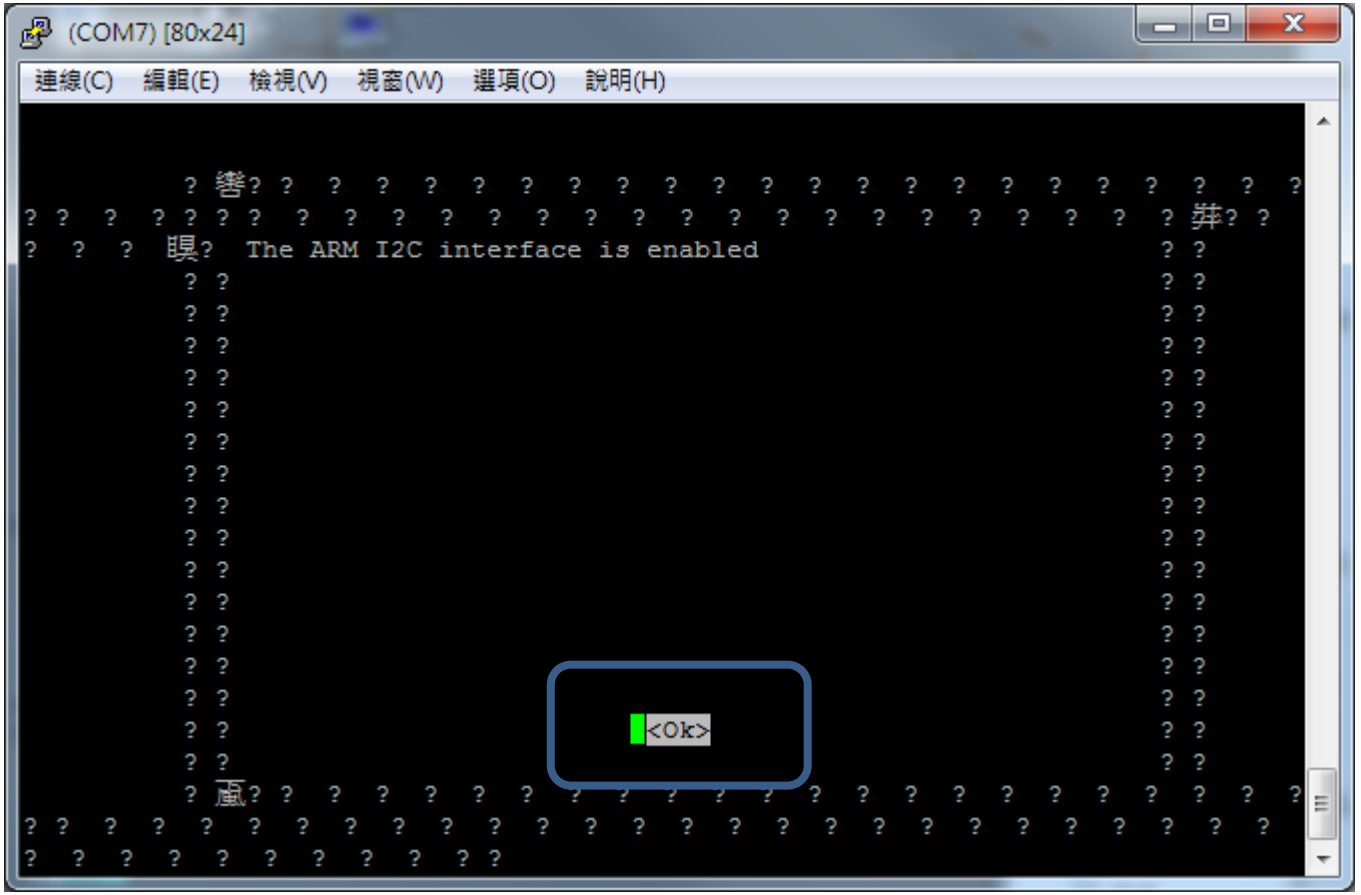
啟用 I2C

- 是否啟用I2C介面？



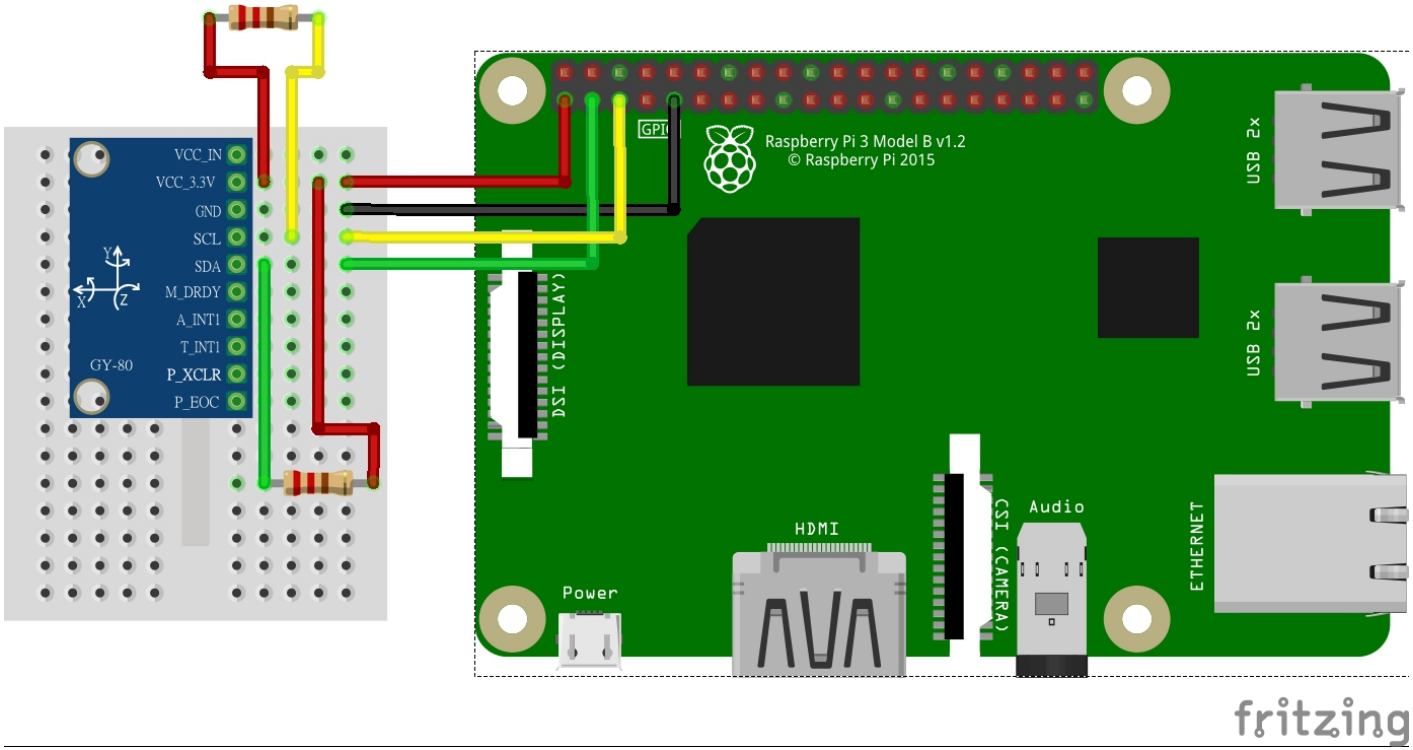
啟用 I2C

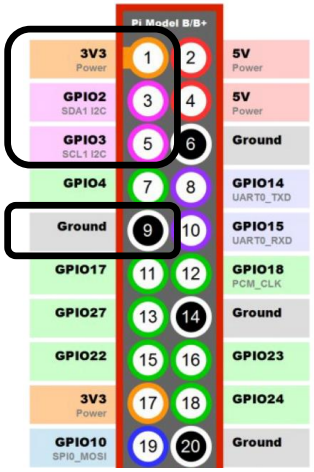
- 啟用完畢


(COM7) [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
The ARM I2C interface is enabled
<Ok>

Sensor連接線路圖

VCC	**Pin 1 (3.3V), 紅色**
GND	**Pin 9 (Ground), 黑色**
SCL	**Pin 5 (SCL), 黃色**
SDA	**Pin 3 (SDA), 綠色**





請記得加兩個2.2k的電阻在SDA/SDL上

測試 I2C

安裝I2C工具： **sudo apt-get install i2c-tools**

有連接i2c裝置

```
pi@raspberrypi:~$ sudo ls -al /dev/*i2c*
crw-rw---- 1 root i2c 89, 1 Mar 7 03:39 /dev/i2c-1
```

沒有連接i2c裝置

```
pi@raspberrypi:~$ ls -al /dev/*i2c*
ls: cannot access /dev/*i2c*: No such file or directory
```

pi@raspberrypi:~$ **sudo i2cdetect -y 1**

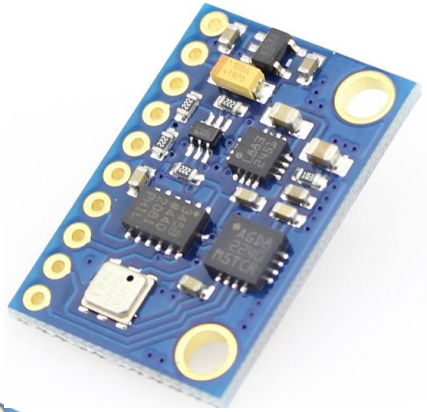
```
pi@raspberrypi:~$ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- 53 -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- 69 -- -- -- -- -- --
70: -- -- -- -- -- -- -- 77
```

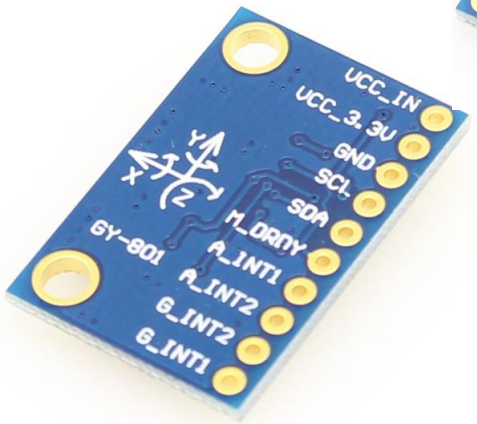
pi@raspberrypi:~$ **sudo i2cdetect -y 1**

```
pi@raspberrypi:~$ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

GY-801九軸感測器

- GY-801九軸感測器
  - 三軸陀螺儀/三軸加速度/三軸電子羅盤+大氣壓力感測
  - 加速度計(ADXL345) ：加速度
  - 陀螺儀(L3G4200) ：角速度
  - 電子羅盤(HMC5883) ：方位
  - 氣壓計(BMP085)：高度
- 通訊介面：I2C





http://wukcsoft.blogspot.tw/2014/06/read-gy-80.html

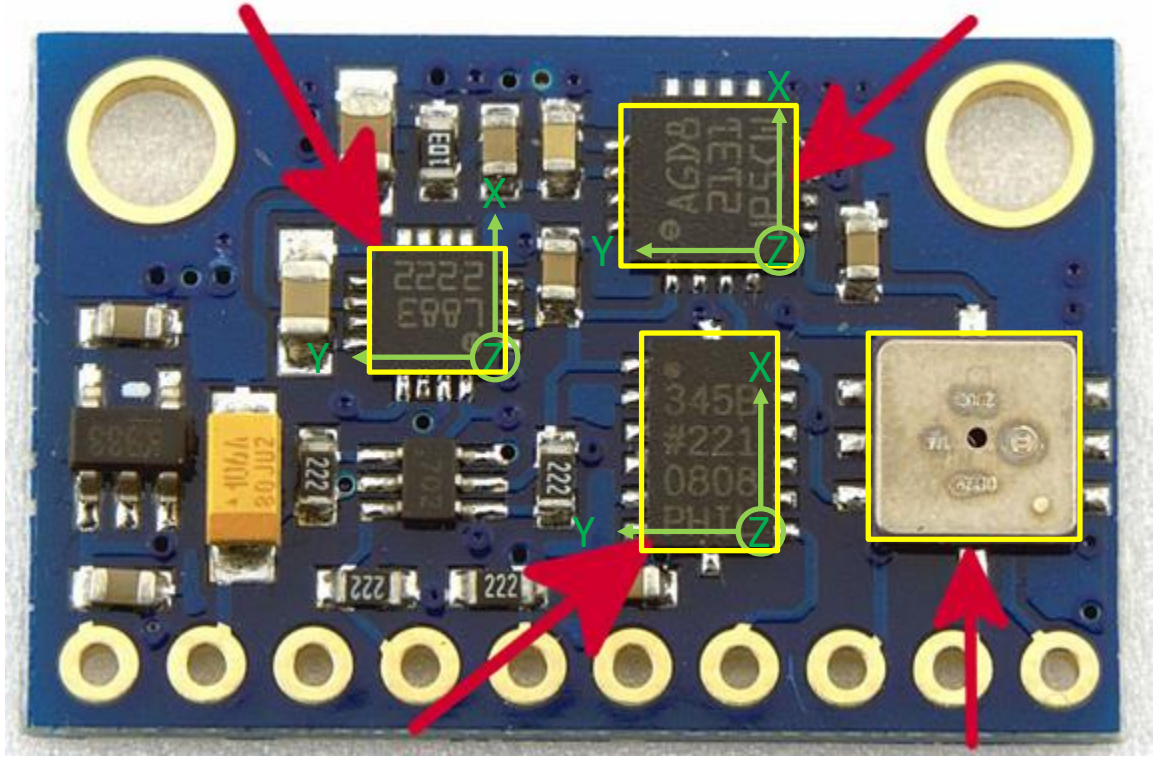
GY-801九軸感測器

- 今天的範例程式來源
  - 加速度計(ADXL345) ：加速度
    - https://github.com/adafruit/Adafruit_Python_ADXL345.git
  - 陀螺儀(L3G4200) ：角速度
    - https://github.com/bashardawood/L3G4200D-Python
  - 電子羅盤(HMC5883) ：方位
    - https://goo.gl/c6iD08
  - 氣壓計(BMP085)：高度 (下堂課)
    - https://github.com/adafruit/Adafruit_Python_BMP

GY-801九軸感測器

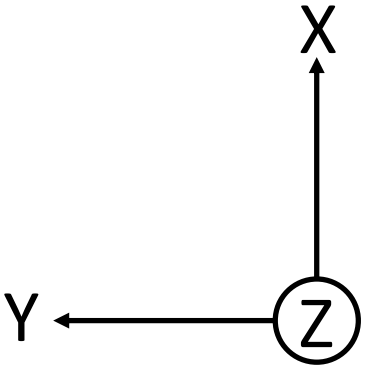
磁力計(HMC5883L)

陀螺儀(L3G4200D)



加速度計(ADXL345)

氣壓計(BMP085)



常見名詞解釋

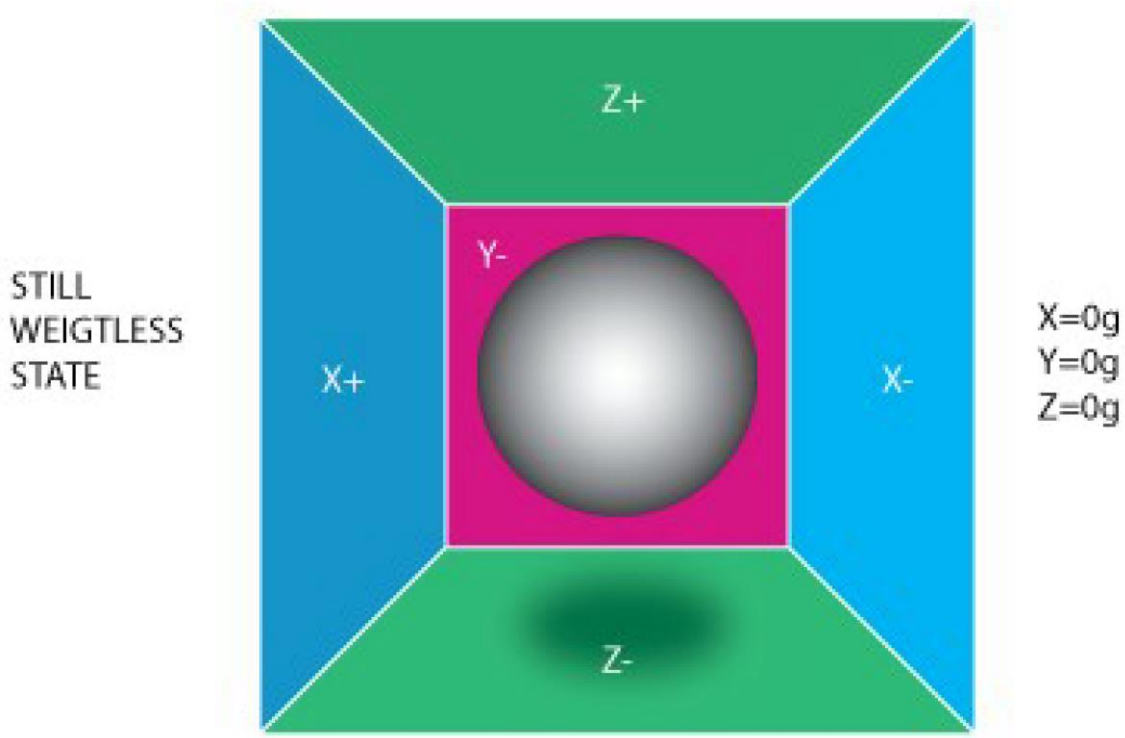
- 慣性測量單元 (IMU): Inertial Measurement Unit
- 自由度 (DOF), Degrees of Freedom
- 加速度感測器 (Accelerometer), G-sensor
- 陀螺儀 (Gyroscope)
- 電子羅盤 (magnetometer)

1. 加速度計

- 加速度計
  - 量測感受到相對於自由落體的加速度(單位 : m/s^2)
  - 可分為單、多軸加速度計
  - 包含 運動加速度 與 重力加速度
  - 慣性加速度減去當地重力加速度
    - 自由落體所感測到的加速度為0
  - 通常會搭配陀螺儀來應用
    - 因為重力加速度並不會隨時間快速變化 , 長時間具有可靠性 , 所以通常透過此特性來校正陀螺儀 , 防止誤差的累積

1. 加速度計

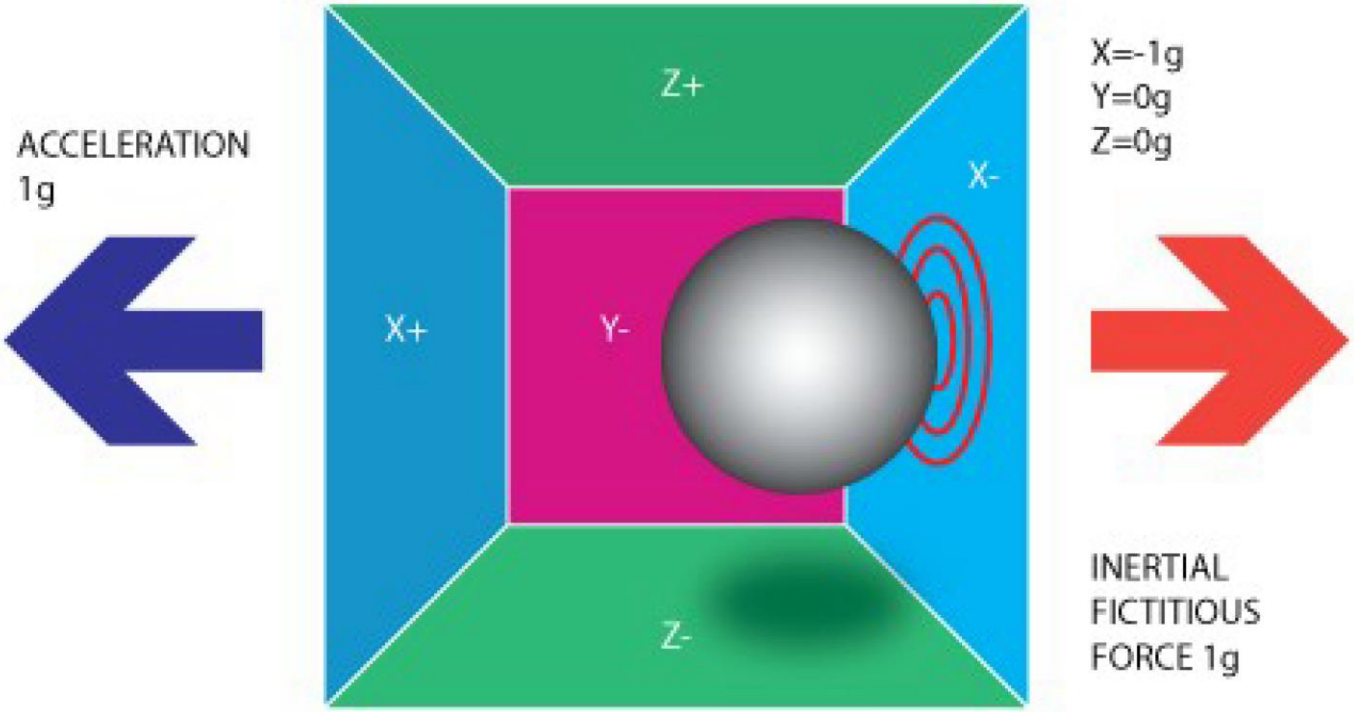
- 一個立方體盒子放著一顆球
- 盒子運動時，可根據球的位置算出加速度的值



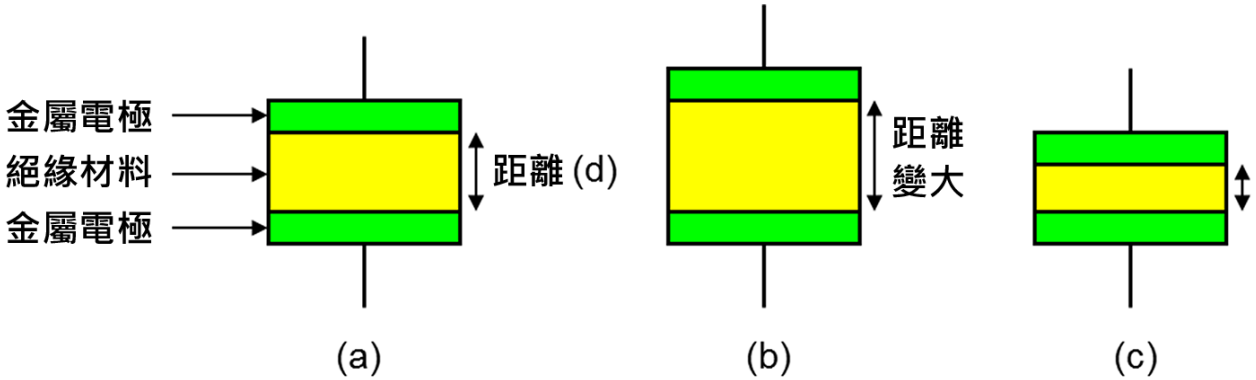
https://zhuanlan.zhihu.com/p/20082486

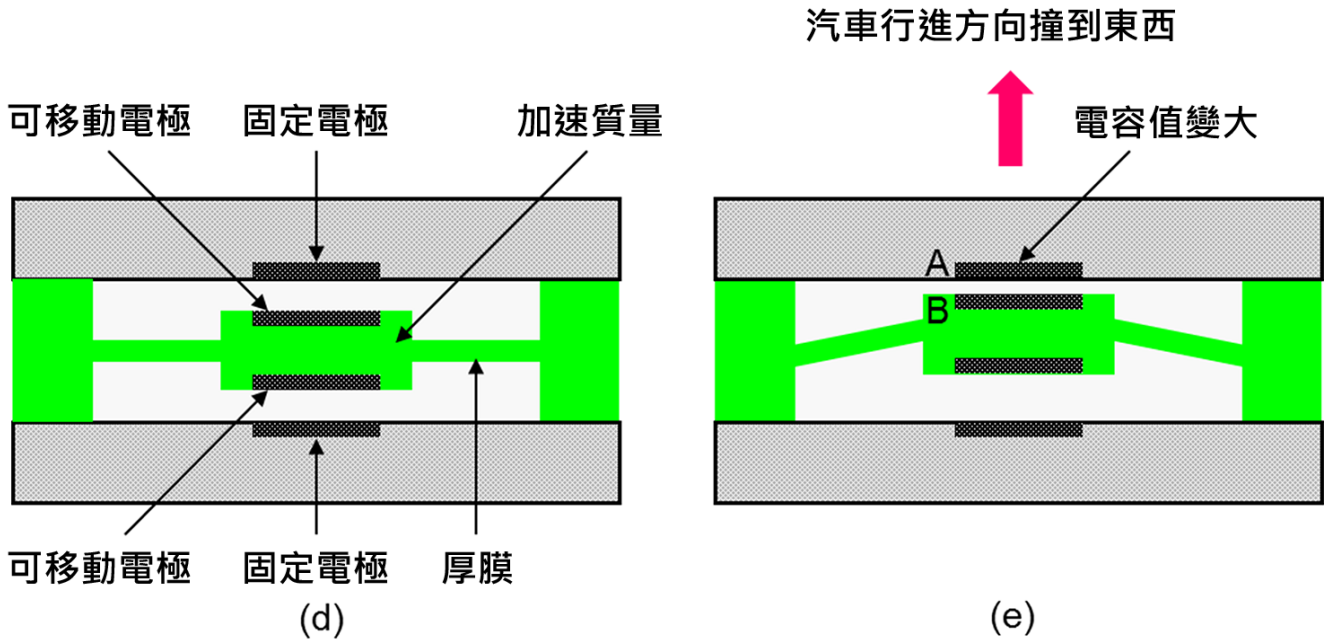
1. 加速度計

- 水平向左施力
  - 球會因為慣性貼向盒內右側



1. 加速度計





電容微加速度器的構造與原理

http://scimonth.blogspot.tw/2014/07/blog-post_94.html

1. 加速度計(ADXL345)

另一種常見的加速度計叫 MPU6050

- ADXL345加速度計
  - 控制晶片：ADXL345晶片
  - 供電電源：3 ~ 5V
  - 通訊方式：I2C/SPI 通信協議
  - 測量範圍：可達±16g、分辨率：13位 (4 mg/LSB)，能夠測量約0.25°的傾角變化
  - 數位輸出：16 位二進制補碼格式
  - 3軸：± 2g、± 4g、± 8g、± 16g

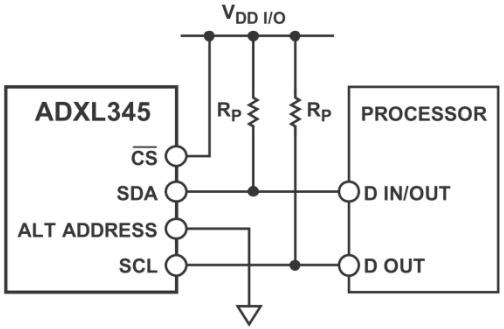
Parameter	Test Conditions	Min	Typ[1]	Max	Unit
OUTPUT RESOLUTION	Each axis				
All *g* Ranges	10-bit resolution		10		Bits
±2 *g* Range	Full resolution		10		Bits
±4 *g* Range	Full resolution		11		Bits
±8 *g* Range	Full resolution		12		Bits
±16 *g* Range	Full resolution		13		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	230	256	282	LSB/*g*
	±2 *g*, 10-bit resolution	230	256	282	LSB/*g*
	±4 *g*, 10-bit resolution	115	128	141	LSB/*g*
	±8 *g*, 10-bit resolution	57	64	71	LSB/*g*
	±16 *g*, 10-bit resolution	29	32	35	LSB/*g*
Sensitivity Deviation from Ideal	All *g*-ranges		±1.0		%
Scale Factor at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	3.5	3.9	4.3	m*g*/LSB
	±2 *g*, 10-bit resolution	3.5	3.9	4.3	m*g*/LSB
	±4 *g*, 10-bit resolution	7.1	7.8	8.7	m*g*/LSB
	±8 *g*, 10-bit resolution	14.1	15.6	17.5	m*g*/LSB
	±16 *g*, 10-bit resolution	28.6	31.2	34.5	m*g*/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C

http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf
http://www.analog.com/media/cn/technical-documentation/data-sheets/ADXL345_cn.pdf

1. 加速度計(ADXL345)

1. 加速度計

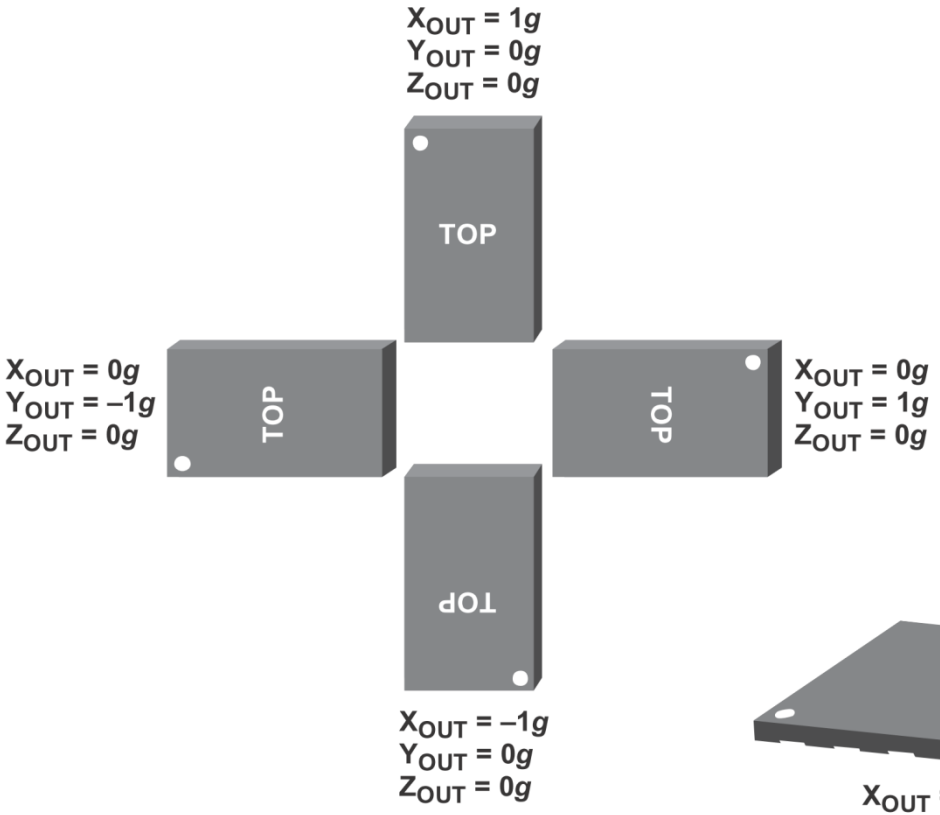
ADXL345 非常適合移動設備應用。它可以在傾斜檢測應用中測量靜態重力加速度，還可以測量運動或衝擊導致的動態加速度。它具有高分辨率（4 mg/LSB），能夠測量約 0.25°的傾角變化。使用 ADXL345 等數字輸出加速度計時，無需進行模數轉換，從而可以節省系統成本和電路板面積。此外，ADXL345是一款小而薄的超低功耗3軸加速度計，分辨率高(13位)，測量範圍達± 16g。數字輸出數據為16位二進制補碼格式，可通過SPI(3線或4線)或I2C數字接口訪問。ADXL345非常適合移動設備應用。它可以在傾斜檢測應用中測量靜態重力加速度，還可以測量運動或衝擊導致的動態加速度。其高分辨率(3.9mg/LSB)，能夠測量不到1.0°的傾斜角度變化。

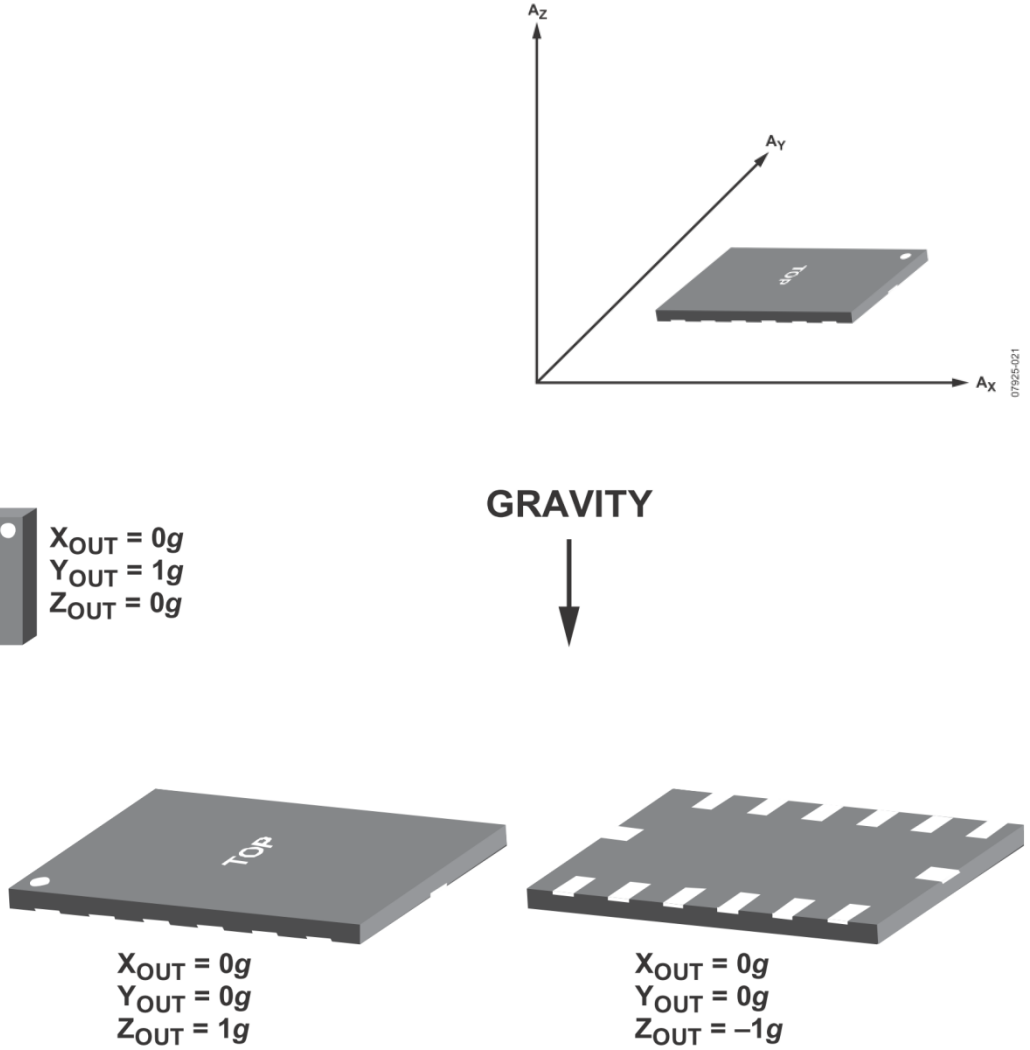


https://www.taiwaniot.com.tw/shop/module-sensor/運動及位置感應器/gy-291-adxl345-數字三軸重力加速度傾斜度模組-iicspi傳輸/

1. 加速度計(ADXL345)

座標系定義





1. 加速度計(ADXL345)

- 下載安裝程式

```
sudo apt-get install git build-essential python-dev
git clone https://github.com/adafruit/Adafruit_Python_ADXL345.git
cd Adafruit_Python_ADXL345
sudo python setup.py install

cd examples/
sudo python simplectest.py
```

```
pi@raspberrypi:~/gy80/1.acc/examples$ sudo python simpletest.py
Printing X, Y, Z axis values, press Ctrl-C to quit...
X=-13, Y=-2, Z=-272
X=-12, Y=-3, Z=-272
X=-12, Y=-2, Z=-272
X=-11, Y=-3, Z=-271
X=-12, Y=-3, Z=-270
X=-12, Y=-1, Z=-272
```

1. 加速度計(ADXL345)

- 讀取加速度資訊

```
import time
import Adafruit_ADXL345
accel = Adafruit_ADXL345.ADXL345()

print('Printing X, Y, Z axis values, press Ctrl-C to quit...')
while True:
  # Read the X, Y, Z axis acceleration values and print them.
  x, y, z = accel.read()
  print('X={0}, Y={1}, Z={2}'.format(x, y, z))
  # Wait half a second and repeat.
  time.sleep(0.5)
```

1. 加速度計(ADXL345)

- Adafruit_ADXL345/ADXL345.py

```
ADXL345_ADDRESS     = 0x53
ADXL345_REG_DEVID    = 0x00 # Device ID
ADXL345_REG_DATAX0    = 0x32 # X-axis data 0 (6 bytes for X/Y/Z)
ADXL345_REG_POWER_CTL  = 0x2D # Power-saving features control
...
```

```
def read(self):
    """Read the current value of the accelerometer and return it as a tuple
    of signed 16-bit X, Y, Z axis values.
    """
    raw = self._device.readList(ADXL345_REG_DATAX0, 6)
    return struct.unpack('<hhh', raw)
```

從ADXL345_REG_DATAX0暫存器位置讀取6byte出來, 作為x/y/z的數值
(參考datasheet的暫存器位址, 即0x??)

http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf

1. 加速度計(ADXL345)

- 測量範圍是+/-2G (default, range=2), 可是讀到的數字是

```
pi@raspberrypi:~/gy80/1.acc/examples$ sudo python simpletest.py
Printing X, Y, Z axis values, press Ctrl-C to quit...
X=-13, Y=-2, Z=-272
X=-12, Y=-3, Z=-272
X=-12, Y=-2, Z=-272
```

- -> 使用下面公式轉換回+/-2G的範圍：

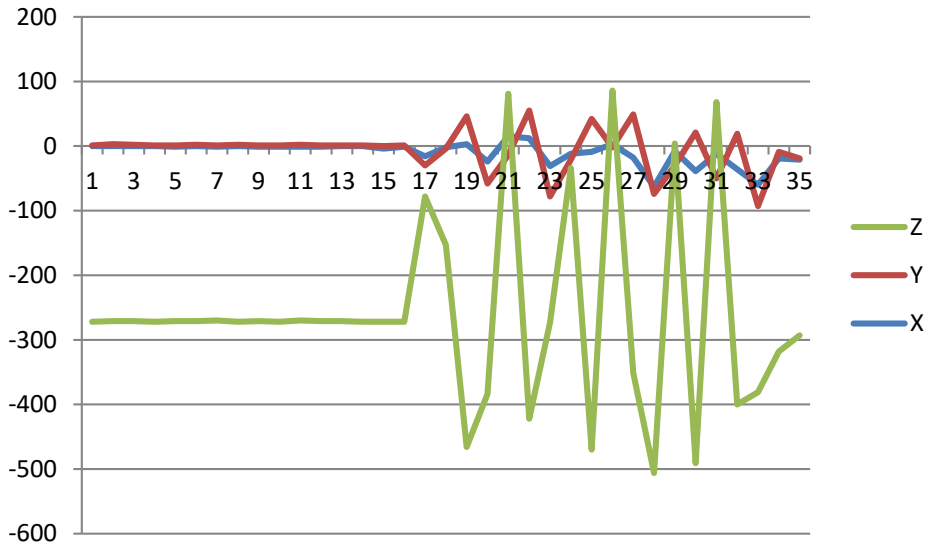
$$G_{Accel} = Raw_{Accel} * \frac{Range}{2^{Resolution-1}} \text{ or } Raw_{Accel}/\text{Sensitivity}$$

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
OUTPUT RESOLUTION	Each axis				
All *g* Ranges	10-bit resolution		10		Bits
±2 *g* Range	Full resolution		10		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	230	256	282	LSB/*g*
	±2 *g*, 10-bit resolution	230	256	282	LSB/*g*

https://theccontinuum.com/2012/09/24/arduino-imu-pitch-roll-from-accelerometer/

1. 加速度計(ADXL345)

- Quiz1-1: 將程式輸出結果改為+/-2G的範圍
  - sudo python simpletest.py > log.txt
  - 上面指令可以把程式執行結果存到log.txt裡面
    終端機不會有訊息, 需按ctrl+c終止
- Quiz1-2: 紀錄加速度的數值,畫出靜止狀態與搖晃狀態的圖
  - 指令: cat log.txt　　可以在終端機顯示檔案內容, 再用滑鼠複製貼上



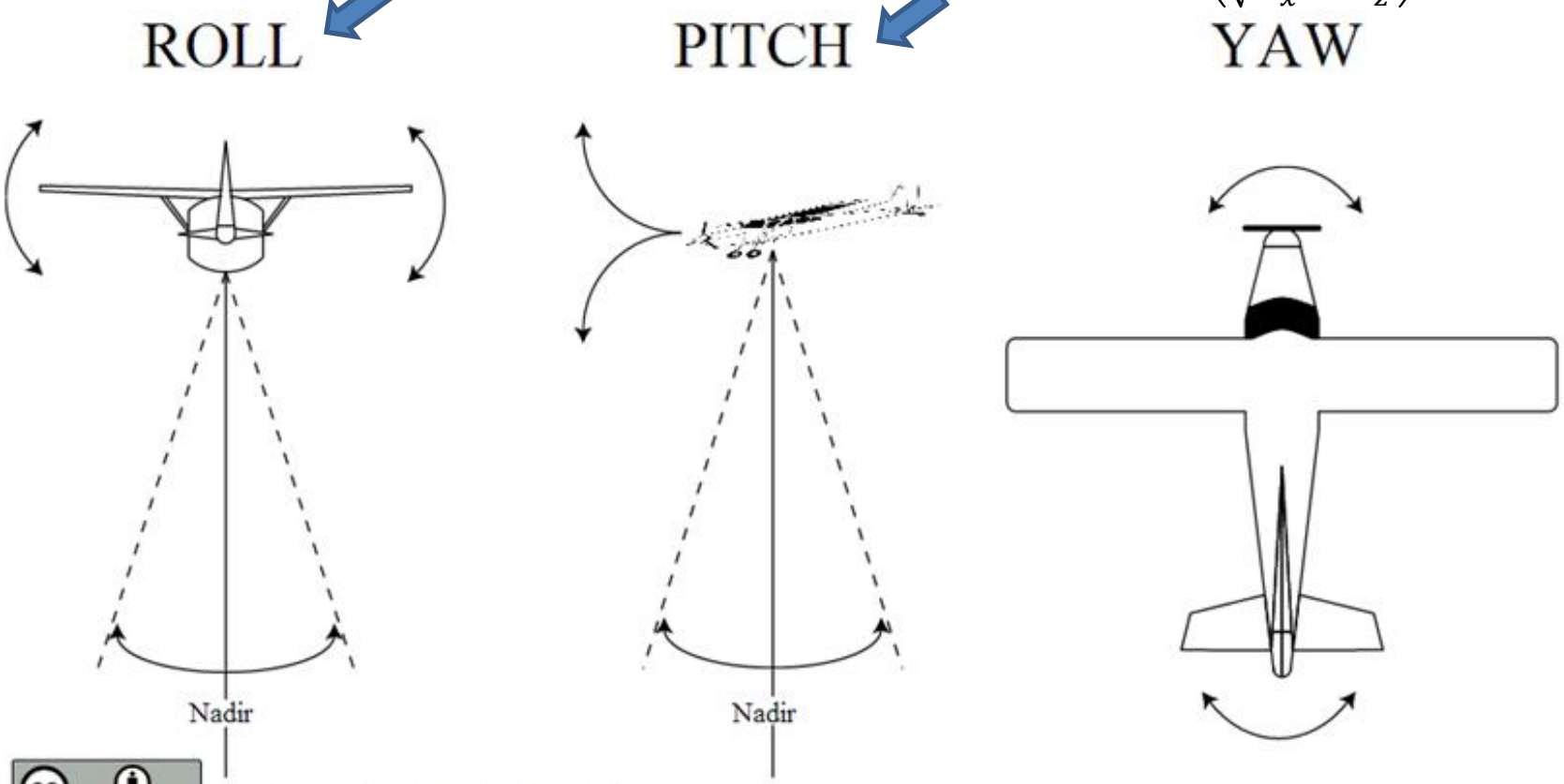
姿態分析：測量角度，基於 加速度計算角度, 然後再加上 陀螺儀 與 前一回的角度資訊 進行資料融合

加速度計的應用

- 姿態分析

$$roll = arctan\left(\frac{-G_x}{G_z}\right)$$

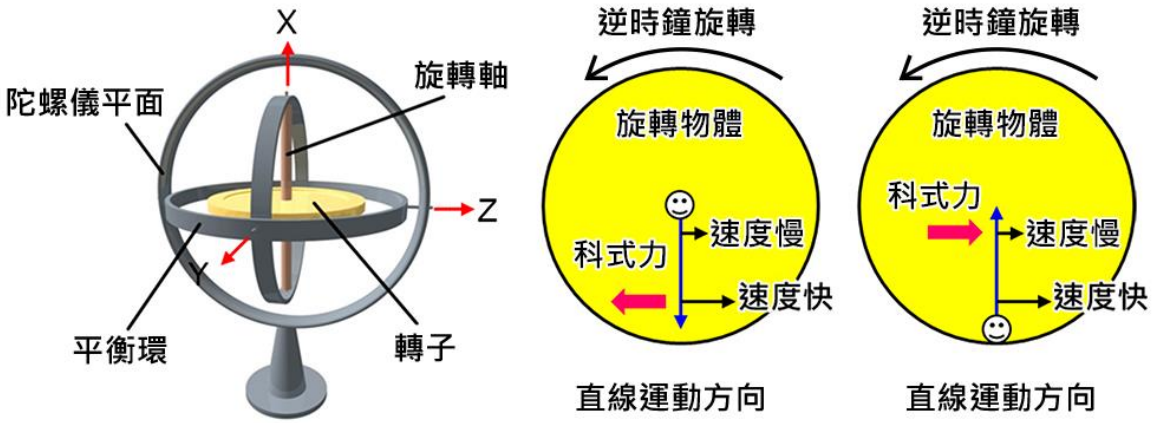
$$pitch = arctan\left(\frac{G_y}{\sqrt{G_x^2 + G_z^2}}\right)$$



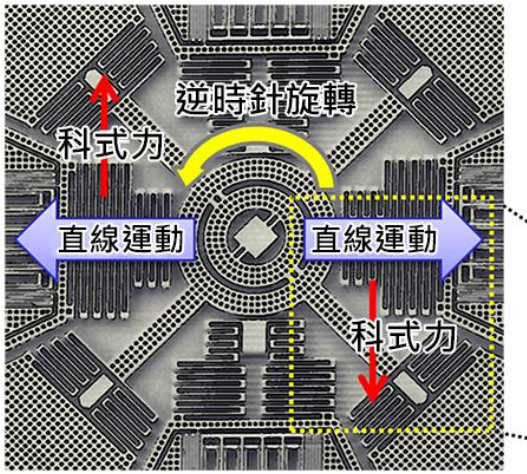
This work is licensed under a Creative Commons Attribution 3.0 Unported License.
Author: http://commons.wikimedia.org/wiki/User:Arbeck

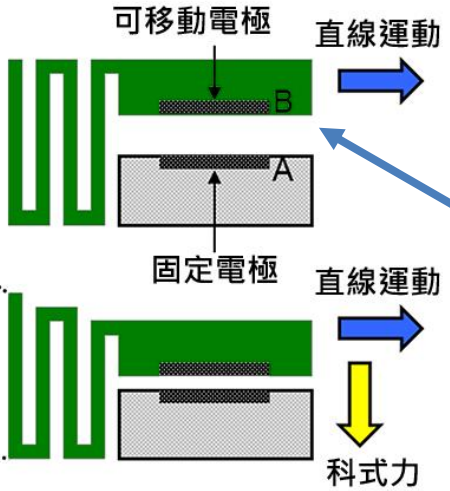
http://bluelemonlabs.blogspot.tw/2013/08/arduino-imu-pitch-roll-from-adxl345.html

2. 陀螺儀



(a) (b) (c)





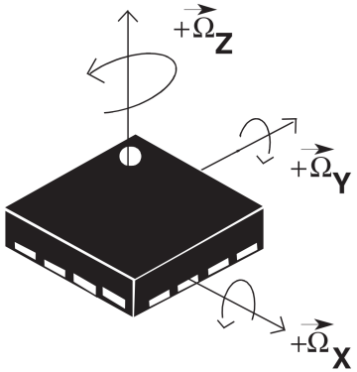
轉動會影響電極距離
(科式力)

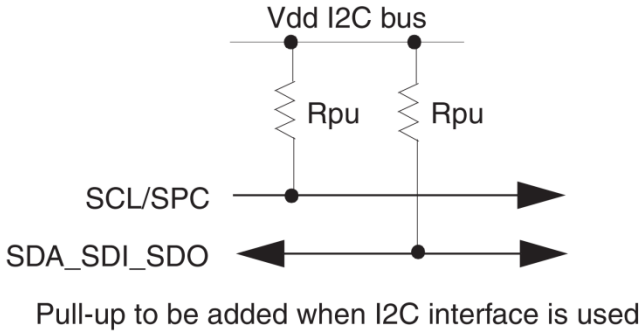
(d) 微陀螺儀的構造與原理 (e)

http://scimonth.blogspot.tw/2014/07/blog-post_94.html

2. 陀螺儀 (L3G4200D)

- L3G4200D陀螺儀
  - 可選擇三種解析範圍(250/500/2000 dps)
  - I2C/SPI數位輸出介面
  - 16 bit-rate 數值資料輸出
  - 8-bit 溫度資料輸出
  - 具有相當大範圍的電壓輸入: 2.4 V to 3.6 V
  - 相容於低電壓的IO (1.8 V)
  - 內嵌 power-down 及 sleep 模式
  - 內嵌溫度感測器
  - 高抗撞擊能力





- 具有多種解析度可選擇 ±250/±500/±2000 dps 且可針對使用者選擇的頻寬來量測角速度率，它在一些遊戲互動、虛擬實境、 GPS系統或機器人領域種當作訊號輸入裝置具有相當不錯的表現。

http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Gyros/3-Axis/17116.pdf
http://www.playrobot.com/gyro/284-sparkfun-tri-axis-gyro-breakout-l3g4200d.html

2. 陀螺儀 (L3G4200D)

- 下載安裝程式

```
cd ~
git clone https://github.com/bashardawood/L3G4200D-Python
cd L3G4200D-Python
sudo python gyro.py
```

- 執行結果

```
pi@raspberrypi:~/L3G4200D-Python$ sudo python gyro.py
     -1013        -256        1602
       946        -683        -195
       943        -681        -198
       947        -683        -193
       940        -679        -194
       947        -672        -192
       935        -680        -193
       948        -514        -192
```

2. 陀螺儀 (L3G4200D)

```
#!/usr/bin/python

from time import sleep
import smbus
import string

def getSignedNumber(number):
  if number & (1 << 15):
    return number | ~65535
  else:
    return number & 65535

i2c_bus=smbus.SMBus(1)
i2c_address=0x69
i2c_bus.write_byte_data(i2c_address,0x20,0x0F)
i2c_bus.write_byte_data(i2c_address,0x23,0x20)
```

載入time的sleep (載入特定成員)

轉換回正負數值

使用二的補數來表示正負號

- 正號 : 0x0000 ~ 0x7FFF (0 ~ 32767)
- 負號 : 0x8000 ~ 0xFFFF (32768 ~ 65535)

設定暫存器位址

2. 陀螺儀 (L3G4200D)

```
while True:
    i2c_bus.write_byte(i2c_address,0x28)
    X_L = i2c_bus.read_byte(i2c_address)
    i2c_bus.write_byte(i2c_address,0x29)
    X_H = i2c_bus.read_byte(i2c_address)
    X = X_H << 8 | X_L
    ....

    X = getSignedNumber(X)
    Y = getSignedNumber(Y)
    Z = getSignedNumber(Z)

    print string.rjust(`X`, 10),
    print string.rjust(`Y`, 10),
    print string.rjust(`Z`, 10)
    sleep(0.02)
```

讀第一個&第二個byte, 結合在一起

用二的補數轉換回來

顯示數字後, 往右補10個空白
(排版用)

2. 陀螺儀 (L3G4200D)

- 執行結果長這樣, 要如何轉換成角度的格式?

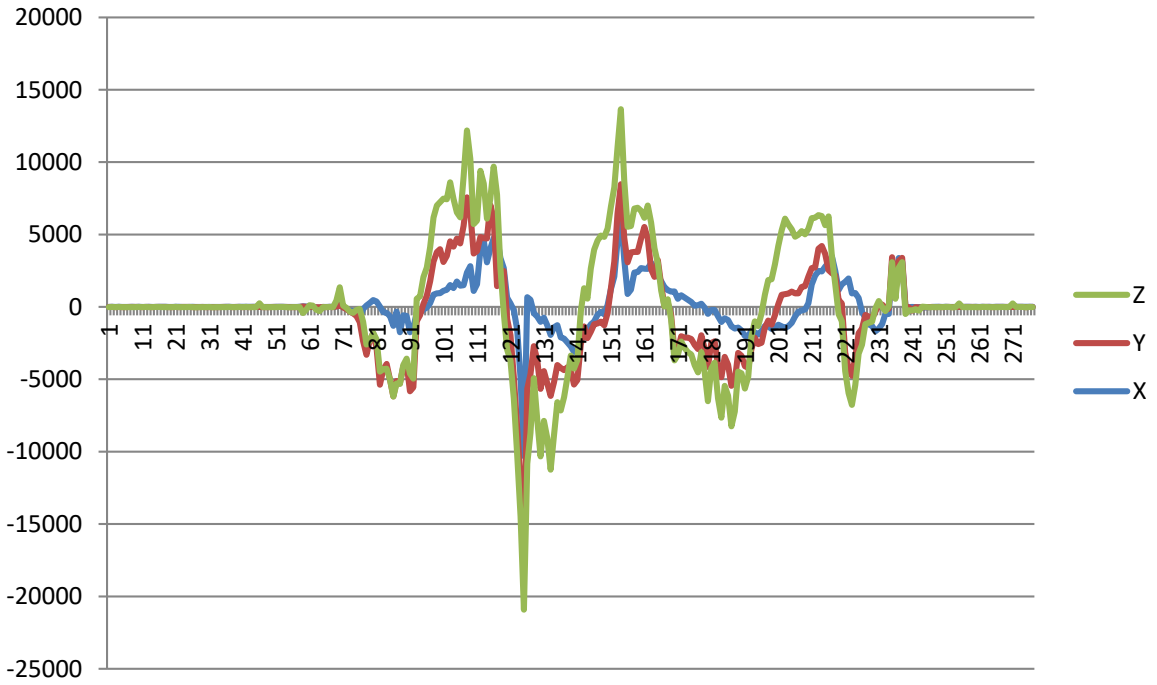
```
pi@raspberrypi:~/L3G4200D-Python$ sudo python gyro.py
     -1013       -256       1602
       946       -683       -195
       943       -681       -198
       947       -683       -193
       940       -679       -194
       947       -672       -192
       935       -680       -193
       948       -514       -192
```

- 把sensor數值轉換為degree
  - degreesPerSecond = (Raw * 8.75) / 1000

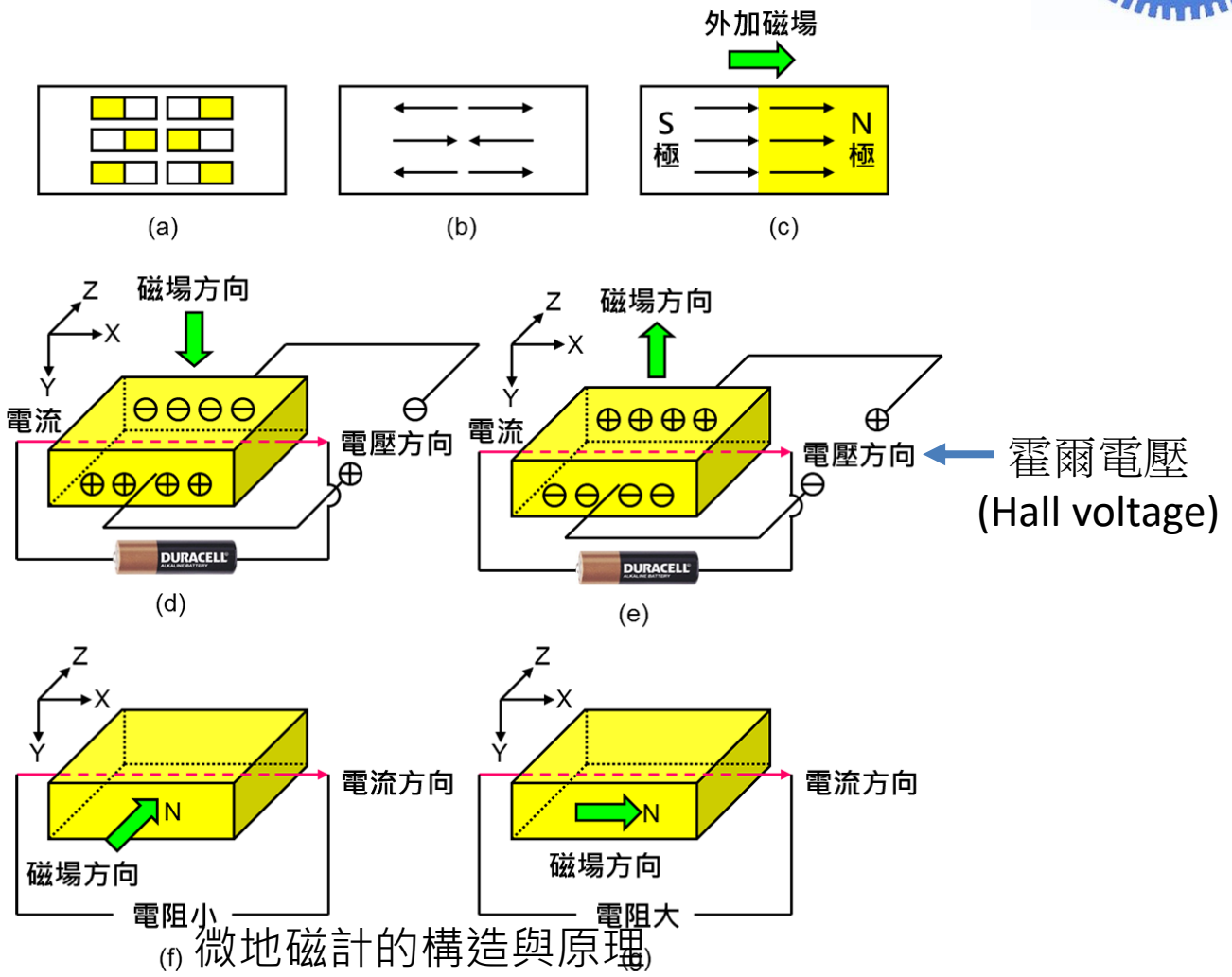
Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		

2. 陀螺儀 (L3G4200D)

- Quiz2: 紀錄陀螺儀的數值, 畫出靜止狀態與搖晃狀態的圖
  - sudo python gyro.py > log.txt
  - 上面指令可以把程式執行結果存到log.txt裡面
    終端機不會有訊息, 需按ctrl+c終止



3. 磁力計

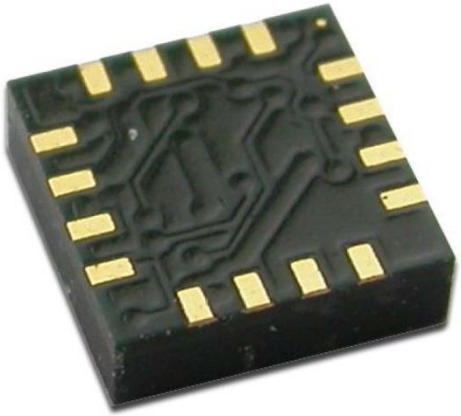


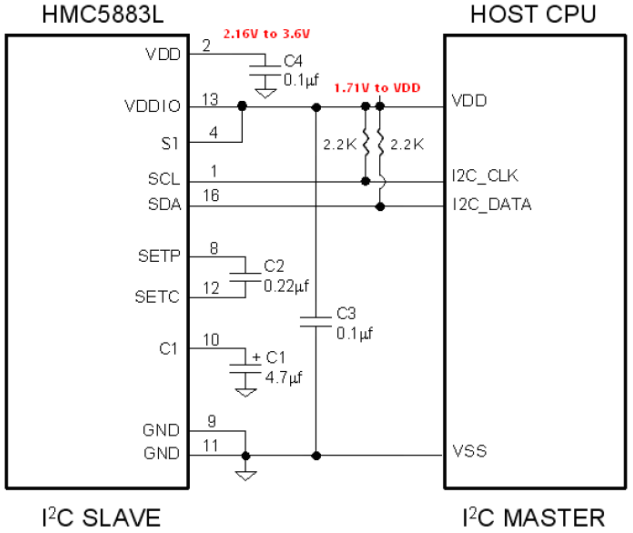
微地磁計的構造與原理

http://scimonth.blogspot.tw/2014/07/blog-post_94.html

3. 磁力計 (HMC5883L)

- HMC5883L 磁力計
  - 測量三軸的地球磁場強度，具有1~2度的精確度
  - 提供三軸個別的磁場強度值，可以作為3D的計算
  - 磁場範圍：±0.88/1.3/1.9/2.5/4.0/4.7/5.6/8.1 gauss
  - 可測量地球磁場在X,Y,Z三軸方向的原始高斯強度(gauss)，並可作為測量地球的磁場北極方向





https://sites.google.com/site/playrobotquickstart/home/hmc583l-dian-zi-luo-pan
https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf
http://web.sensor-ic.com:8000/ZLXIAZAI/20130122/HMC5883L中文规格书-Feb%202011.pdf

3. 磁力計 (HMC5883L)

- 根據datasheet設定暫存器

Register Access

This section describes the process of reading from and writing to this device. The devices uses an address pointer to indicate which register location is to be read from or written to. These pointer locations are sent from the master to this slave device and succeed the 7-bit address (0x1E) plus 1 bit read/write identifier, i.e. 0x3D for read and 0x3C for write.

```
bus = smbus.SMBus(1)
addrHMC = 0x1e

def init_imu():
  # HMC setting
  bus.write_byte_data(addrHMC, 0, 0b01110000) # Set to 8 samples @ 15Hz
  bus.write_byte_data(addrHMC, 1, 0b00100000) # 1.3 gain LSb / Gauss 1090 (default)
  bus.write_byte_data(addrHMC, 2, 0b00000000) # Continuous sampling
```

3. 磁力計 (HMC5883L)

bus.write_byte_data(addrHMC, 0, 0b01110000
bus.write_byte_data(addrHMC, 1, 0b00100000
bus.write_byte_data(addrHMC, 2, 0b00000000)

Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read ➡ read_word_2c(addrHMC, 3)
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read ➡ read_word_2c(addrHMC, 5)
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read ➡ read_word_2c(addrHMC, 7)
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

Table2: Register List

3. 磁力計 (HMC5883L)

bus.write_byte_data(addrHMC, 1, 0b00100000) # 1.3 gain LSb / Gauss 1090 (default)

- Configuration Register B

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

Table 7: Configuration B Register

GN2	GN1	GN0	Recommended Sensor Field Range	Gain (LSb/ Gauss)	Digital Resolution (mG/LSb)	Output Range
0	0	0	± 0.88 Ga	1370	0.73	0xF800–0x07FF (-2048–2047)
0	0	1	± 1.3 Ga	1090 (default)	0.92	0xF800–0x07FF (-2048–2047)

+/- 1.3Ga; Gain 1090

3. 磁力計 (HMC5883L)

```
bus.write_byte_data(addrHMC, 0, 0b01110000) # Set to 8 samples @ 15Hz
```

- Configuration Register A

CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Table 3: Configuration Register A

Location	Name	Description
CRA7	CRA7	Bit CRA7 is reserved for future function. Set to 0 when configuring CRA.
CRA6 to CRA5	MA1 to MA0	Select number of samples averaged (1 to 8) per measurement output. 00 = 1(Default); 01 = 2; 10 = 4; 11 = 8

8 samples

DO2	DO1	DO0	Typical Data Output Rate (Hz)
0	0	0	0.75
0	0	1	1.5
0	1	0	3
0	1	1	7.5
1	0	0	15 (Default)
1	0	1	30
1	1	0	75
1	1	1	Reserved

Table 5: Data Output Rates

15 Hz

3. 磁力計 (HMC5883L)

```
bus.write_byte_data(addrHMC, 2, 0b00000000) # Continuous sampling
```

- Mode Register

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
HS(0)	(0)	(0)	(0)	(0)	(0)	MD1 (0)	MD0 (1)

Table 10: Mode Register

MD1	MD0	Operating Mode
0	0	Continuous-Measurement Mode. In continuous-measurement mode, the device continuously performs measurements and places the result in the data register. RDY goes high when new data is placed in all three registers. After a power-on or a write to the mode or configuration register, the first measurement set is available from all three data output registers after a period of $2/f_{DO}$ and subsequent measurements are available at a frequency of f_{DO}, where f_{DO} is the frequency of data output.
0	1	Single-Measurement Mode (Default). When single-measurement mode is selected, device performs a single measurement, sets RDY high and returned to idle mode. Mode register returns to idle mode bit values. The measurement remains in the data output register and RDY remains high until the data output register is read or another measurement is performed.

設定為Continuous-Measurement Mode

3. 磁力計 (HMC5883L)

- A common approach used to estimate the heading is to measure two orthogonal components of the magnetic vector and calculate the heading as the arctangent of their ratio:

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

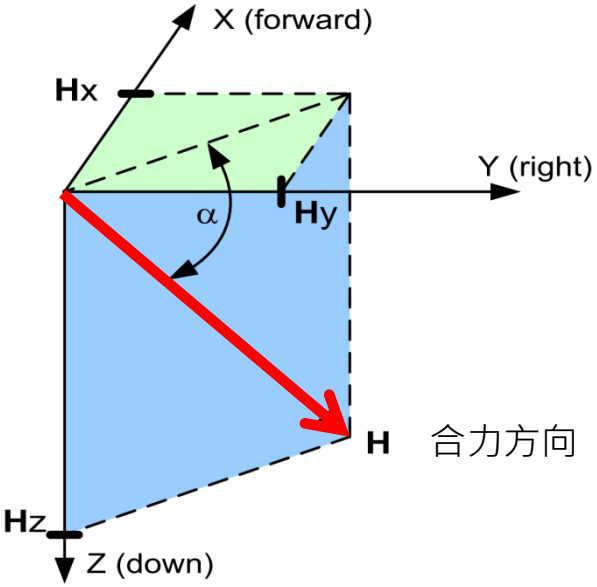


Fig. Magnetic Field Inclination Angle

http://www.cypress.com/file/130456/download

3. 磁力計 (HMC5883L)

```python
import sys, os, math, time, thread, smbus, random, requests

bus = smbus.SMBus(1)
addrHMC = 0x1e

def read_word(address, adr):
    high = bus.read_byte_data(address, adr)
    low = bus.read_byte_data(address, adr + 1)
    val = (high << 8) + low
    return val

def read_word_2c(address, adr):
    val = read_word(address, adr)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val

def main():

    bus.write_byte_data(addrHMC, 0, 0b01110000)  # Set to 8 samples @ 15Hz
    bus.write_byte_data(addrHMC, 1, 0b00100000)  # 1.3 gain LSb / Gauss 1090 (default)
    bus.write_byte_data(addrHMC, 2, 0b00000000)  # Continuous sampling

    while True:
        x = read_word_2c(addrHMC, 3)
        y = read_word_2c(addrHMC, 7)
        z = read_word_2c(addrHMC, 5)

        print x,",",y,",",z
        time.sleep(0.1)

if __name__ == "__main__":
    main()
```

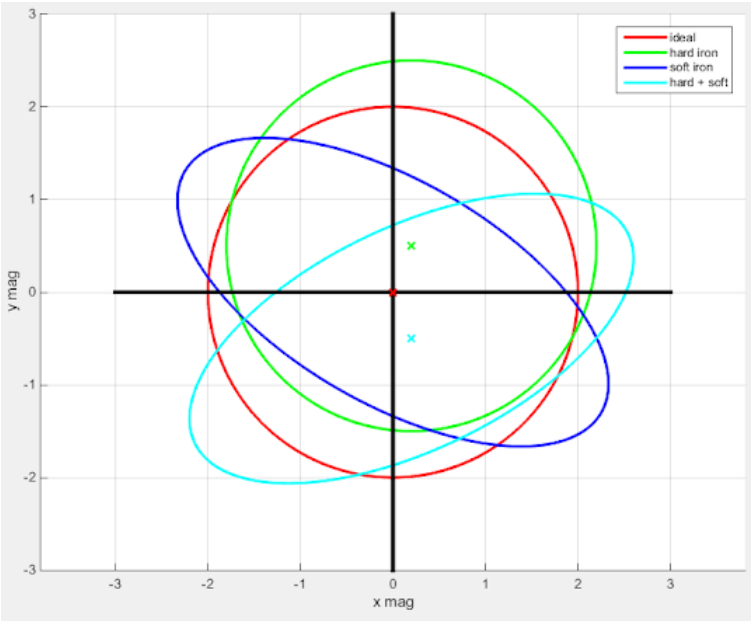
讀取x/y/z的資料

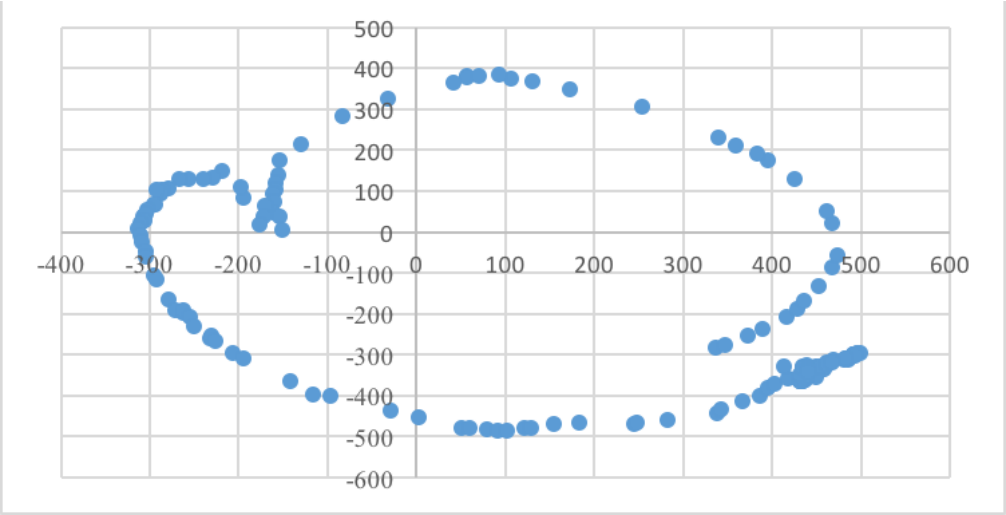
顯示讀取的x/y/z的資料

下載範例程式：**wget https://goo.gl/c6iD08 -O compass.py**

3. 磁力計 (HMC5883L)

- Quiz3: 紀錄磁力計數值, 將磁力計水平旋轉一圈紀錄xy變化
  - python compass.py > log.csv
  - 上面指令可以把程式執行結果存到log.csv裡面
    終端機不會有訊息, 需按ctrl+c終止





http://kitsprout.blogspot.com/2015/10/qcopter_7.html

3.磁力計單位

- 原始資料(raw data), 除上sensitivity的數字, 就可以得到磁場強度(gauss)
  - 磁場範圍：±0.88/1.3/1.9/2.5/4.0/4.7/5.6/8.1 gauss
  - 得到-4096時, 表示overflow

Performance

Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

Summary

- Practice Lab (加速度, 陀螺儀, 電子羅盤)

- Write code for Quiz 1 - 3, then demonstrate it to TAs
  - Quiz1-1: 將程式輸出結果改為+/-2G的範圍
  - Quiz1-2: 紀錄加速度的數值,畫出靜止狀態與搖晃狀態的圖
  - Quiz2: 紀錄陀螺儀的數值, 畫出靜止狀態與搖晃狀態的圖
  - Quiz3: 紀錄磁力計數值, 將磁力計水平旋轉一圈紀錄xy變化

Reference

- From 台灣樹莓派
  - slideshare投影片
    - http://www.slideshare.net/raspberrypi-tw/presentations
  - github程式碼
    - https://github.com/raspberrypi-tw
- Datasheet
  - 加速度
    - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf
  - 陀螺儀
    - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf
  - 電子羅盤
    - http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
  - 氣壓計
    - http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- 慣性感測器簡介 (Chih-Wei Yi)
  - https://goo.gl/BX2cor