



# 物聯網技術結合AI大數據 分析實作(1)

徐凡耘  
2020-07

# ABOUT ME

## Education

- NCU (MIS) 、NCCU (CS)

## Experience

- 外商資料倉儲、國內電信業

## Teaching

- 資策會

## User Group

- TW Spark User Group
- TW Hadoop User Group

## Consulting

- 聯瞻資訊顧問、數數科技資訊社顧問



# DAY1

## ▶ 上午

- ▶ 物聯網大數據介紹與實際案例分享
- ▶ 樹莓派3介紹與作業系統安裝



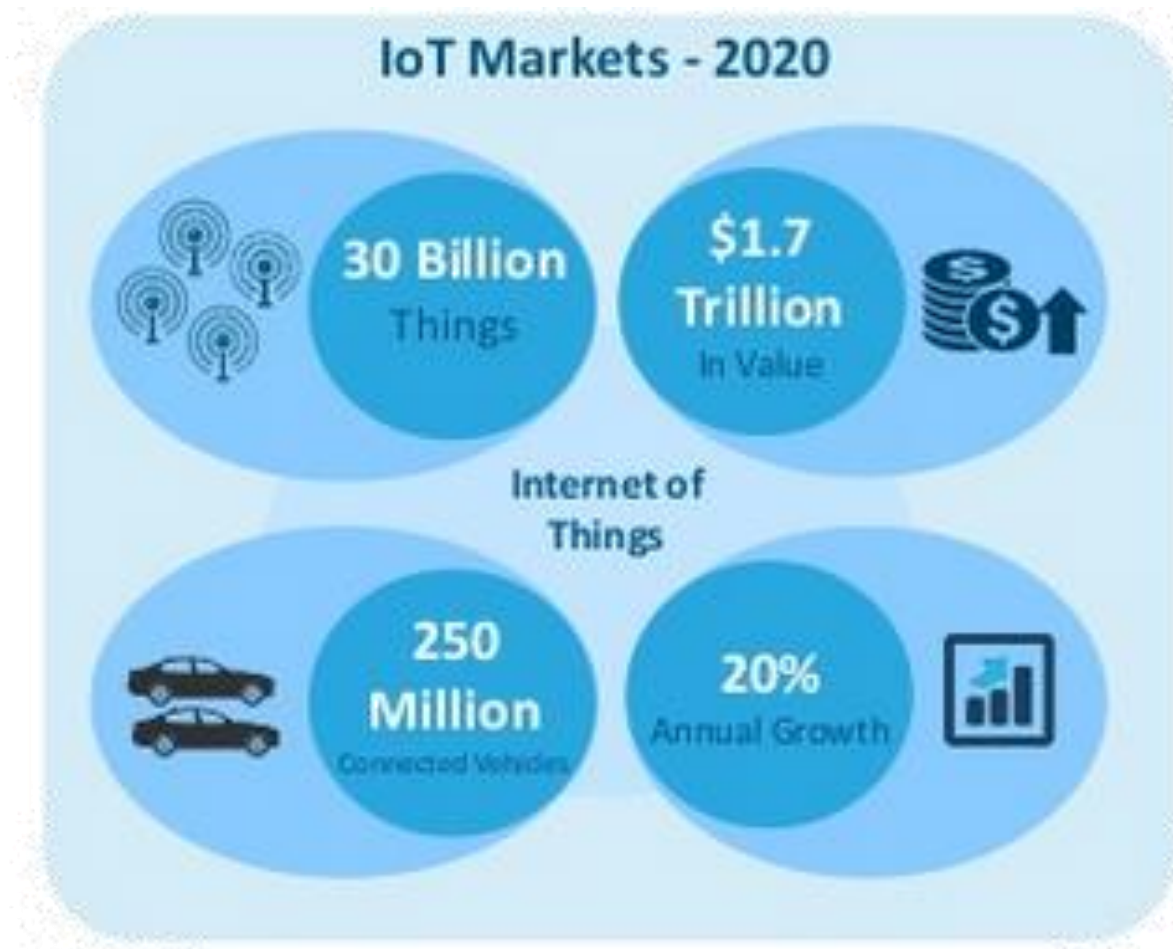
## ▶ 下午

- ▶ **Raspbian** 環境介紹與相關資源
- ▶ **Docker on 樹莓派3**
- ▶ **Docker** 指令操作
- ▶ **Docker** 操作 **GPIO**





# 認識物聯網

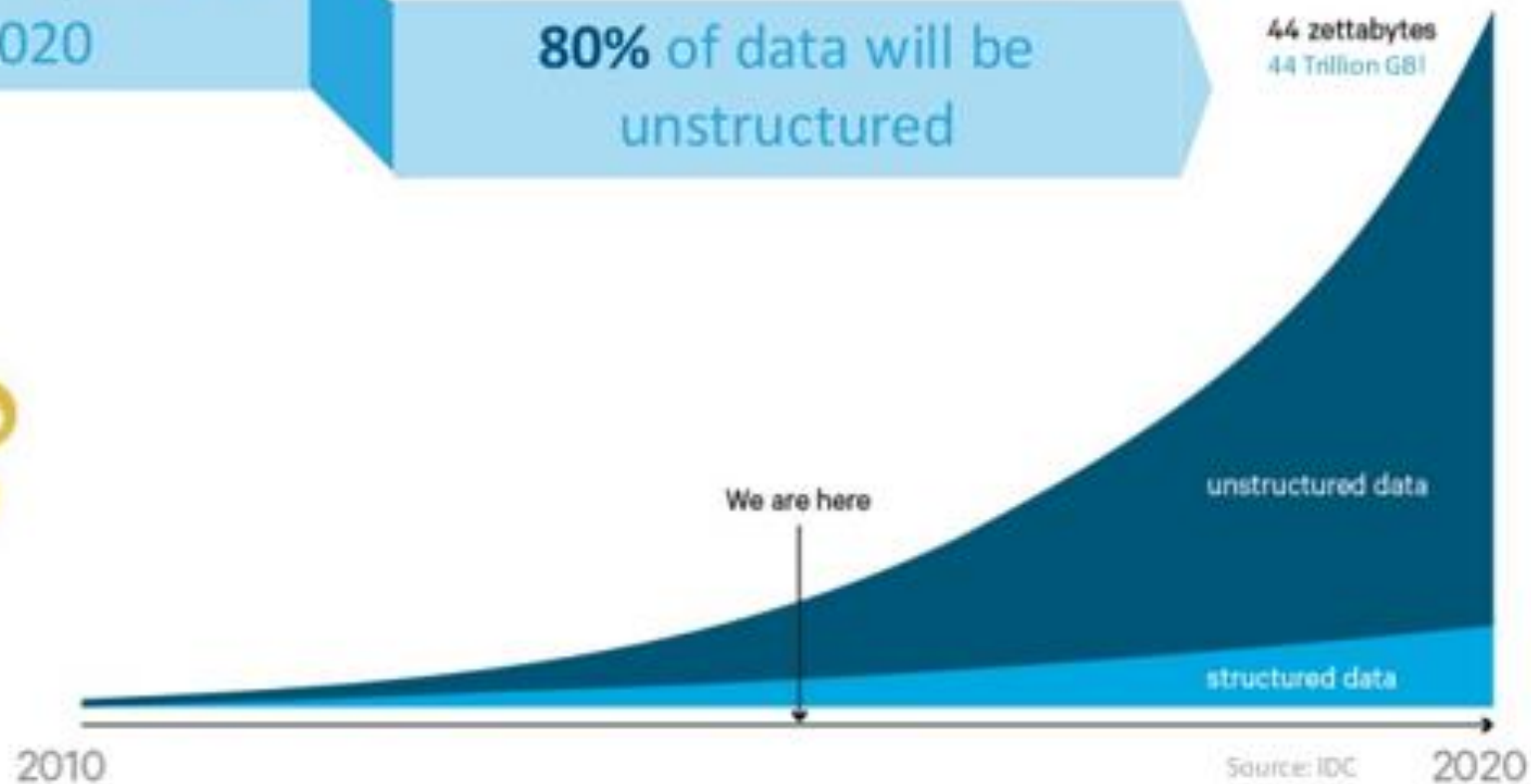


# 認識物聯網

Data expected to explode to  
**44 ZB** by 2020

**80%** of data will be  
unstructured

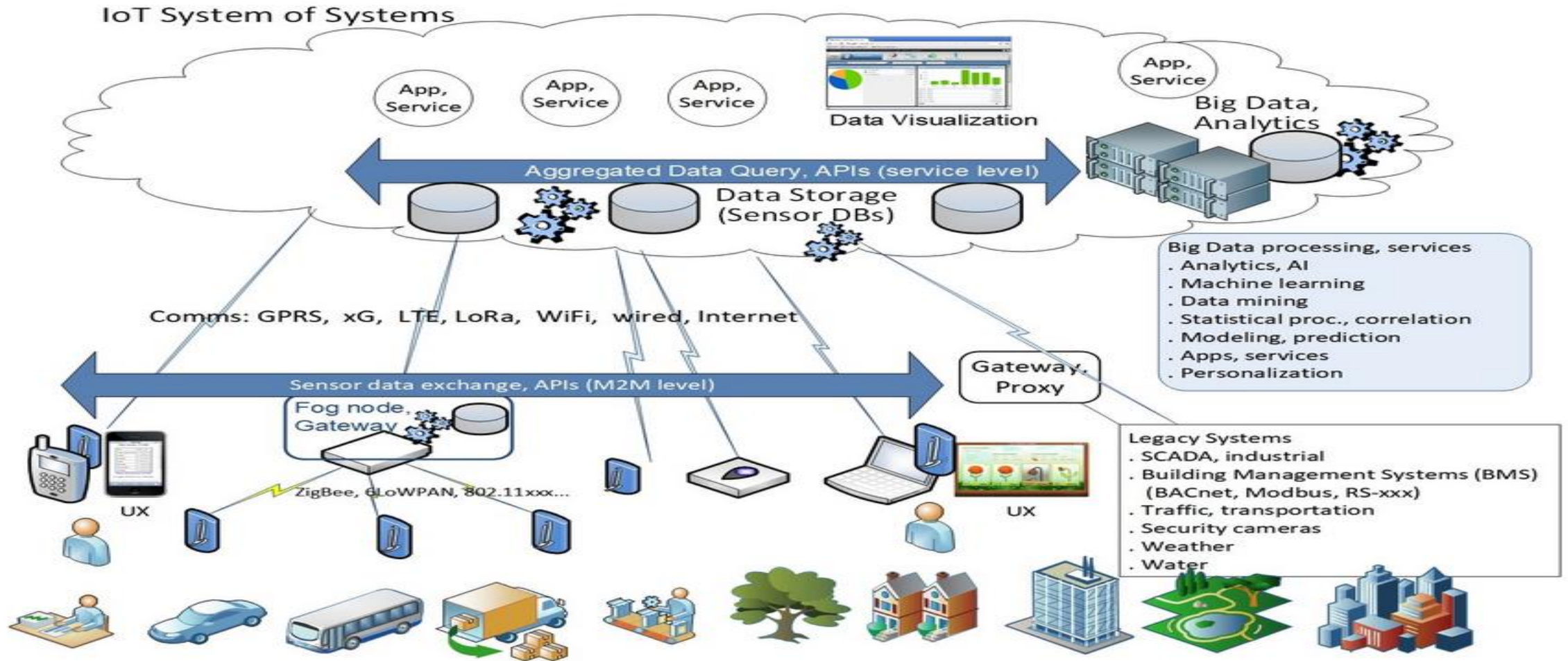
44 zettabytes  
44 Trillion GB





# 認識物聯網

SCADA: Supervisory Control And Data Acquisition (數據採集系統)  
BMS: Building Management Systems (建築物自動化設備管理系統)





# 認識物聯網

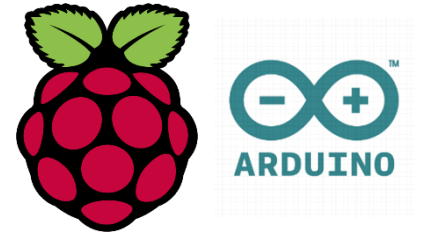






# 認識物聯網

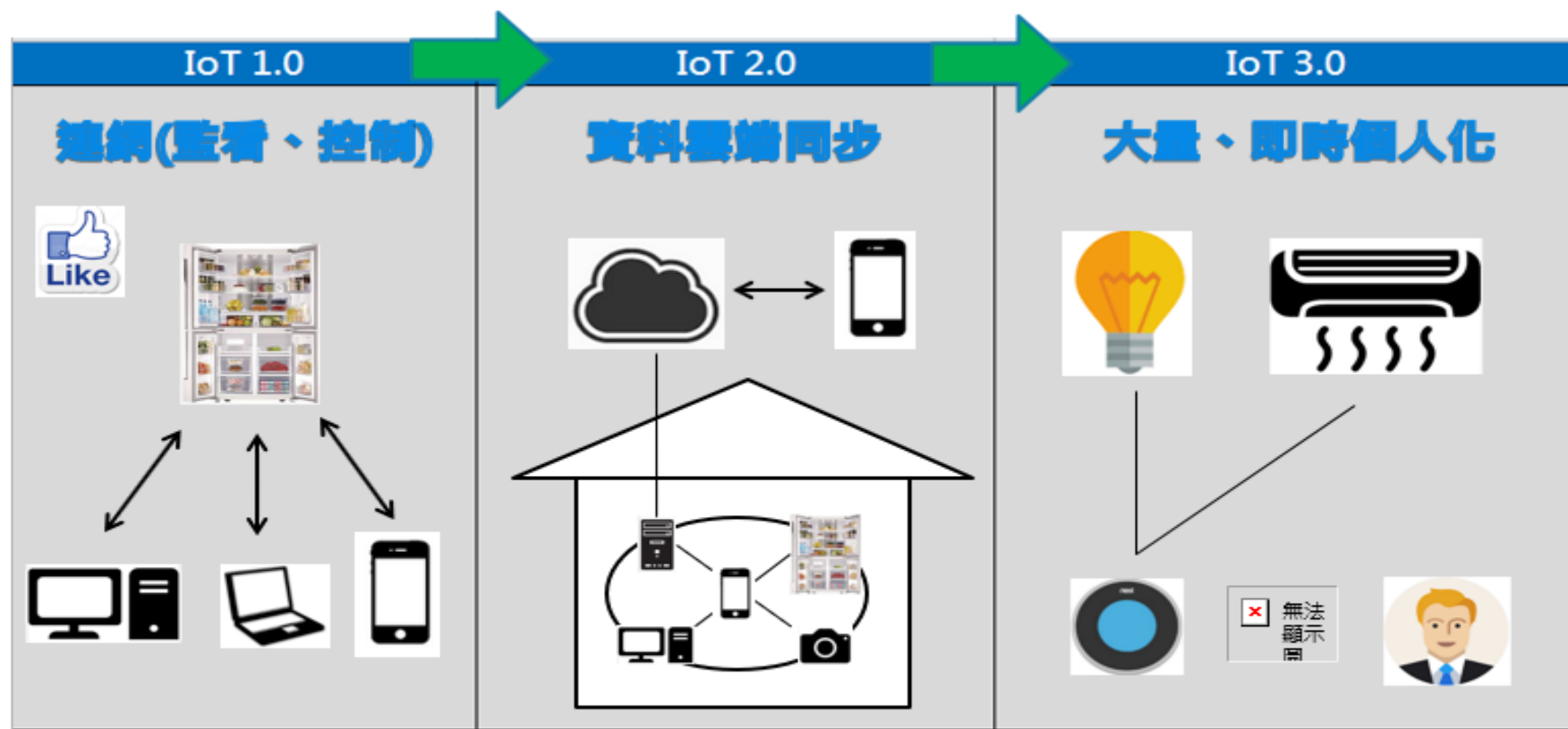
- ✓ Machine to machine (M2M) 架構
- ✓ 具備微處理機、網路通訊模組的微控制器
- ✓ 視需求決定是否需要作業系統
- ✓ 常見的物聯網微控制器設備：Arduino、ESP32、Raspberry PI、Nvidia Jetson Nano、Google Edge TPU
- ✓ 常見的作業系統：Raspberry OS、Windows IoT



# 物聯網將提供更多元的服務



# 連網、雲端同步、客製化服務



例如:  
可由 PC 得知冰箱溫度

例如:  
家電資料與外部資料結合  
提供服務

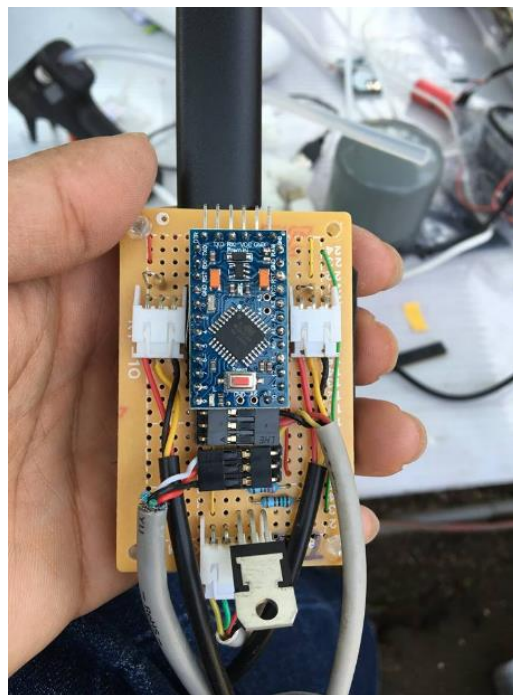
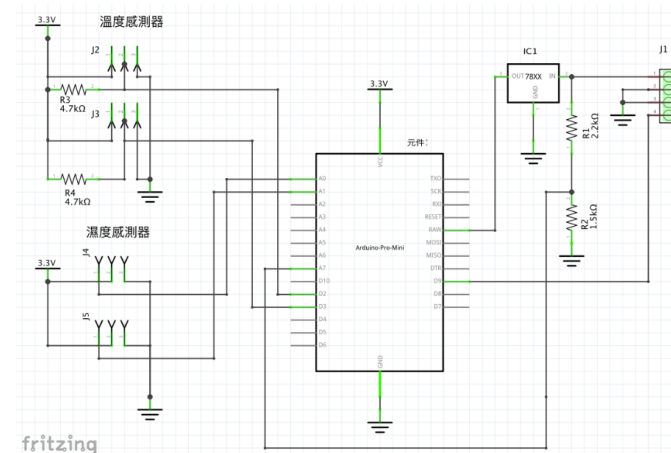
例如:  
系統判斷主人到家，並且  
根據過去學習經驗主動把冷氣打開



# 阿龜微氣候天眼通

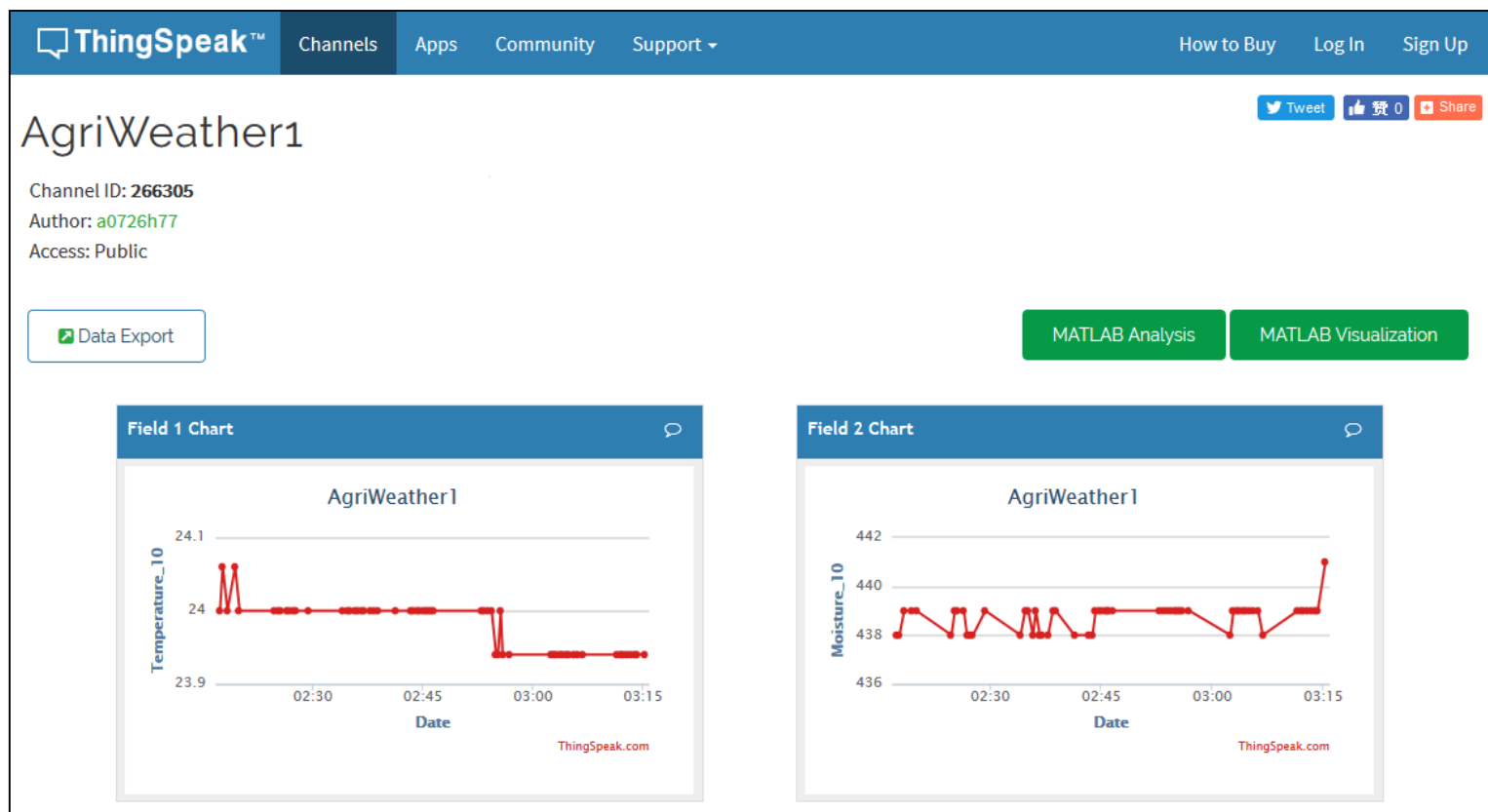
耕種環境環境資料

<https://www.facebook.com/agriweather/>





# 阿龜微氣候天眼通



<https://thingspeak.com/channels/266305>

# 空氣盒子

## ✓ 即時偵測資料

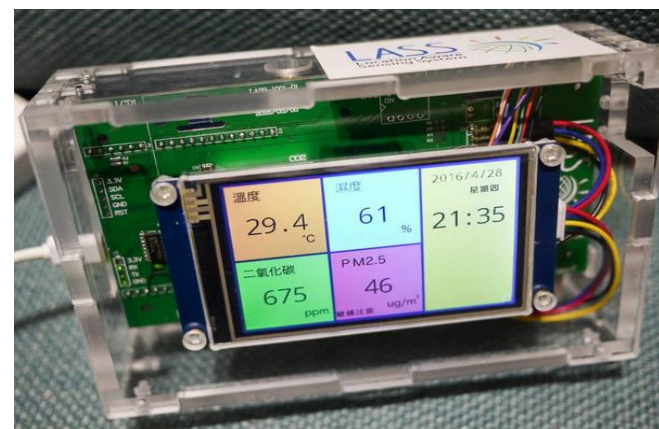
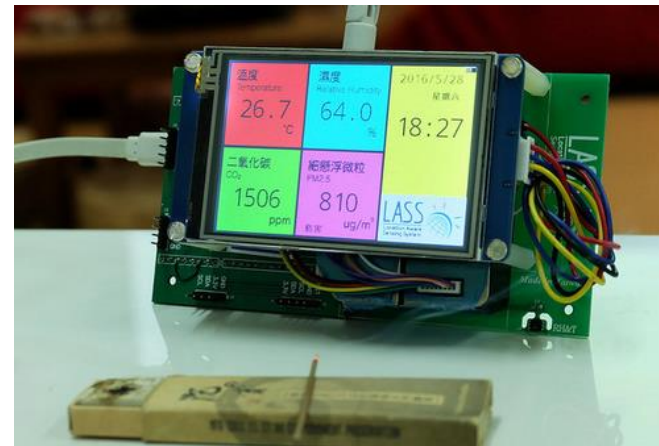
- ✓ 一小時內各地空氣盒子所蒐集的資料

## ✓ 歷史偵測資料

- ✓ 七天內的歷史資料

## ✓ 設備資料

- ✓ 空氣盒子設備的相關資料



# 空氣盒子

## ✓ 空氣品質 (Air Quality Indicator)

✓ 一公里見方空氣品質

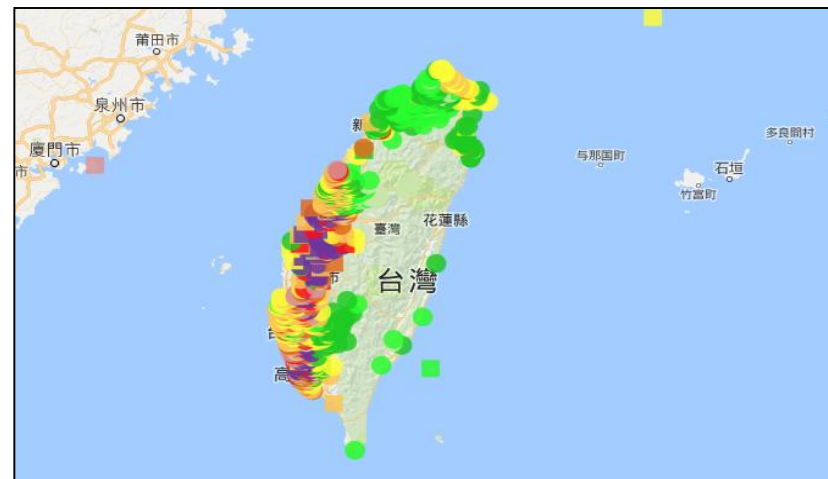
## ✓ 偵測、預測、監控、告警

✓ 空氣汙染指標

## ✓ 資料視覺化

✓ <https://airbox.edimaxcloud.com/>

✓ [https://pm25.lass-net.org/zh\\_tw/](https://pm25.lass-net.org/zh_tw/)



# 環境感測器網路系統

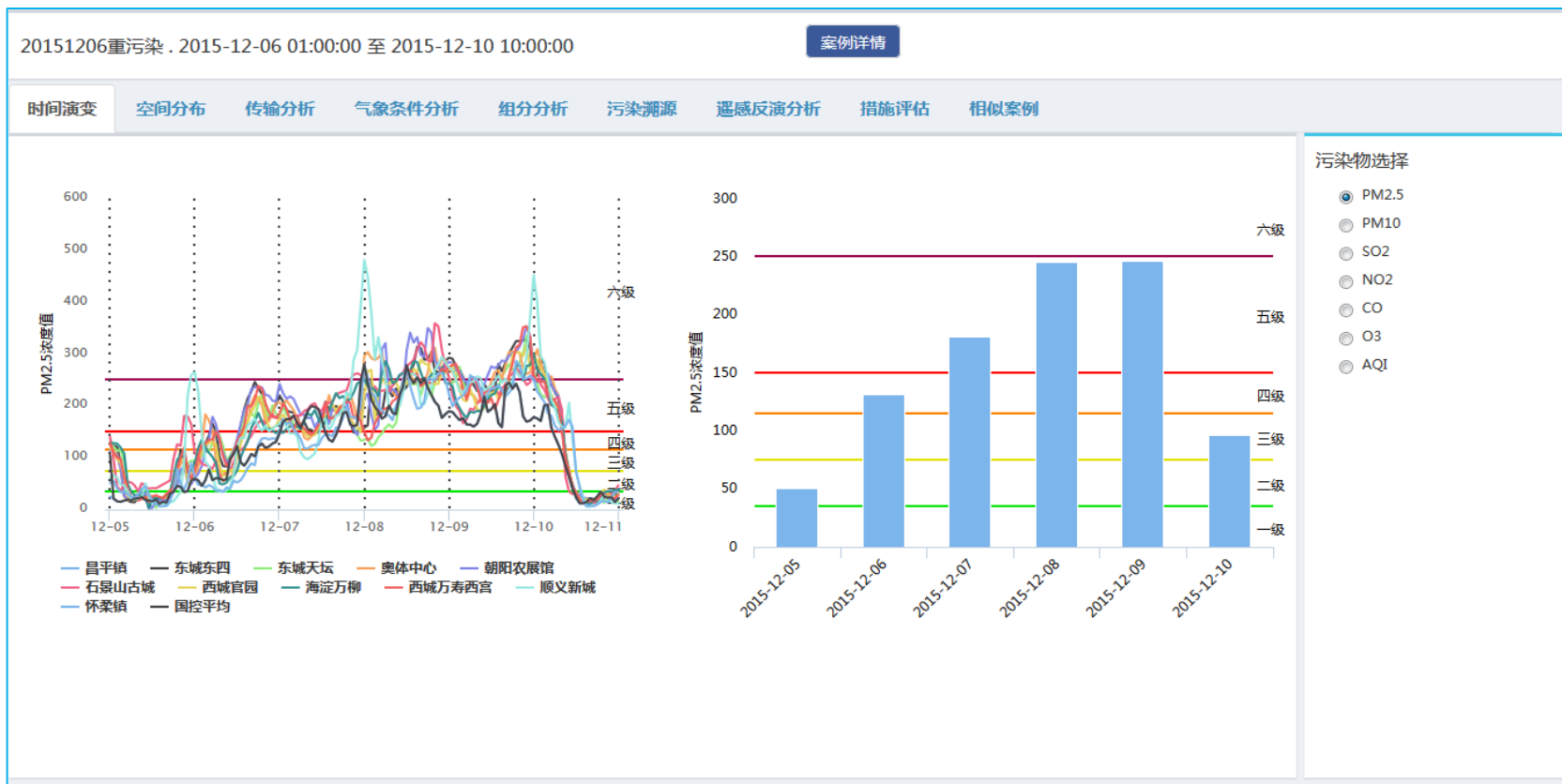
## ✓ LASS ( Location Aware Sensor System)

- ✓ 在物聯網的應用中，環境變化的即時監測很重要的一個領域，包括空氣品質、大地防災（如土石流預警）、公共建設監控（如橋樑），都對監控系統有很大的需求



# 環境感測器網路系統

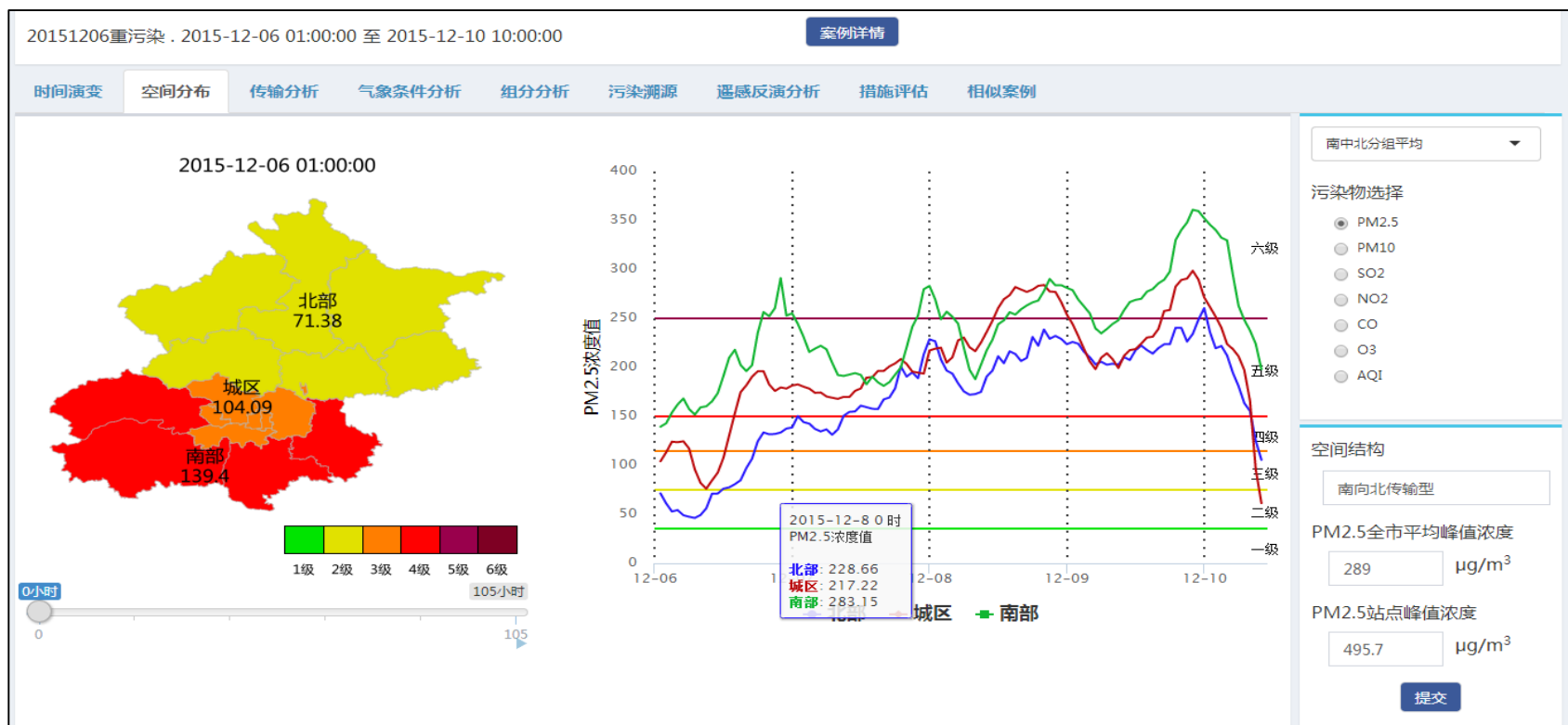
## ✓ 時間演變資料分析





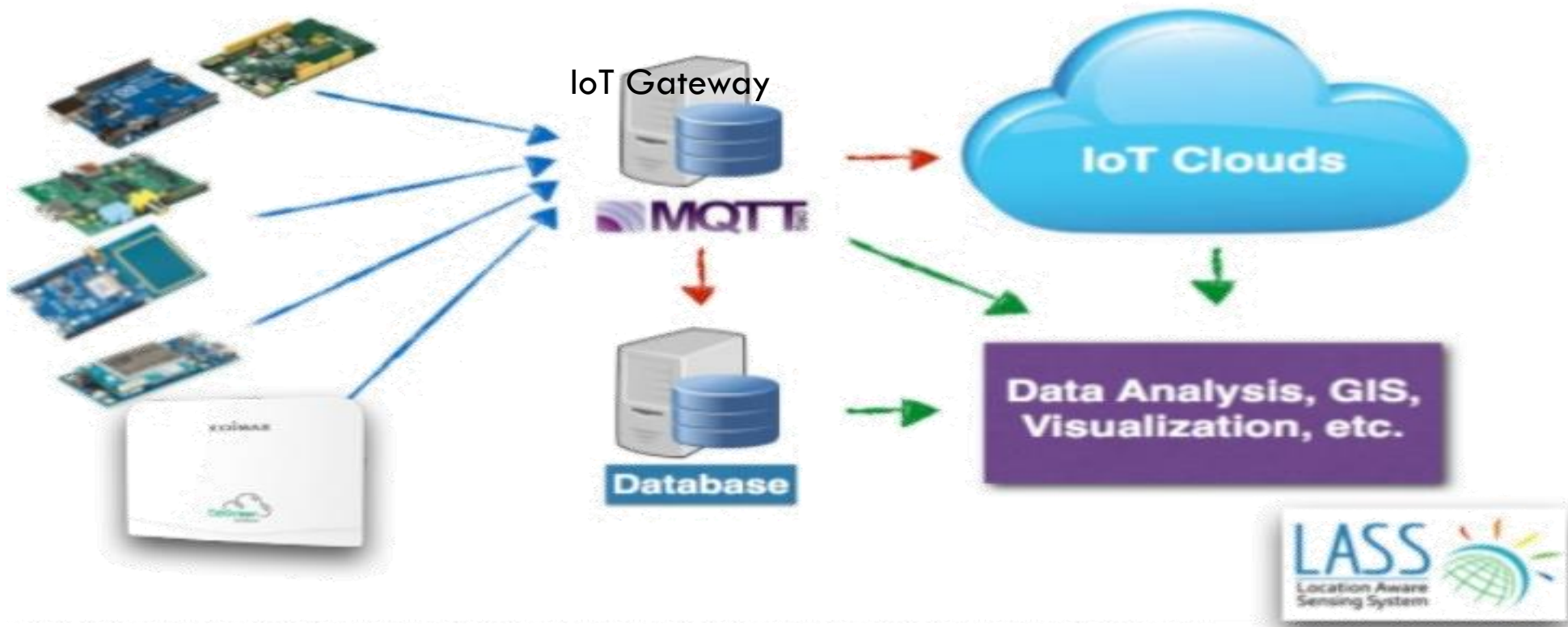
# 環境感測器網路系統

## ✓ 空間分布資料分析



# 環境感測器網路系統

- ✓ 快、慢資料流設計 (輕量級設計)



# Advantech IoT Solution Architecture

## Domain Specific Cloud Services (SaaS)



## Solution Ready Platforms (SRP)



## Embedded Platforms & Edge Intelligence Servers (EIS)



## IoT Sensing Devices



## Cloud Services

3rd party SaaS  
Dashboards  
Data Insights  
Machine Learning

REST API / Node-RED

## WISE-PaaS

WebAccess/SCADA  
WebAccess/HMI  
WebAccess/IVS  
WebAccess/IMM  
WebAccess/NMS  
Microsoft Azure  
ARM mbed Cloud  
Bluemix Ali AWS...

## Edge Intelligence

Windows / Linux /  
Industrial Android  
Device Management  
WISE-PaaS/RMM  
WISE-PaaS/Security  
WISE-PaaS/OTA  
Local Analytics  
3rd party containers

## WISE-Agent

RTOS / mbed OS  
MQTT / AMQP  
SSL / TLS



# 行動物聯網



# 樹莓派介紹





# 樹莓派介紹

- ✓ Raspberrypi.org 基金會開發設計
- ✓ 以低價硬體與自由軟體刺激電腦科學教育
- ✓ 公開的 datasheet、線路圖、原始碼
- ✓ 由 element14、RS Components、Egoman 生產製造

英國樹莓派組織

<https://www.raspberrypi.org/about/>



# 樹莓派介紹



✓ Raspberry Pi OS (原名稱為  
Raspbian)

✓ 可搭配 64-bit 的作業系統，

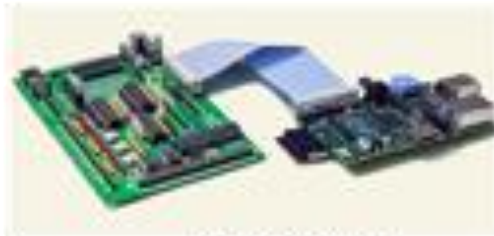
例如 Ubuntu 20.04 LTS

	樹莓派Pi 3 B+	樹莓派Pi 4
技術	藍芽升級4.2版本、 無線升級雙頻網路 2.4Ghz + 5Ghz 透過USB 可將網路提升至300Mbps	藍芽升級5.0版本、 無線雙頻網路 2.4Ghz + 5Ghz
芯片	雙核GPU的Broadcom 升級BCM2837B0， ARM的Cortex-A8(A53)四核處理器	雙核GPU的Broadcom BCM2837， ARM的Cortex-A72 四核處理器
處理速度	1.4 GHz	升級 換 1.5 GHz
內存量	1 GB LPDDR2 memory	2GB、4GB、8GB (3種) LPDDR2 memory
USB 2.0	4 x USB 2.0 sockets	<b>2x USB 3.0 ports</b> <b>2x USB 2.0 ports</b>
SD卡	MicroSD	MicroSD
插針	40 pin	40 pin
電源	+5.1V @ 2.5A，多了POE功能(網路線供電)	換 USB-C - 5V 3A，POE功能(網路線供電)
尺寸	86 x 56 x 17mm	85 x 56 x 17mm
HDMI接頭	HDMI接頭 X1	換 micro HDMI 接頭 X2
	產品壽命最少保證 至 2023年	

# 各式擴充板



PiFace



Gertboard



cooking-hacks



Pi Rack + Embedded Pi



raspicomm



Wolfson Audio Card



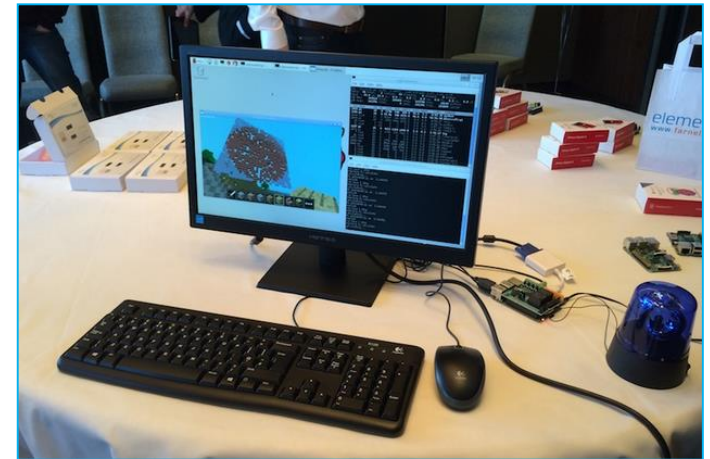
Arduino

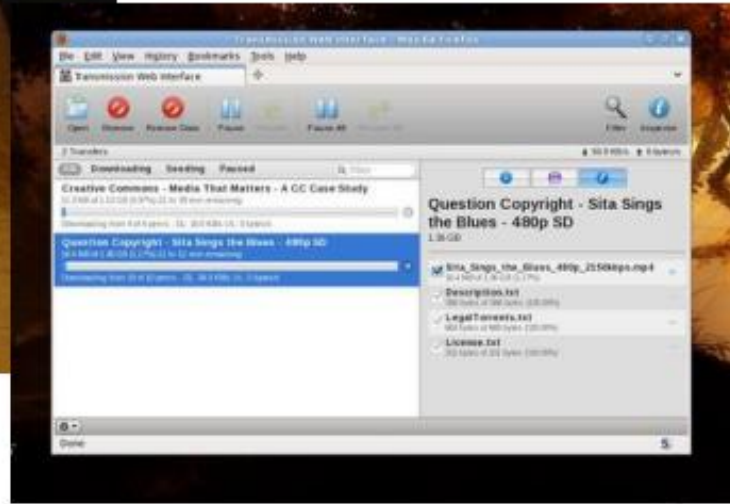


AlaMode

# 也是一台小電腦

- ✓ HDMI monitor
- ✓ USB interface
- ✓ 2.5 Amp Power Adapter
- ✓ Micro-USB cable
- ✓ 8GB+ MicroSD card
- ✓ Ethernet/ Wifi
- ✓ Computer to load initial card image
- ✓ GPIO

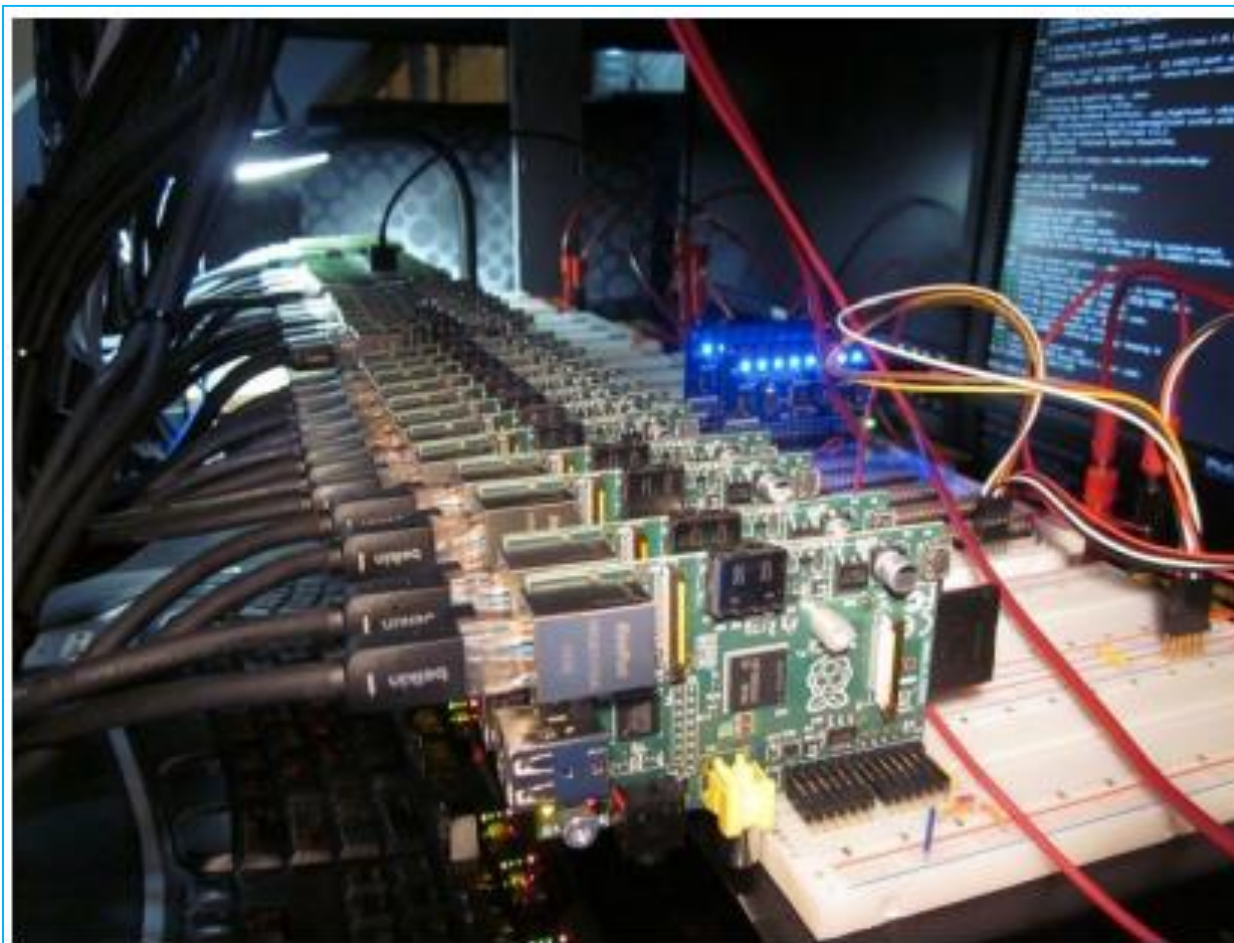






# 省電伺服器

- ✓ Web Server
- ✓ Mail Server
- ✓ FTP Server
- ✓ File Server
- ✓ VPN Server
- ✓ Printer Server
- ✓ Git Server



# 無線路由器

- ✓ Server
- ✓ Client



免費的路由器程式: <https://openwrt.org/>

# 無人自動監視器





# 衛星導航設備





# 遊戲機



遊戲模擬器:

<https://lifehacker.com/how-to-turn-your-raspberry-pi-into-a-retro-game-console-498561192>

<http://www.makezine.com.tw/make2599131456/portaberry-pi-3draspberry-pi>

# 比特幣採礦機

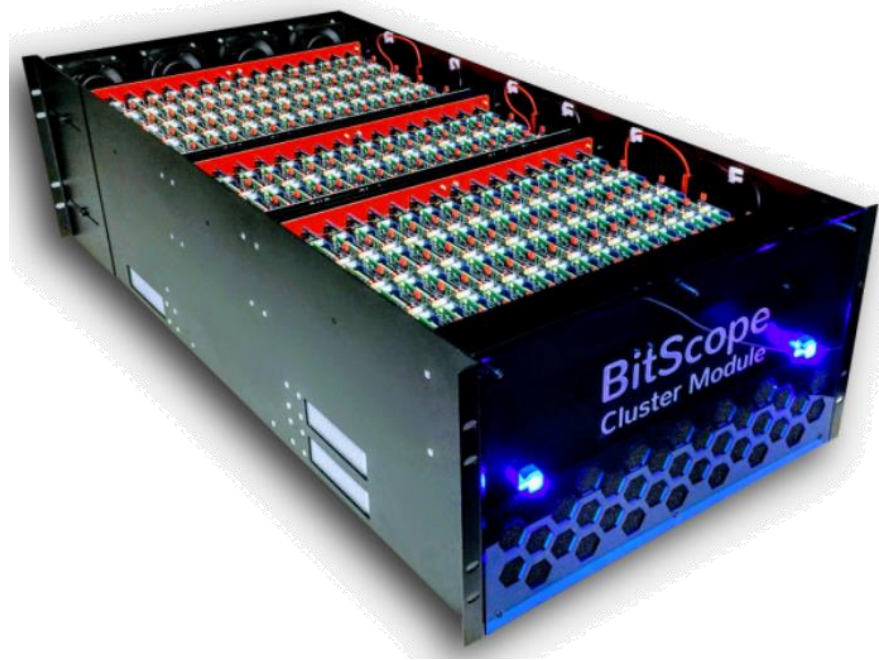


Company	Product	Cost (BTC)	Cost (USD)	Cost (EUR)	Speed (GHash/sec)	Power (Watts)
ASICMiner	Block Erupter USB	0.18	26	19	0.3	3
ASICMiner	Block Erupter Blade (overclock)	4.00	583	432	10.7	75
Avalon	Batch 3	75.00	10,935	8,100	65.0	620
Bitfury (EU)	Starter Kit (EU October Delivery)	9.26	1,350	1,000		0
Bitfury (EU)	Full Kit (EU August Delivery)	138.89	20,250	15,000		0
Bitfury (EU)	Full Kit (EU October Delivery)	69.44	10,125	7,500		0
Bitfury (US)	Starter Kit (October Delivery)	8.92	1,300	963		0
Bitfury (US)	Full Kit (August Delivery)	132.03	19,250	14,259		0
Bitfury (US)	Full Kit (October Delivery)	54.87	8,000	5,926	400.0	380

28

<https://tradeblock.com/blog/bitfury-begins-shipping-to-customers/>

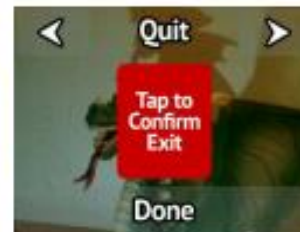
# 平行運算叢集



<https://www.ithome.com.tw/news/118304>

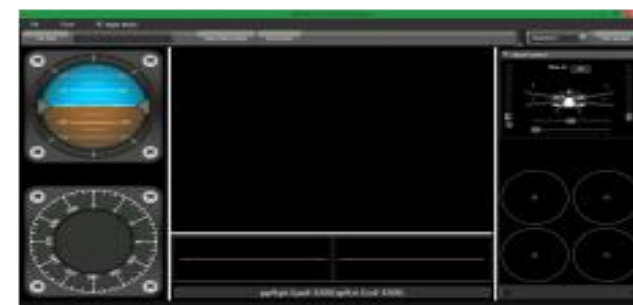


# 數位相機





# 四軸飛行器



打造四軸飛行器: <https://www.slideshare.net/itembedded/raspberry-pi-56539014>

# 孵化育種



<https://www.facebook.com/twFUNSUN/videos/1811451789075532>

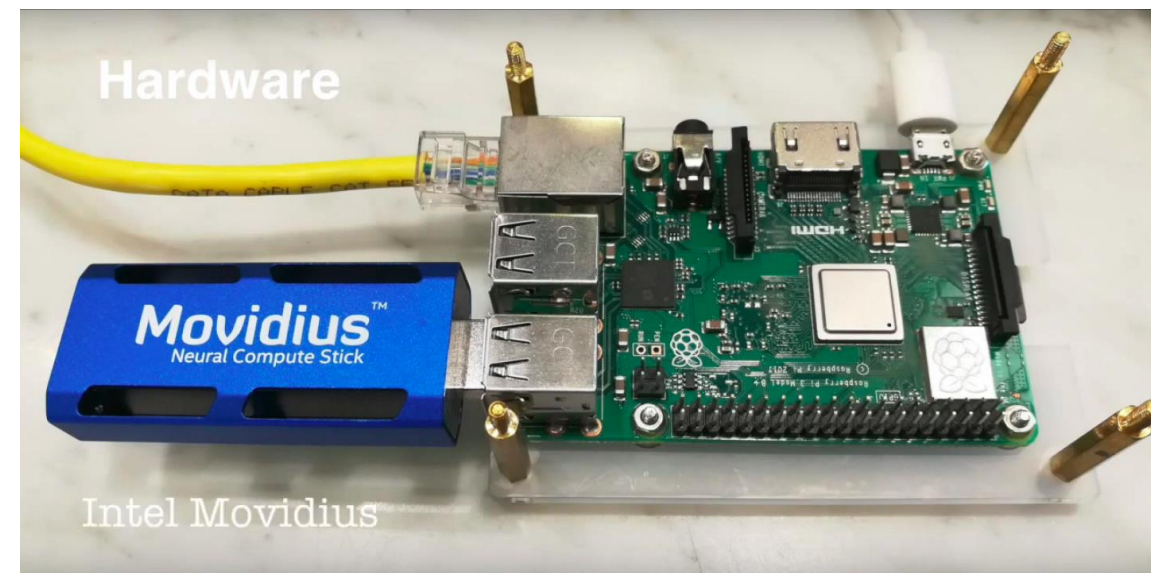
# TENSORFLOW LITE ON RASPBERRY PI

- ✓ Object detection

- ✓ <https://www.youtube.com/watch?v=aimSGOAUl8Y>

# 數位神經棒

- ✓ Neural compute stick2
  - ✓ 搭配 OpenVINO套件，加入影像辨識速度
  - ✓ <https://blog.cavedu.com/2019/08/30/opencvino-movidius>
  - ✓ <https://www.youtube.com/watch?v=NI6KppDqQnE>
  - ✓ <https://www.youtube.com/watch?v=xN117Ts4JJI>





# AWS-IOT-工業智造

## ✓ 參考連結

✓ <https://www.slideshare.net/AmazonWebServices/awsiot-155978430>

## 統計學解釋

變異係數：變異係數（coefficient of variation），又稱離散係數，是一個衡量資料離散程度的、沒有量綱的統計量。其值為標準差與平均值之比。

變異係數的計算公式為：

$$CV = \frac{\sigma}{\mu}$$



# 作業系統與安裝

## ✓ 下載 SDFormatter

- [https://www.sdcard.org/cht/downloads/formatter\\_4/eula\\_windows](https://www.sdcard.org/cht/downloads/formatter_4/eula_windows)

## ✓ 下載 Win32 Disk Imager

- <https://sourceforge.net/projects/win32diskimager>

## ✓ 下載 Image (2019-07: Raspbian Buster)

- <https://www.raspberrypi.org/downloads>



# 作業系統與安裝

## ✓ 接上螢幕輸出裝置 (HDMI)

## ✓ 修改設定

- ✓ 設定 System (SSH、VNC)
- ✓ 設定 Interfaces
- ✓ 設定 Localisation

## ✓ 啟動樹莓派網路 (建議手機無線網路)

## ✓ 修改root遠端登入

```
sudo nano /etc/ssh/sshd_config
```

```
PermitRootLogin yes
```

## ✓ 修改密碼

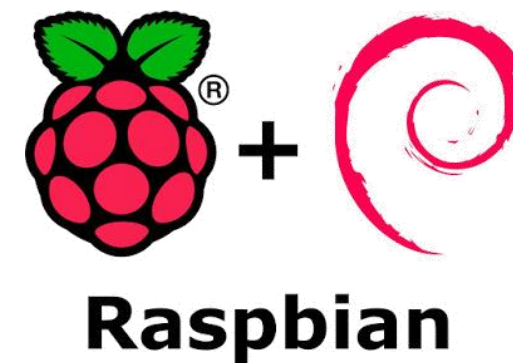
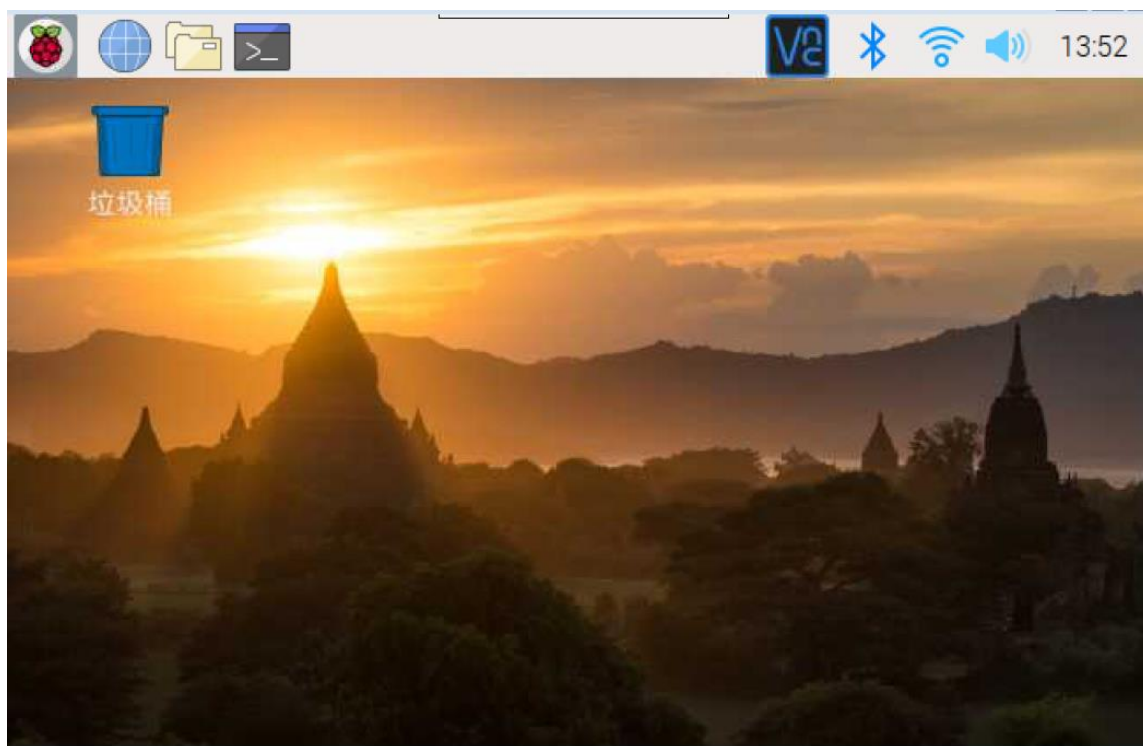
```
sudo passwd root
```

## ✓ 更新作業系統 & 重新開機

```
sudo apt-get update
```

# 作業系統介紹

- ✓ 衍生自 Debian 的發行版，目前 Raspberry Pi 的主力發行版
- ✓ 適合初學者，支援多、資料多，Trouble shooting 也比較容易





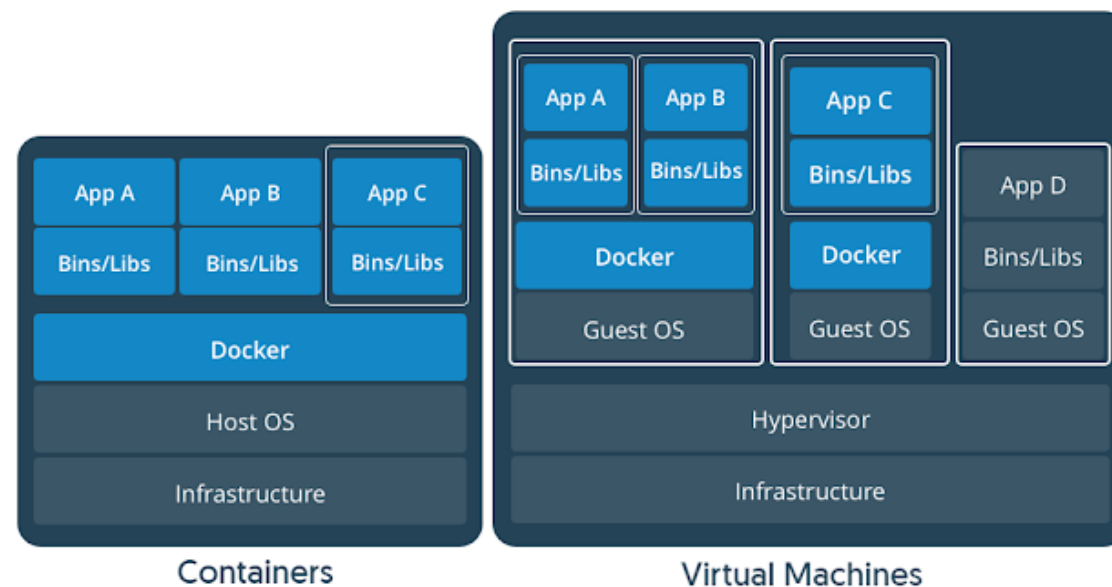
# 空氣盒子

## ✓ 下載課程程式

```
cd ~ ; sudo git clone https://github.com/orozcohsu/2019-ltu-airbox.git
```

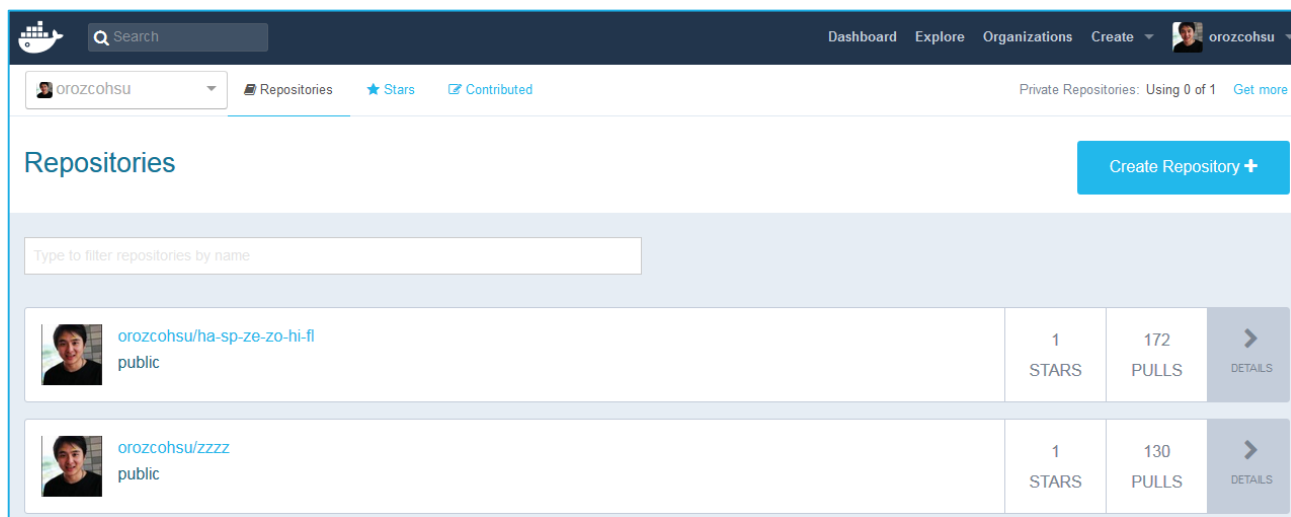
# DOCKER 簡介

- ✓ VMWARE
- ✓ Hyper-V
- ✓ Docker的技術為了解決重複存放的作業系統及提升效能而產生的一種技術
- ✓ 將作業系統從每個虛擬化抽離出來，分別使用容器放置所需執行的服務或程式



# DOCKER 簡介

- ✓ 介紹 Image 、 Container 及 Repository
- ✓ 叢集管理 K8S 套件
- ✓ 請於 Dockerhub 建立自己的帳號



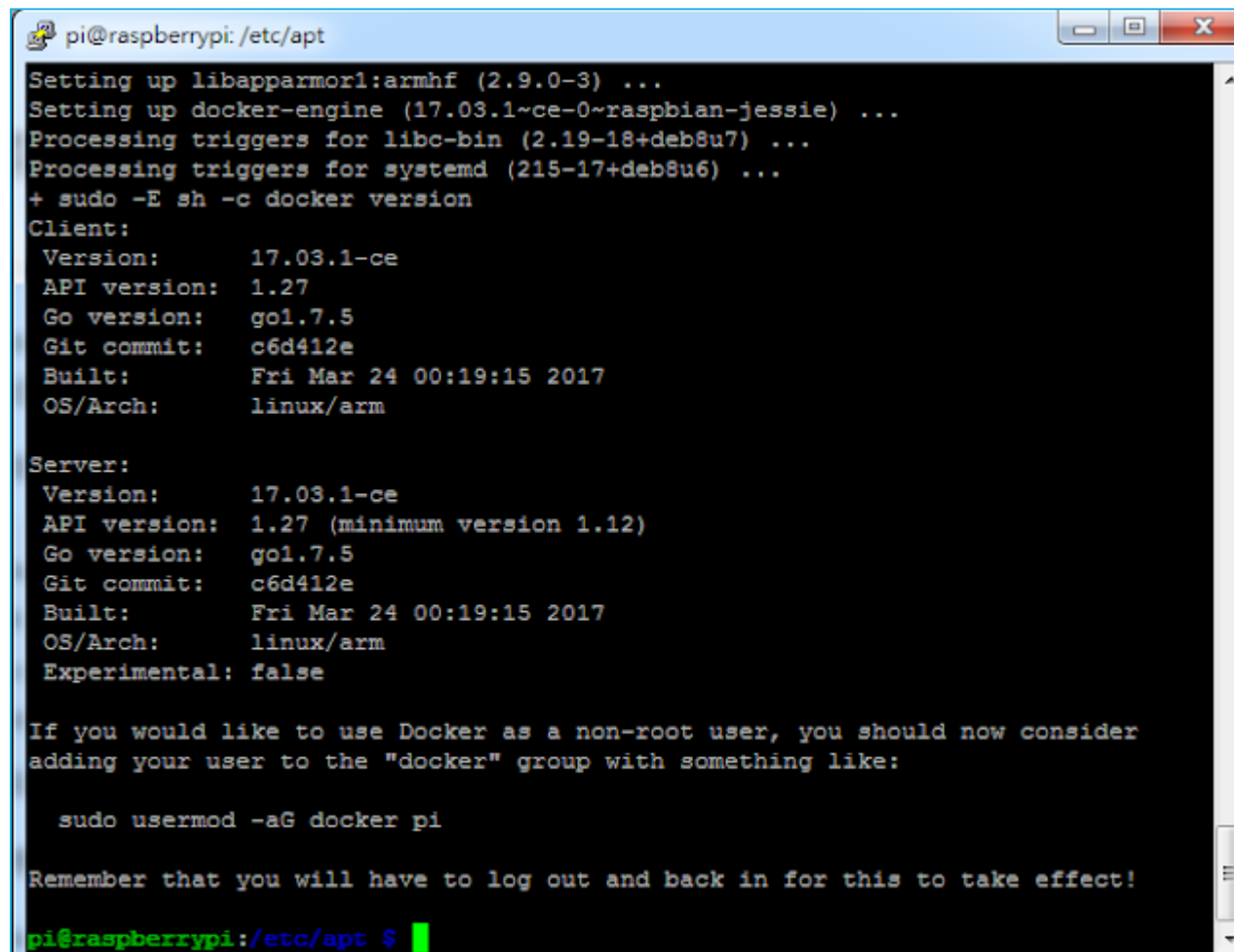
# DOCKER 簡介

- ✓ 以 root 登入
- ✓ 安裝 Docker engine

```
curl -sSL https://get.docker.com | sh
```

- ✓ 安裝完成之後，需要重新開機

```
reboot
```



```
pi@raspberrypi: /etc/apt
Setting up libapparmor1:armhf (2.9.0-3) ...
Setting up docker-engine (17.03.1~ce-0~raspbian-jessie) ...
Processing triggers for libc-bin (2.19-18+deb8u7) ...
Processing triggers for systemd (215-17+deb8u6) ...
+ sudo -E sh -c docker version
Client:
Version:      17.03.1-ce
API version:  1.27
Go version:   go1.7.5
Git commit:   c6d412e
Built:        Fri Mar 24 00:19:15 2017
OS/Arch:      linux/arm

Server:
Version:      17.03.1-ce
API version:  1.27 (minimum version 1.12)
Go version:   go1.7.5
Git commit:   c6d412e
Built:        Fri Mar 24 00:19:15 2017
OS/Arch:      linux/arm
Experimental: false

If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

    sudo usermod -aG docker pi

Remember that you will have to log out and back in for this to take effect!

pi@raspberrypi: /etc/apt $
```



# DOCKER 簡介

- ✓ 將 **pi** 使用者加入 **docker** 群組

```
sudo usermod -aG docker pi
```

- ✓ 自動啟動 **docker** 服務

```
sudo systemctl enable docker
```

- ✓ 尋找網路上可用的 **images** (樹莓派專屬)

```
sudo docker search armhf
```



apline 只有5MB

**docker** 教學介紹

<https://training.play-with-docker.com/ops-s1-hello/>

# DOCKER 簡介

## ✓ 從 **dockerhub** 取得 **images**

```
sudo docker pull resin/rpi-raspbian
```

## ✓ 尋找 **docker** 本身的 **images**

```
sudo docker images
```

## ✓ 啟動容器

```
sudo docker run -it resin/rpi-raspbian
```

映像檔(Image)	<ul style="list-style-type: none"><li>• 是一個唯讀的模板</li><li>• <b>docker</b> 執行容器前需要本地存在對應的映像檔，如果映像檔不存在本地，<b>docker</b> 會從映像檔倉庫下載</li></ul>
容器(Container)	<ul style="list-style-type: none"><li>• 容器是獨立執行的一個或一組應用，以及它們的執行態環境，對應到虛擬機，類比執行一整套作業系統</li><li>• <b>docker</b> 的容器是從映像檔建立的執行實例，它可以被啟動、開始、停止、刪除每個容器都是相互隔離的、保證安全的平台</li></ul>
倉庫(Repository)	<ul style="list-style-type: none"><li>• 是集中存放映像檔檔案的場所</li><li>• 官方提供 <a href="#">Docker Hub</a> 作為映像檔登錄及註冊(Registry)，這是類似 Github 程式碼儲存服務</li><li>• 一個 Docker Registry 中可以包含多個倉庫(Repository)</li><li>• 每個倉庫可以包含多個標籤(Tag)；每個標籤對應一個映像檔</li></ul>

-i: 開啟對話視窗

-t: 配置一個虛擬終端機

# DOCKER 簡介

## ✓ 離開內層容器

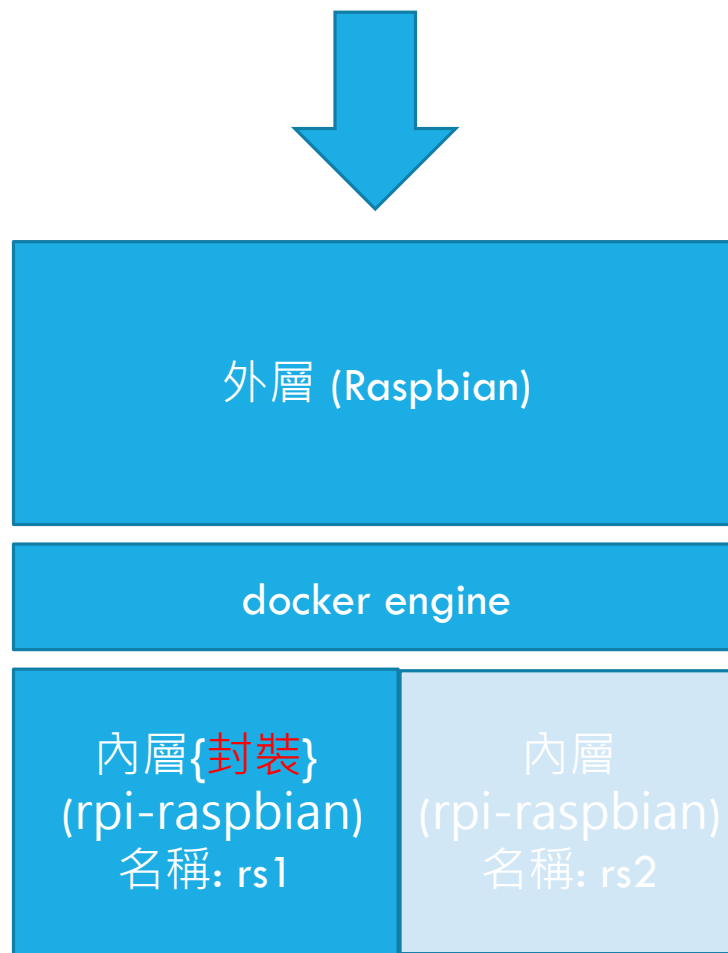
Ctrl 按下不放加按一下 p 與 q 鍵

## ✓ 檢查 docker 目前狀態

```
sudo docker ps -a
```

## ✓ 再進去內層容器

```
sudo docker exec -it [容器ID] /bin/bash
```



# DOCKER 簡介

- ✓ 建立 **dockerhub** 帳號
- ✓ <https://hub.docker.com>



# DOCKER 簡介

## ✓ 關閉容器

```
sudo docker rm -f [容器ID]
```

## ✓ 啟動新的容器

```
sudo docker run -it resin/rpi-raspbian
```

# DOCKER 簡介

## ✓ 刪除image

```
sudo docker rmi -f resin/rpi-raspbian
```

```
sudo docker images
```

# 製作 GPIO LED IMAGE

- ✓ 建立一個目錄 **GPIO**

```
sudo mkdir -p /home/pi/GPIO
```

- ✓ 在 **GPIO** 目錄內建立 **Dockerfile** 檔案

```
sudo touch /home/pi/GPIO/Dockerfile
```

- ✓ 編輯 **Dockerfile** 檔案內容

```
sudo vi/home/pi/GPIO/Dockerfile
```

```
FROM resin/rpi-raspbian:latest
```

```
ENTRYPOINT []
```

```
RUN apt-get -q update && \  
    apt-get -qy install python python-pip \  
    python-dev python-rpi.gpio
```

# 製作 GPIO LED IMAGE

## ✓ 製作新的 image

```
cd /home/pi/GPIO && sudo docker build -t gpio .
```

**gpio**: image名稱

## ✓ 查看新的 image

```
sudo docker images
```

## ✓ 我們把這個 image 當作自己的**母片**



因為沒有  
resin/rpi-raspbian 這個  
image，所以系統  
會自動下載



# 製作 GPIO LED IMAGE

- ✓ 製作另外一個新的 image

```
mkdir -p /home/pi/GPIO1 && cd /home/pi/GPIO1 && touch led.py
```

- ✓ 編輯 led.py

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
led_pin = 17
GPIO.setup(led_pin, GPIO.OUT)

while(True):
    GPIO.output(led_pin, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(led_pin, GPIO.LOW)
    time.sleep(1)
```

讓python可以控制樹莓派上的腳位

腳位採BCM方式

第17隻腳位(實體第11隻腳位)

腳位高電位

腳位低電位

# 製作 GPIO LED IMAGE

## ✓ 編輯 Dockerfile

```
cd /home/pi/GPIO1 && sudo nano Dockerfile
```

我們用剛剛  
建立的母片

Dockerfile 的內容

```
FROM gpio:latest
```

```
ADD ./led.py ./led.py  
CMD ["python", "led.py"]
```

## ✓ 製作新的 image

```
cd /home/pi/GPIO1 && sudo docker build -t 帳號/gpio1 .
```

帳號是自己在 dockerhub 上面申請的帳號

## ✓ 查看新的 image

```
sudo docker images
```

# 製作 GPIO LED IMAGE

✓ 查看腳位 (網路孔朝下)

gpio readall

Pi 2											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5V			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v			
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	

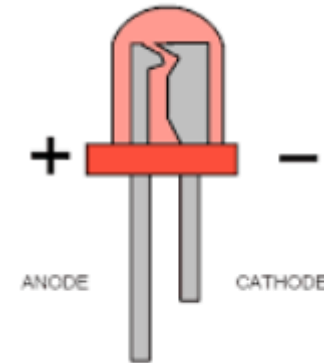
# DOCKER 操作 LED IMAGE (控制 LED 燈)

✓ 將 LED 插上樹莓派

✓ 執行 LED 程式

```
sudo docker run --privileged --rm 帳號/gpio1
```

privileged: 授與該容器全部權限



LED

長腳為正極，接 3.3V

短腳為負極，接GPIO0 (第11隻)



# DOCKER 操作 LED IMAGE (控制 LED 燈)

✓ 結束執行中的容器

✓ 上傳容器到 Dockerhub

```
docker login  
docker push 帳號/gpio
```

```
docker push 帳號/gpio1
```

✓ 查看 Dockerhub



orozcohsu/gpio1 ☆

By orozcohsu • Updated a minute ago

Container

# 作業

- ✓ 試試看做出不同的 **images**，在不同的 **BCM** 上發光
  - ✓ gpio1 => 1秒
  - ✓ gpio2 => 5秒
  - ✓ gpio3 => 10秒