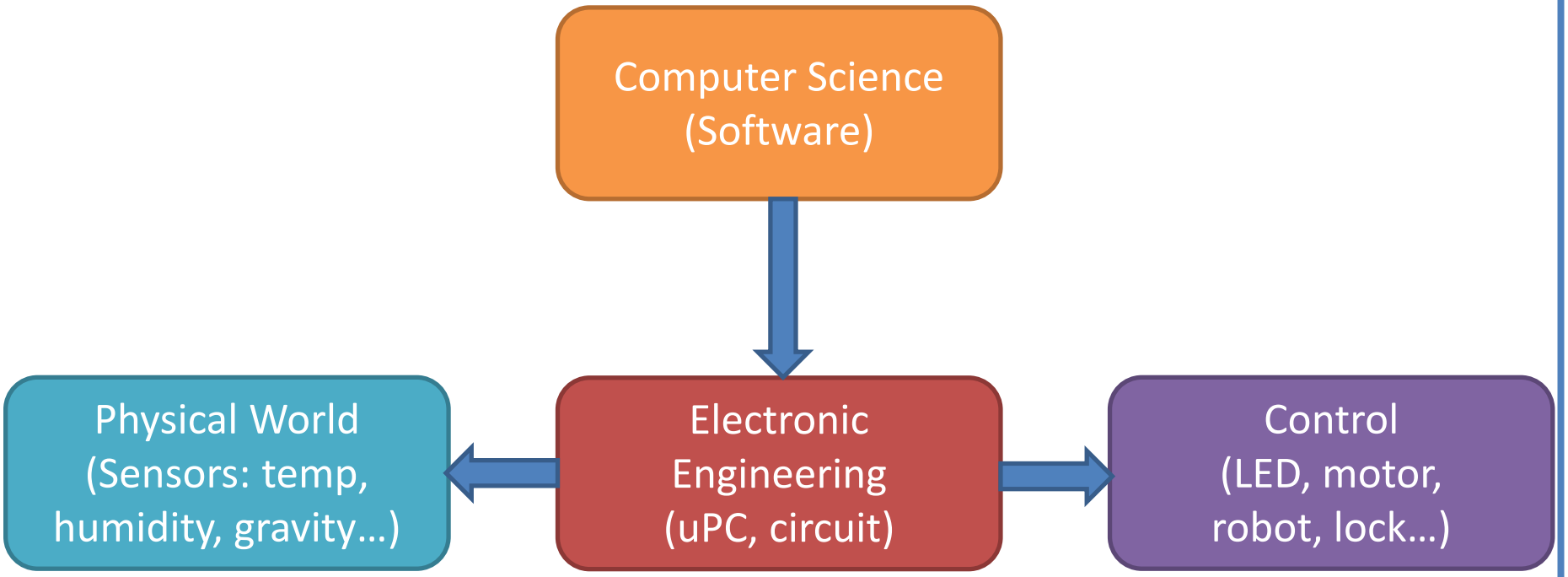
# 物聯網平台Raspberry PI 種子教師培訓

2019.01.17

# Aim of This Course



# Internet of Things

From connecting devices to human value

1 **Device connection**

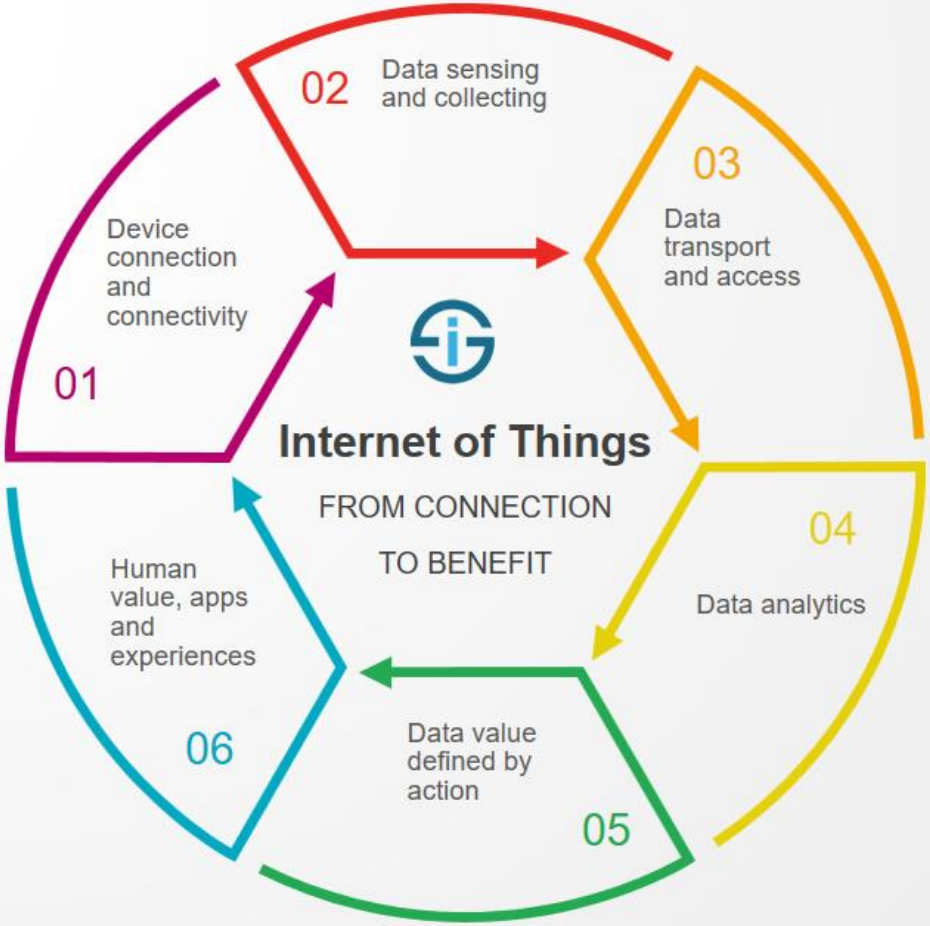
IoT devices
IoT connectivity
Embedded intelligence

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
  - 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, analyzing data**
    - **Communications**
    - **Cloud platform**
  - **13:30 – 15:00, 15:30 – 17:00: 動手操作, Q&A**

# 4. I/O of Raspberry PI, analyzing data

National Chiao Tung University

# Demo video

- MOTi Gym Tracker
  - https://www.facebook.com/motiwearable/videos/969310286482936/
    - 自動辨識20種以上的健身與重訓動作
    - 自動紀錄健身動作的運動次數
    - 追蹤紀錄分析，並呈現你的健身趨勢
    - 可建立個人健身課程
    - 分析個人健身目標達成率

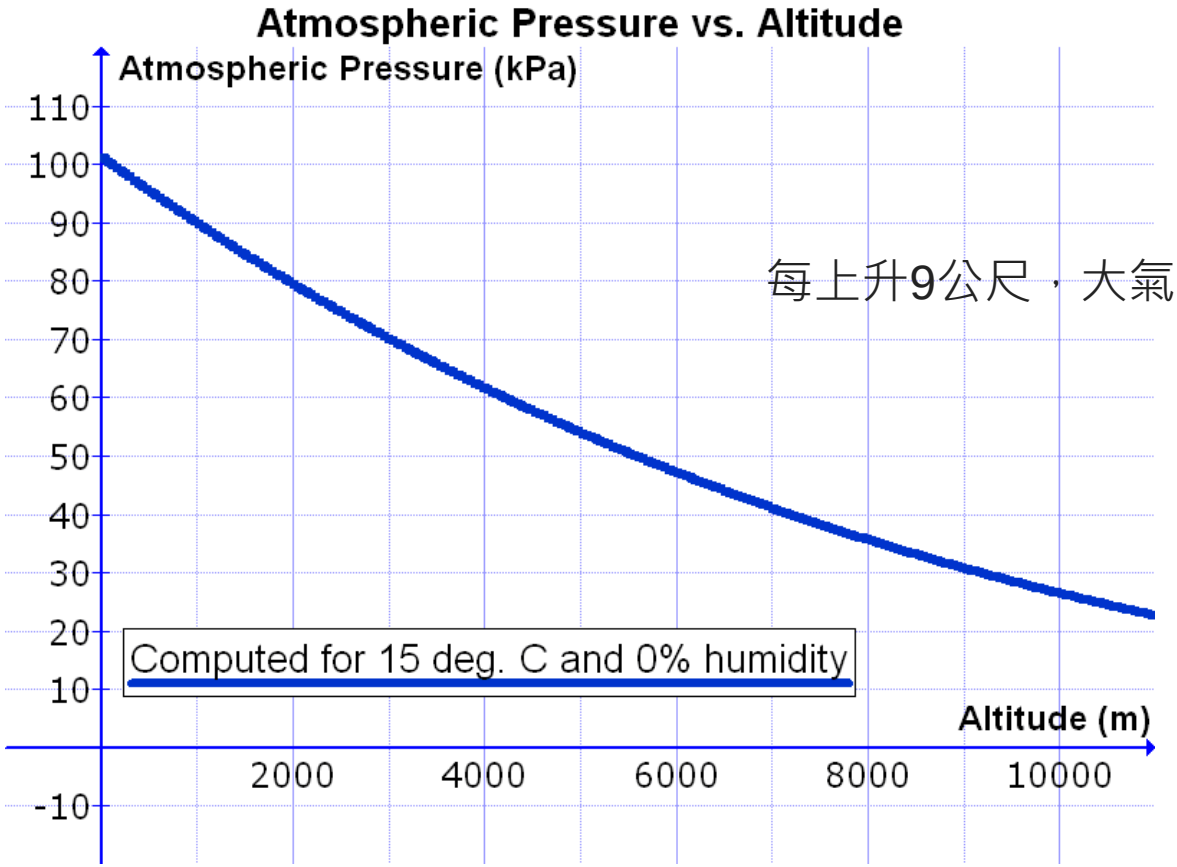
https://www.moti-wearable.com/shop/moti/

# Outline

- 1. 氣壓計 BMP
  - GY-801具有加速度/陀螺儀/電子羅盤/氣壓計
- 2. sensor資料處理方式
  - Low-pass filter
  - Kalman filter
  - Complementary filter

# 氣壓計

- 大氣壓力會隨著高度改變




每上升9公尺，大氣壓降低100Pa

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

# 氣壓計

- 大氣壓力與高度的關係

$$p = p_0 \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R_0 \cdot L}} \approx p_0 \cdot \left(1 - \frac{g \cdot h}{c_p \cdot T_0}\right)^{\frac{c_p \cdot M}{R_0}},$$

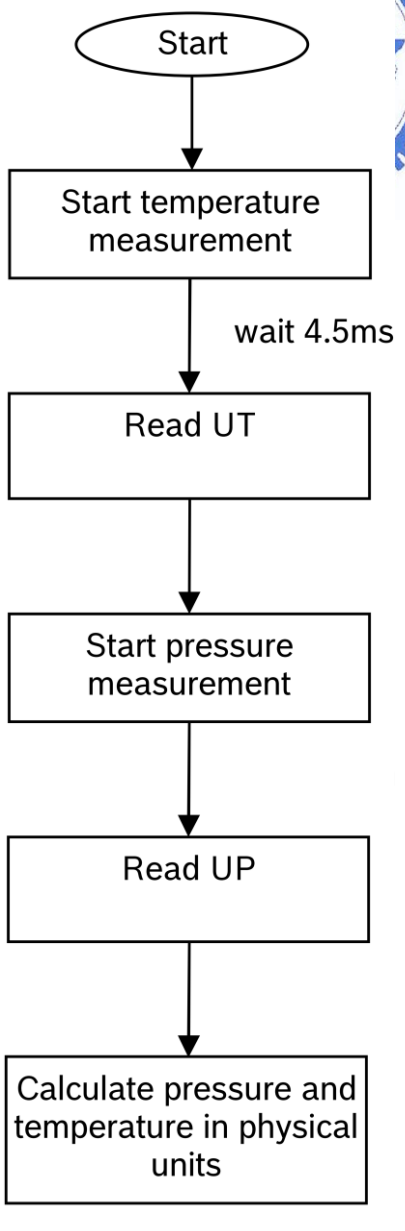
$$p \approx p_0 \cdot \exp\left(-\frac{g \cdot M \cdot h}{R_0 \cdot T_0}\right)$$

| Parameter | Description | Value |
|---|---|---|
| $p_0$ | sea level standard atmospheric pressure | 101325 Pa |
| $L$ | temperature lapse rate, = $g/c_p$ for dry air | 0.0065 K/m |
| $c_p$ | constant pressure specific heat | ~ 1007 J/(kg•K) |
| $T_0$ | sea level standard temperature | 288.15 K |
| $g$ | Earth-surface gravitational acceleration | 9.80665 $m/s^2$ |
| $M$ | molar mass of dry air | 0.0289644 kg/mol |
| $R_0$ | universal gas constant | 8.31447 J/(mol•K) |

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

# 氣壓計 BMP085

- Pressure sensing range: 300-1100 hPa
  - (9000m to -500m above sea level)
- Up to 0.03hPa / 0.25m resolution
- -40 to +85°C operational range
- +-2°C temperature accuracy
  - Temperature measurement included
- 2-pin i2c interface on chip




https://www.sparkfun.com/datasheets/Components/General/BST-BMP085-DS000-05.pdf
https://www.sparkfun.com/tutorials/253

# 氣壓計 BMP085

- BMP085 sample code:
  - https://github.com/adafruit/Adafruit_Python_BMP

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

cd examples/
sudo python simplectest.py
```

執行結果

```
pi@raspberrypi:~/gy80/4.bmp/examples$ sudo python simpletest.py
Temp = 25.50 *C
Pressure = 100536.00 Pa
Altitude = 65.90 m
Sealevel Pressure = 100536.00 Pa
```

# 氣壓計 BMP085

- Source code: examples/simpletest.py

載入module

```
#!/usr/bin/python
import Adafruit_BMP.BMP085 as BMP085
sensor = BMP085.BMP085()

print('Temp = {0:0.2f} *C'.format(sensor.read_temperature()))
print('Pressure = {0:0.2f} Pa'.format(sensor.read_pressure()))
print('Altitude = {0:0.2f} m'.format(sensor.read_altitude()))
print('Sealevel Pressure = {0:0.2f}
Pa'.format(sensor.read_sealevel_pressure()))
```

去看 Adafruit_BMP/BMP085.py 吧

# 氣壓計 BMP085

- 在 Adafruit_BMP/BMP085.py 裡面

```
# BMP085 default address.
BMP085_I2CADDR          = 0x77

# Operating Modes
BMP085_ULTRALOWPOWER  = 0
BMP085_STANDARD       = 1
BMP085_HIGHRES        = 2
BMP085_ULTRAHIGHRES   = 3
```

Code有設定mode=BMP085_STANDARD (可依需求改成其他mode)

| Mode | Parameter *oversampling_setting* | Internal number of samples | Conversion time pressure max. [ms] | Avg. current @ 1 sample/s typ. [µA] | RMS noise typ. [hPa] | RMS noise typ. [m] |
|---|---|---|---|---|---|---|
| ultra low power | 0 | 1 | 4.5 | 3 | 0.06 | 0.5 |
| standard | 1 | 2 | 7.5 | 5 | 0.05 | 0.4 |
| high resolution | 2 | 4 | 13.5 | 7 | 0.04 | 0.3 |
| ultra high resolution | 3 | 8 | 25.5 | 12 | 0.03 | 0.25 |

在code裡面設定datasheet提到的參數

# 氣壓計 BMP085

- 讀取校正資料(Calibration coefficients )

| | BMP085 reg adr | |
|---|---|---|
| Parameter | MSB | LSB |
| AC1 | 0xAA | 0xAB |
| AC2 | 0xAC | 0xAD |
| AC3 | 0xAE | 0xAF |
| AC4 | 0xB0 | 0xB1 |
| AC5 | 0xB2 | 0xB3 |
| AC6 | 0xB4 | 0xB5 |
| B1 | 0xB6 | 0xB7 |
| B2 | 0xB8 | 0xB9 |
| MB | 0xBA | 0xBB |
| MC | 0xBC | 0xBD |
| MD | 0xBE | 0xBF |

```
# BMP085 Registers
BMP085_CAL_AC1      = 0xAA  # R   Calibration data (16 bits)
BMP085_CAL_AC2      = 0xAC  # R   Calibration data (16 bits)
BMP085_CAL_AC3      = 0xAE  # R   Calibration data (16 bits)
BMP085_CAL_AC4      = 0xB0  # R   Calibration data (16 bits)
BMP085_CAL_AC5      = 0xB2  # R   Calibration data (16 bits)
BMP085_CAL_AC6      = 0xB4  # R   Calibration data (16 bits)
BMP085_CAL_B1       = 0xB6  # R   Calibration data (16 bits)
BMP085_CAL_B2       = 0xB8  # R   Calibration data (16 bits)
BMP085_CAL_MB       = 0xBA  # R   Calibration data (16 bits)
BMP085_CAL_MC       = 0xBC  # R   Calibration data (16 bits)
BMP085_CAL_MD       = 0xBE  # R   Calibration data (16 bits)
```

Code的這些參數來自datasheet資料

在code裡面設定datasheet提到的參數

# 氣壓計 BMP085

- 讀取溫度與壓力的資訊 (設定address)

```
# Commands
BMP085_READTEMPCMD        = 0x2E
BMP085_READPRESSURECMD  = 0x34
```

```
# BMP085 Registers
BMP085_CONTROL          = 0xF4
BMP085_TEMPDATA         = 0xF6
BMP085_PRESSUREDATA  = 0xF6
```

datasheet

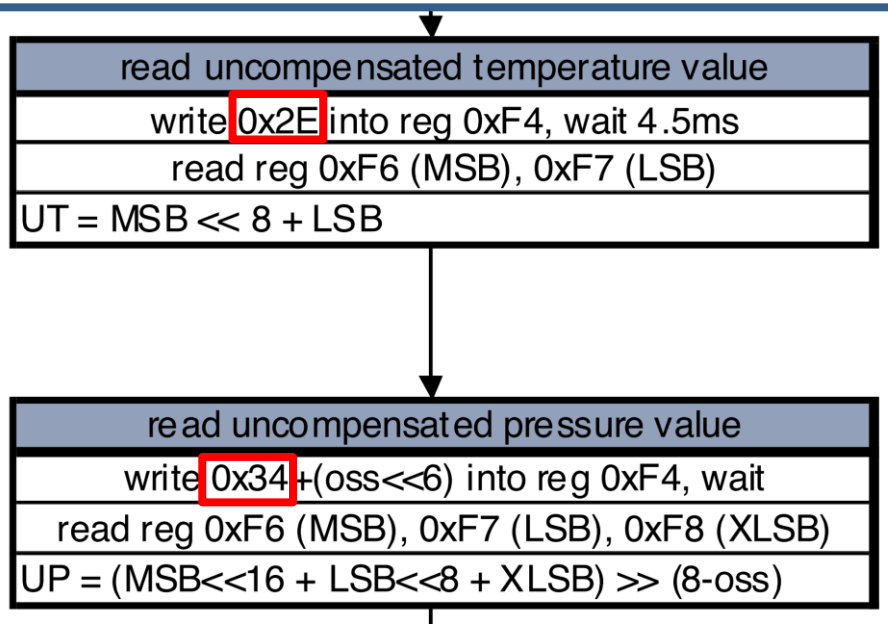



Fig. Calculation of pressure and temperature for BMP085 (節錄)

在code裡面設定datasheet提到的參數

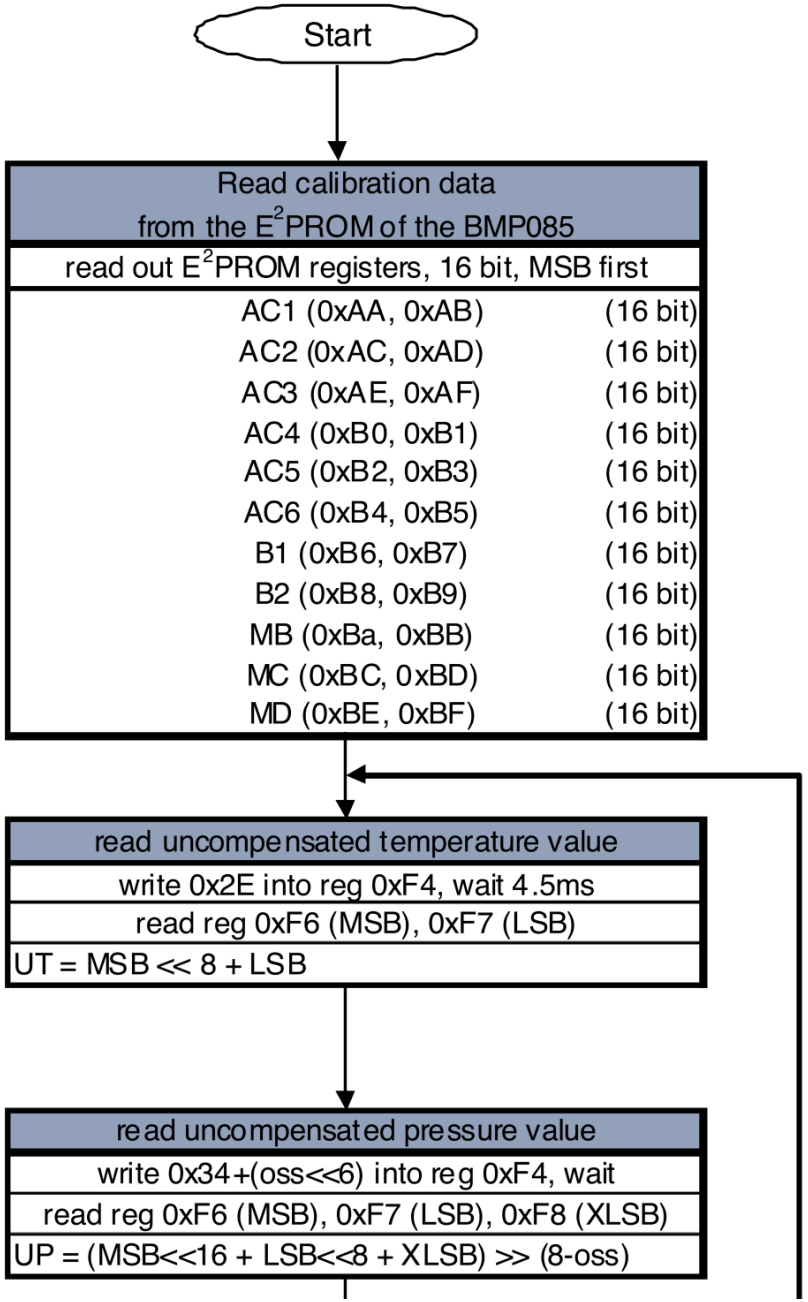
# 氣壓計 BMP085

- Debug用的校正參數 (datasheet的範例)

```
def _load_datasheet_calibration(self):
    # Set calibration from values
    # in the datasheet example.
    # Useful for debugging the
    # temp and pressure calculation accuracy.
    self.cal_AC1 = 408
    self.cal_AC2 = -72
    self.cal_AC3 = -14383
    self.cal_AC4 = 32741
    self.cal_AC5 = 32757
    self.cal_AC6 = 23153
    self.cal_B1 = 6190
    self.cal_B2 = 4
    self.cal_MB = -32767
    self.cal_MC = -8711
    self.cal_MD = 2868
```

| | |
|---|---|
| AC1 = | 408 |
| AC2 = | -72 |
| AC3 = | -14383 |
| AC4 = | 32741 |
| AC5 = | 32757 |
| AC6 = | 23153 |
| B1 = | 6190 |
| B2 = | 4 |
| MB = | -32768 |
| MC = | -8711 |
| MD = | 2868 |

Datasheet example




| | example: | C code function: | type: |
|---|---|---|---|
| | | bmp085_get_cal_param | |
| AC1 = | 408 | | short |
| AC2 = | -72 | | short |
| AC3 = | -14383 | | short |
| AC4 = | 32741 | | unsigned short |
| AC5 = | 32757 | | unsigned short |
| AC6 = | 23153 | | unsigned short |
| B1 = | 6190 | | short |
| B2 = | 4 | | short |
| MB = | -32768 | | short |
| MC = | -8711 | | short |
| MD = | 2868 | | short |
| | | bmp085_get_ut | |
| UT = | **27898** | | long |
| oss = | 0 | | |
| *= oversampling_setting* | | | short (0 .. 3) |
| *(ultra low power mode)* | | bmp085_get_up | |
| UP = | **23843** | | long |

Datasheet 流程圖

**calculate true temperature**

$X1 = (UT - AC6) * AC5 / 2^{15}$

$X2 = MC * 2^{11} / (X1 + MD)$

$B5 = X1 + X2$

$T = (B5 + 8) / 2^{4}$ 校正參數會在這邊用到

**calculate true pressure**

$B6 = B5 - 4000$

$X1 = (B2 * (B6 * B6 / 2^{12})) / 2^{11}$

$X2 = AC2 * B6 / 2^{11}$

$X3 = X1 + X2$ 校正參數會在這邊用到

$B3 = ((AC1*4+X3) << oss + 2) / 4$

$X1 = AC3 * B6 / 2^{13}$

$X2 = (B1 * (B6 * B6 / 2^{12})) / 2^{16}$

$X3 = ((X1 + X2) + 2) / 2^{2}$

$B4 = AC4 * (unsigend\ long)(X3 + 32768) / 2^{15}$

$B7 = ((unsigned\ long)UP - B3) * (50000 >> oss)$

if (B7 < 0x80000000) { p = (B7 * 2) / B4 }

else { p = (B7 / B4) * 2 }

$X1 = (p / 2^{8}) * (p / 2^{8})$

$X1 = (X1 * 3038) / 2^{16}$

$X2 = (-7357 * p) / 2^{16}$

$p = p + (X1 + X2 + 3791) / 2^{4}$

**display temperature and pressure value**

bmp085_get_temperature

| | | | |
|---|---|---|---|
| X1 = | 4743 | | long |
| X2 = | -2344 | | long |
| B5 = | 2399 | | long |
| **T =** | **150** | ***temp in 0.1℃*** | long |

BMP085_calpressure

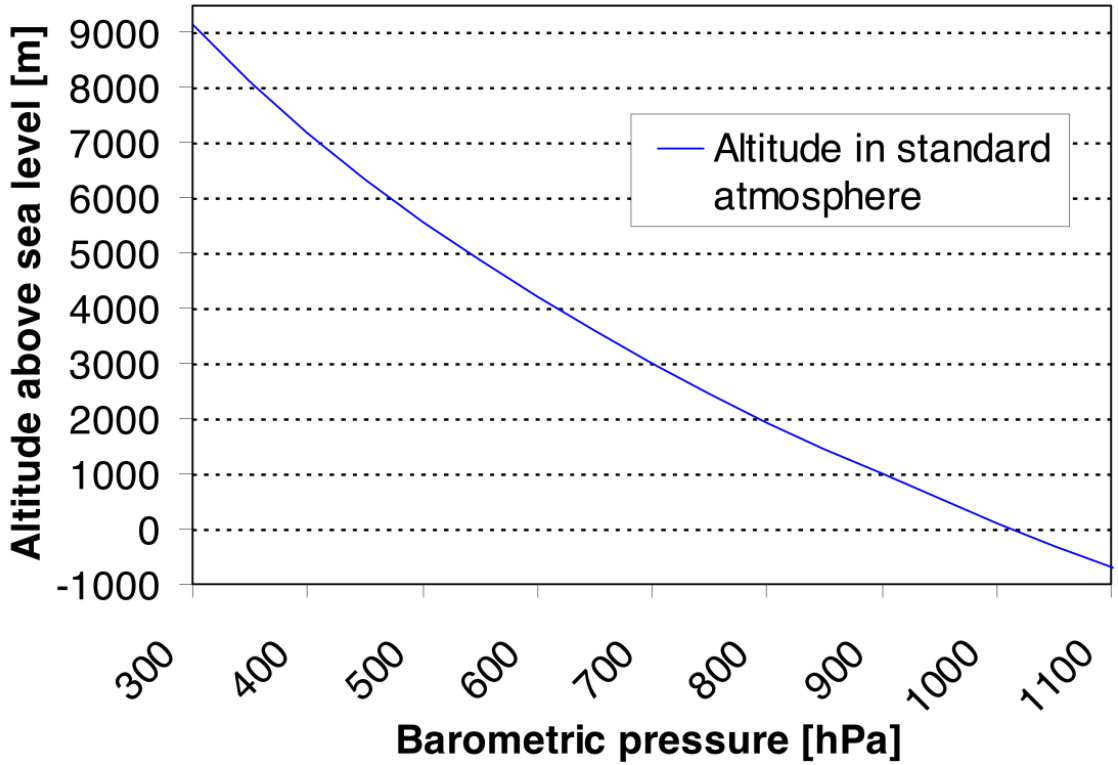
| | | | |
|---|---|---|---|
| B6 = | -1601 | | long |
| X1 = | 1 | | long |
| X2 = | 56 | | long |
| X3 = | 57 | | long |
| B3 = | 422 | | long |
| X1 = | 2810 | | long |
| X2 = | 59 | | long |
| X3 = | 717 | | long |
| B4 = | 33457 | | unsigned long |
| B7 = | 1171050000 | | unsigned long |
| p = | 70003 | | long |
| | | | long |
| X1 = | 74529 | | long |
| X1 = | 3454 | | long |
| X2 = | -7859 | | long |
| **p =** | **69964** | ***press. in Pa*** | long |

Datasheet 流程圖

# 氣壓計 BMP085

- 執行程式 -> 得到大氣壓力 -> 計算高度
  - 根據測量到的氣壓值, 跟水平面的氣壓值相比
  - 再利用下面公式計算高度

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$



# 氣壓計 BMP085

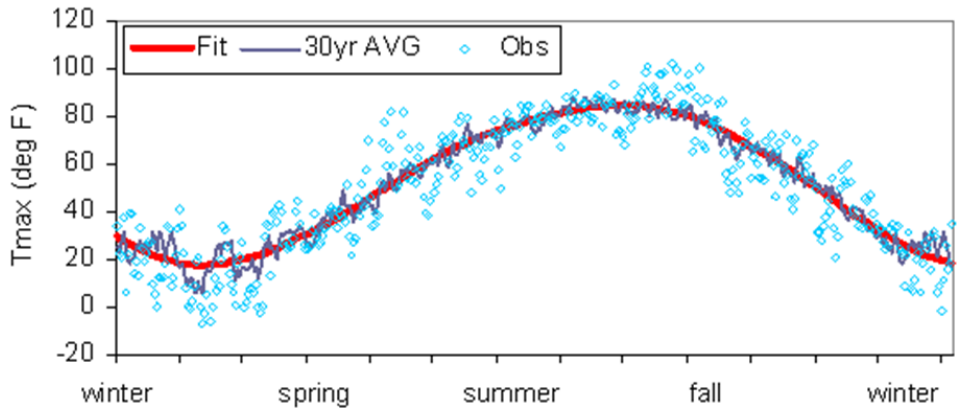
- **Quiz1: 測量目前的氣壓計資訊, 目前的高度是多少? 覺得合理嗎**

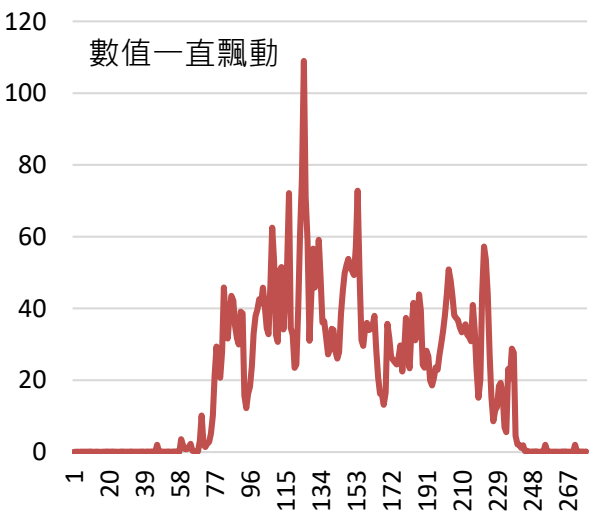
- **Quiz2: 整合目前的code, 讓python可以同時顯示加速度+陀螺儀+電子羅盤+氣壓計**

```
ACC: x:  -26 ,y:  -6 ,z:  -251 ; MAG: x:  17 ,y:  -3 ,z:  8 ; GYRO: x:  -359 ,y  -43 ,z:  -112; h: 60 .
ACC: x:  -27 ,y:  -8 ,z:  -246 ; MAG: x:   8 ,y:  -4 ,z:  4 ; GYRO: x:  -359 ,y: -43 ,z:  -112; h: 61 .
ACC: x:  -28 ,y:  -6 ,z:  -248 ; MAG: x:  -4 ,y:  1 ,z:  15 ; GYRO: x:  -359 ,y  -43 ,z:  -112; h: 60 .
```

# 讀取感測資料後

- Sensor的數值一直飄動, 可以怎麼處理? -> 試試filter
  1. Low-pass filter
  2. Kalman filter
  3. Complementary filter



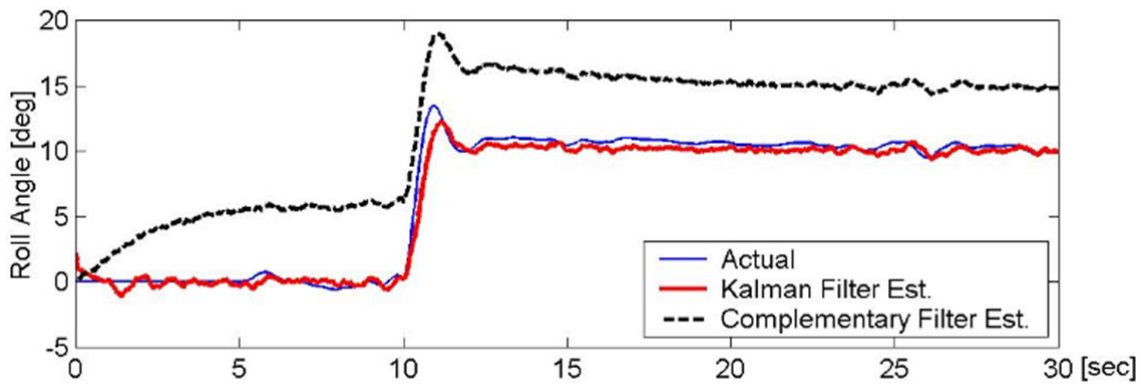




Fig source: https://github.com/snowpuppy/augreality/wiki/Alec-green-notebook
Fig source: http://www.nws.noaa.gov/om/csd/pds/PCU2/statistics/Stats/part2/Filter_LP.htm

# (1) Low-pass Filter

- 低通濾波器的功能是使低頻訊號通過，讓高頻訊號衰減，適用於高頻雜訊之過濾。
  - 例如溫度或流量感測器，訊號變化頻率低，即可用低通濾波器來除去如馬達產生的電氣雜訊。

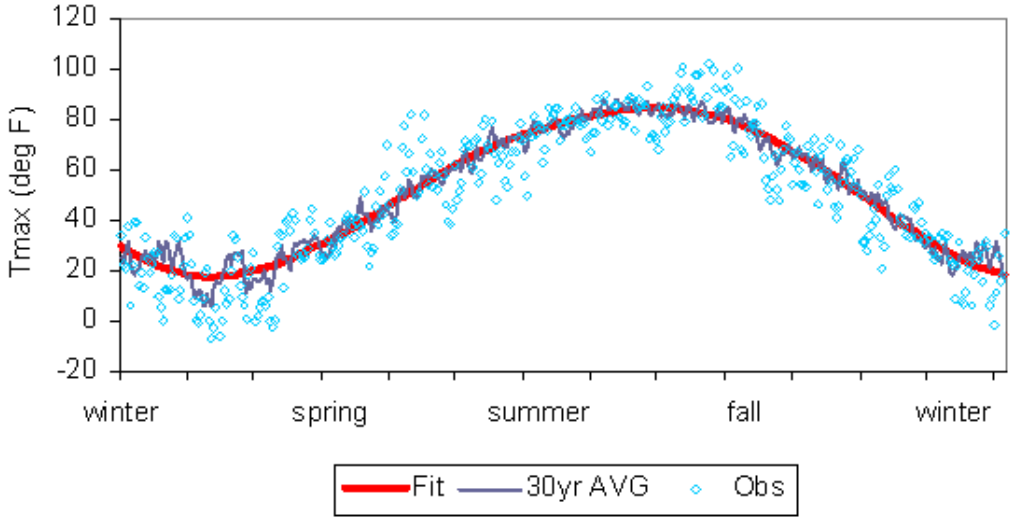



Fig source: http://www.nws.noaa.gov/om/csd/pds/PCU2/statistics/Stats/part2/Filter_LP.htm

http://designer.mech.yzu.edu.tw/articlesystem/article/compressedfile/(2002-09-24)%20主動式濾波器簡介.pdf

# (1) Low-pass Filter

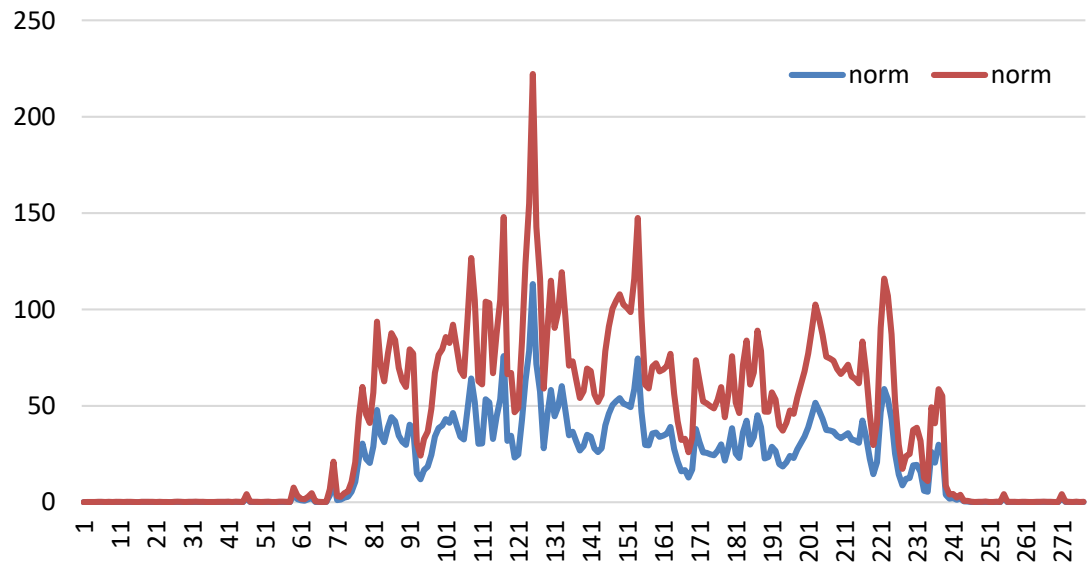
- Pseudocode algorithm
  - simulates the effect of a low-pass filter on a series of digital samples

```
// Return RC low-pass filter output samples, given input samples,
// time interval dt, and time constant RC
function lowpass(real[0..n] x, real dt, real RC)
  var real[0..n] y
  var real α := dt / (RC + dt)
  y[0] := x[0]
  for i from 1 to n
     y[i] := α * x[i] + (1-α) * y[i-1]
  return y
```

https://en.wikipedia.org/wiki/Low-pass_filter

# (1) Low-pass Filter

- 結合sensor數值與low pass filter?
  - y[i] := α * x[i] + (1-α) * y[i-1]　　α: smoothing factor, 0 < α < 1
  - y: 過濾後的資料
  - x: 原始資料(raw data)
- **Quiz3-1: 試著用low-pass filter分析sensor數值**
  - 可用excel觀察套用filter前後的變化



# (2) Kalman filter

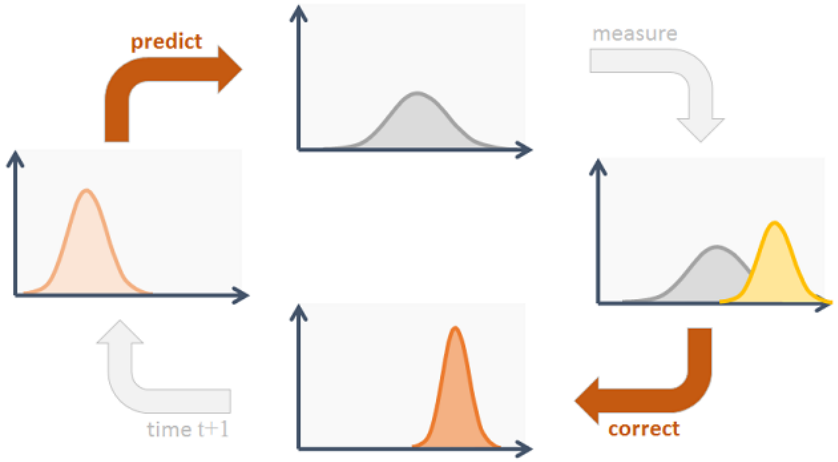
- Kalman Filter 的原理在於利用現在的狀態資訊加上下一秒的輸入，推斷出下一時刻的輸出
- Kalman Filter 利用高斯分佈(Gaussian Distribution)，在經過其遞迴的演算法則來對外在感測之誤差做有效的估測與融合。
  - 常用於機器人定位相關的研究與實驗
  - Example: 2D Kalman Tracking
    - https://www.youtube.com/watch?v=LSoJXCQIAvQ

*A Kalman filter was used to assist with navigation in the Apollo 11 moon landing.*

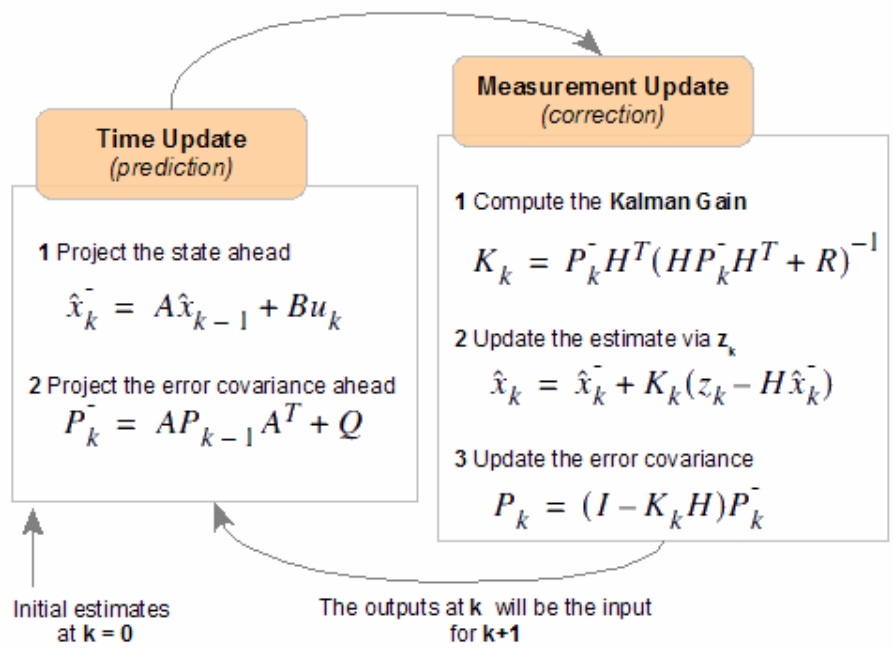
https://stanford.edu/class/ee363/lectures/kf.pdf
http://www.iceira.ntu.edu.tw/project-plans/200-iii

# (2) Kalman filter

- 預測 -> 更新 -> …




The cycle of a Kalman filter.




https://www.codeproject.com/Articles/865935/Object-Tracking-Kalman-Filter-with-Ease
http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies

# (2) Kalman filter

**Predict** [ edit ]

| | |
|---|---|
| Predicted (*a priori*) state estimate | $\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1\|k-1} + \mathbf{B}_k \mathbf{u}_k$ |
| Predicted (*a priori*) estimate covariance | $\mathbf{P}_{k\|k-1} = \mathbf{F}_k \mathbf{P}_{k-1\|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$ |

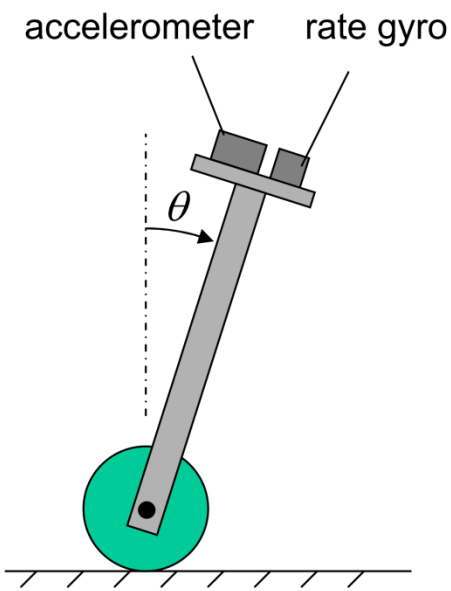
**Update** [ edit ]

| | |
|---|---|
| Innovation or measurement residual | $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k\|k-1}$ |
| Innovation (or residual) covariance | $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k\|k-1} \mathbf{H}_k^T + \mathbf{R}_k$ |
| *Optimal* Kalman gain | $\mathbf{K}_k = \mathbf{P}_{k\|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$ |
| Updated (*a posteriori*) state estimate | $\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$ |
| Updated (*a posteriori*) estimate covariance | $\mathbf{P}_{k\|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k\|k-1}$ |

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

# (3) Complementary filter

- Accel 數值長期是穩定，短期內卻可能振盪劇烈
- Gyro 值正好相反，短期內可靠，長時間卻會飄移
  - 可以用complementary filter做處理

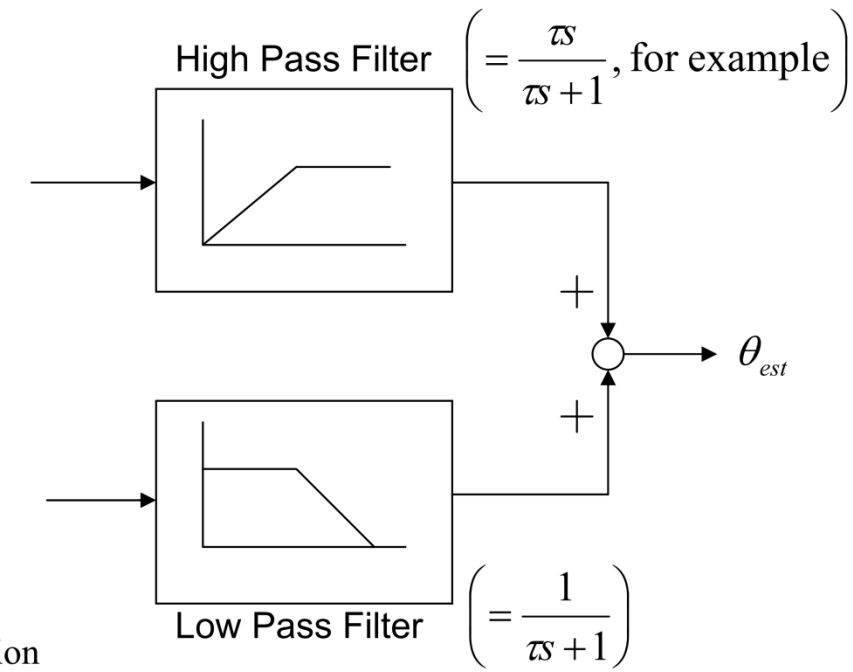



$$\theta \approx \int (\text{angular rate})\, dt$$

- not good in long term due to integration

$$\theta \approx \sin^{-1}\left(\frac{\text{accel. output}}{g}\right)$$

- only good in long term
- not proper during fast motion




https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-333-aircraft-stability-and-control-fall-2004/lecture-notes/lecture_15.pdf

# (3) Complementary filter

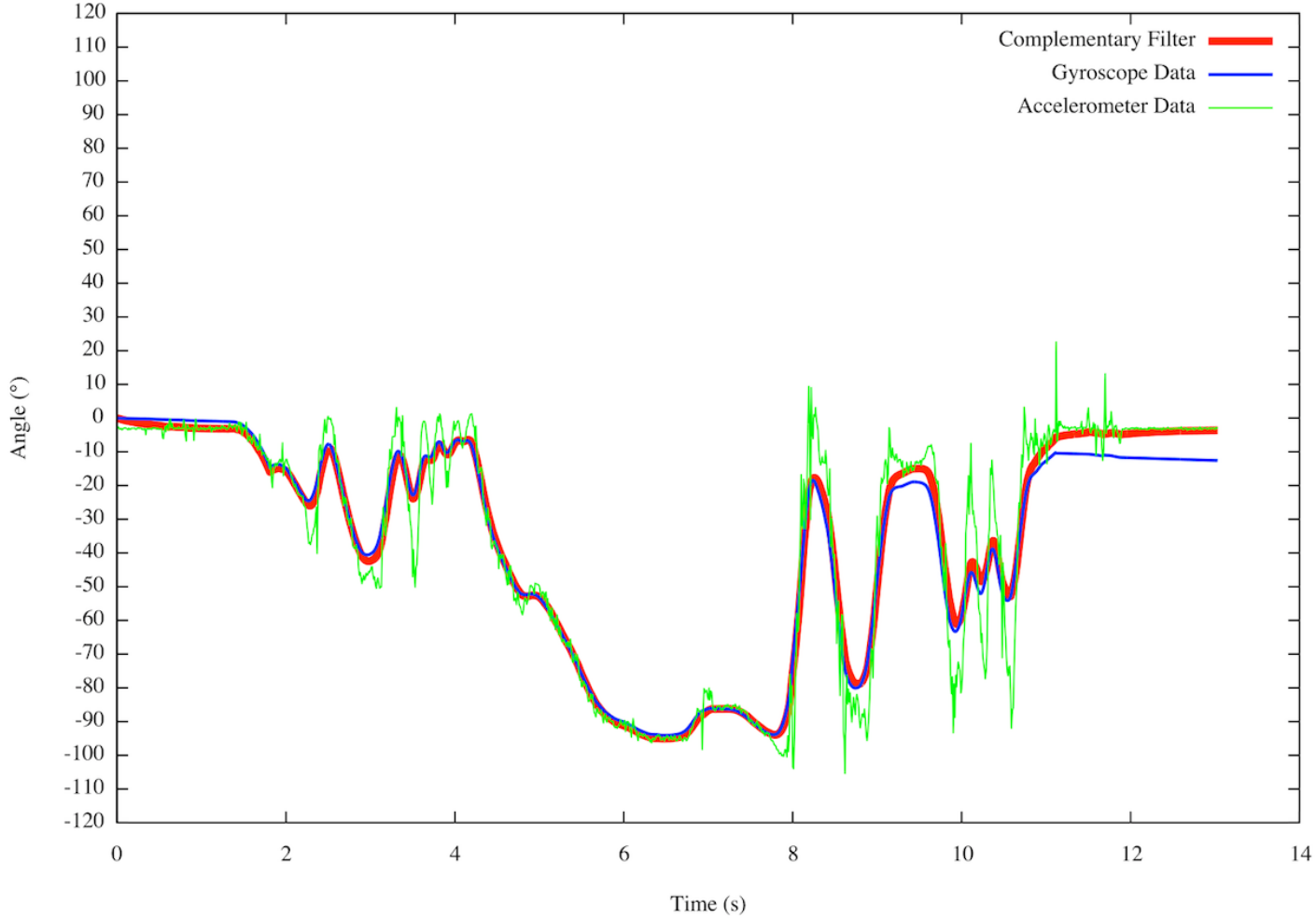



Fig source: http://www.pieter-jan.com/node/11

# BerryIMU

- The code currently performs angle measurements using the gyroscope and accelerometer, which are fused using a **complementary filter** and **kalman filter**.
  - 利用加速度取得角度資訊, 搭配 陀螺儀 與 前一回的角度資訊 進行資料融合

```
AccXangle =  (math.atan2(ACCy,ACCz)+M_PI)*RAD_TO_DEG
CFangleX=AA*(CFangleX+rate_gyr_x*LP) +(1 - AA) * AccXangle
```

- Source code:
  - https://github.com/mwilliams03/BerryIMU/blob/master/python-BerryIMU-gryo-accel-compass/berryIMU.py

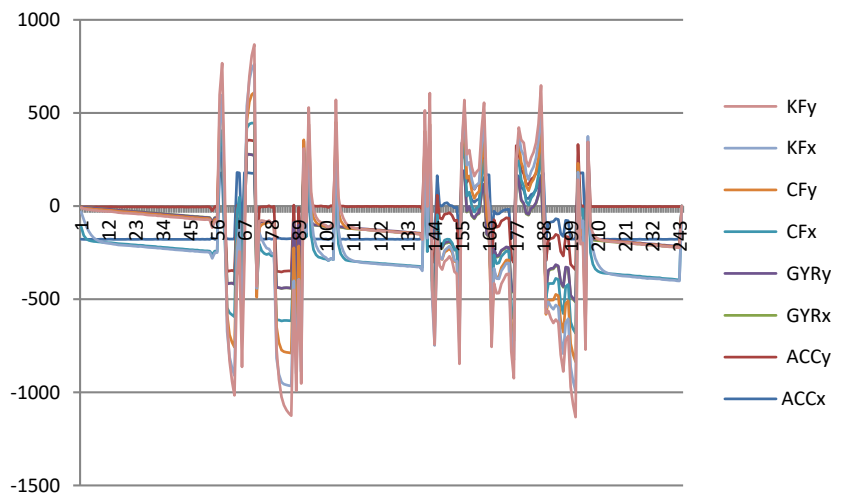
http://ozzmaker.com/berryimu-quick-start-guide/

# BerryIMU

- 下載source code (Kalman & Complementary filter)

```
wget https://raw.githubusercontent.com/mwilliams03/BerryIMU/master/python-BerryIMU-gryo-accel-compass/berryIMU.py
```

  - 此python code是基於LSM9DS0(9軸sensor), 無法直接使用
  - 將Q2整合的感測資料(10軸), 輸入至此sample code裡面
- **Quiz3-2: 觀察kalman filter與complementary filter之輸出資料, 並畫成圖表**



# Summary

- Practice Lab (氣壓計, Filter sample code)

- Write code for Quiz 1 - 3, then demonstrate it to TAs
  - Quiz1: 測量目前的氣壓計資訊, 目前的高度是多少? 覺得合理嗎
  - Quiz2: 整合目前的code, 讓python可以同時顯示加速度+陀螺儀+電子羅盤+氣壓計
  - Quiz3-1: 試著用low-pass filter分析sensor數值
  - Quiz3-2: 觀察kalman filter與complementary filter之輸出資料, 並畫成圖表

# 4.2 PI camera

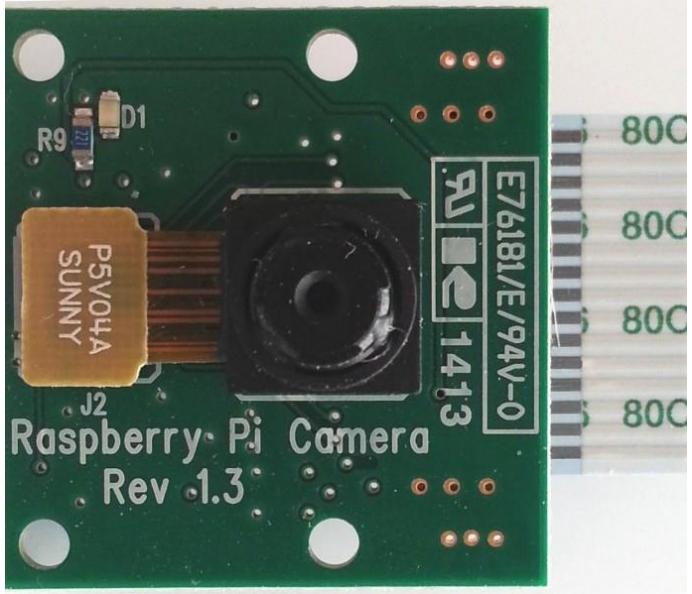
## I/O of Raspberry PI, analyzing data

https://www.slideshare.net/raspberrypi-tw/raspberry-pi-camera-python-opencv-day1

**National Chiao Tung University**

# PI camera

- Type of Raspberry Pi Camera (15-Pins, CSI介面)




Raspberry Pi Camera Module

NoIR Camera Module

http://elinux.org/Rpi_Camera_Module

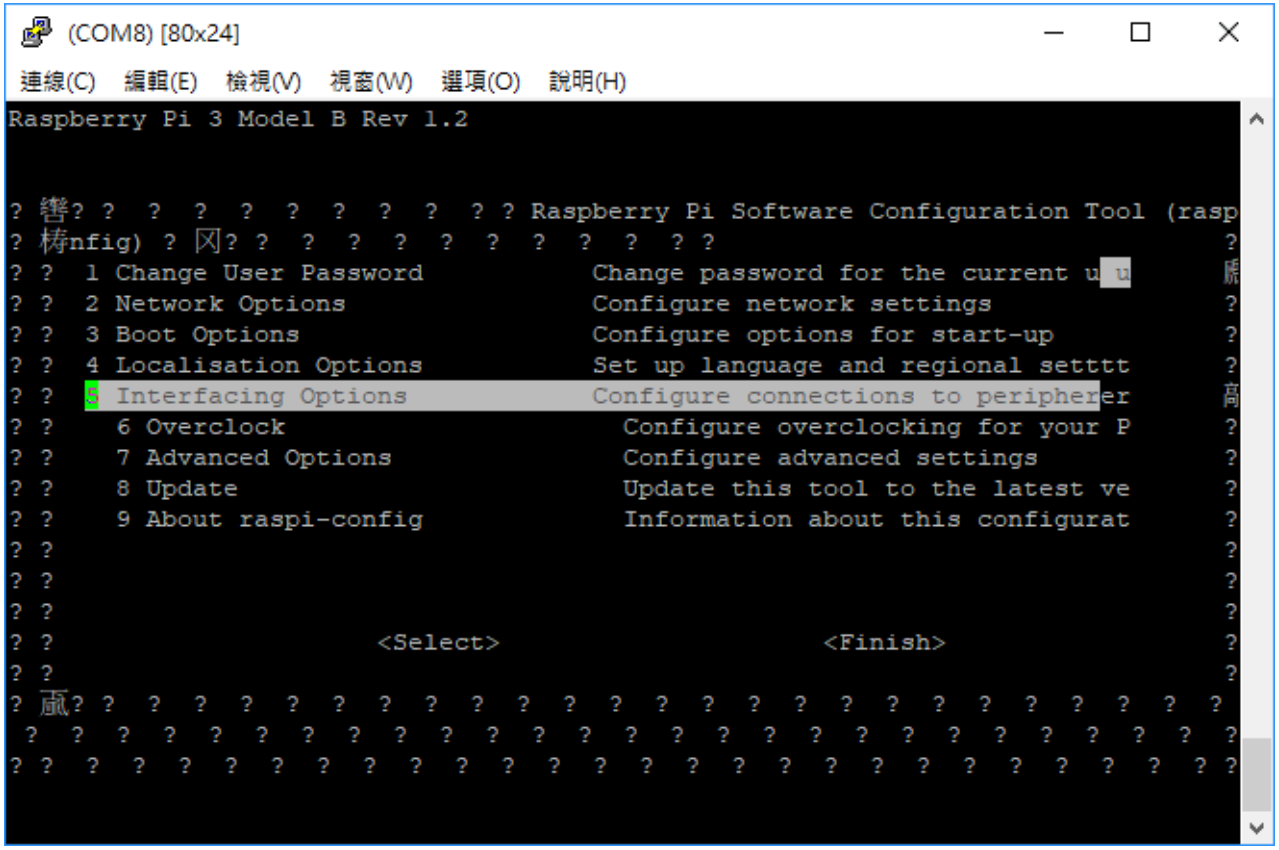
# PI camera spec.

- Sensor: OmniVision OV5647 Color CMOS QSXGA (5M)
- 靜態拍照最高解析度：2592 x 1944 pixel
- Pixel Size: 1.4 x 1.4 um
- Lens: f=3.6 mm, f/2.9
- Angle of View: 54 x 41 degrees
- Field of View: 2.0 x 1.33 m at 2 m
- Fixed Focus: 1m to infinity
- 動態攝影最高解析度：1080p@30 FPS with H.264/AVC

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

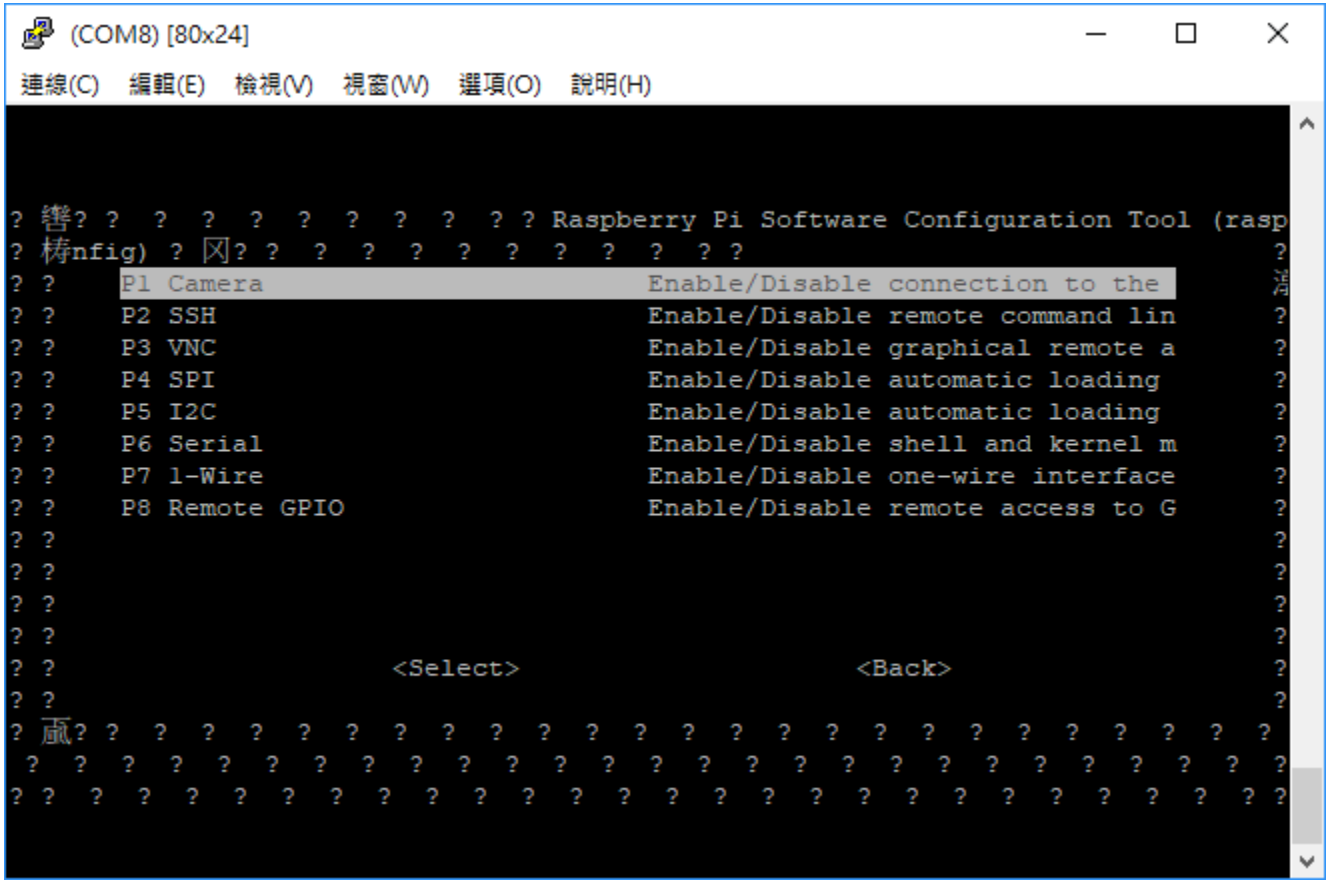
# Enable PI camera

- 在關機的狀態下安裝PI Camera (接上排線)
- 在終端機執行 sudo raspi-config



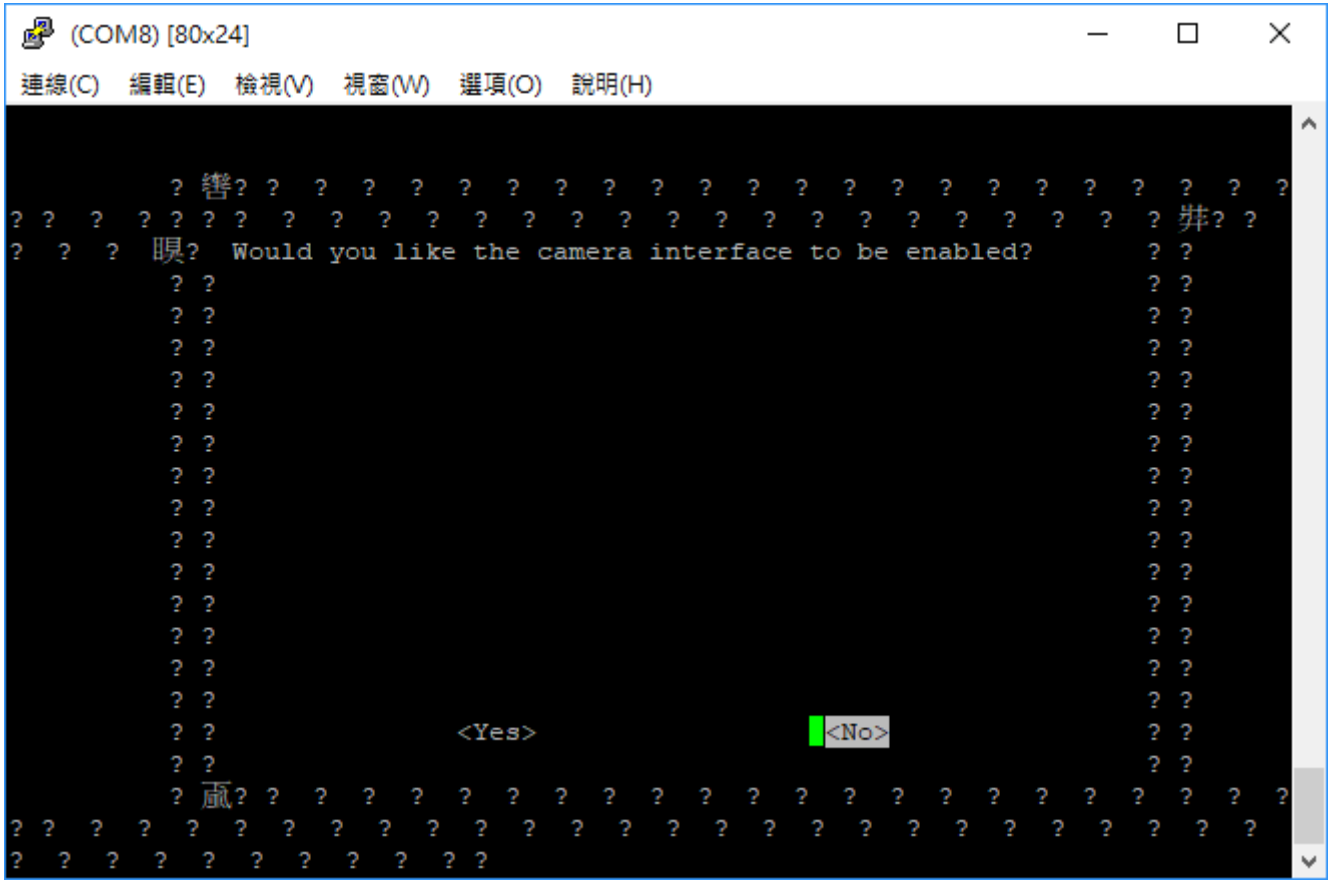
# Enable PI camera

- 啟用PI camera



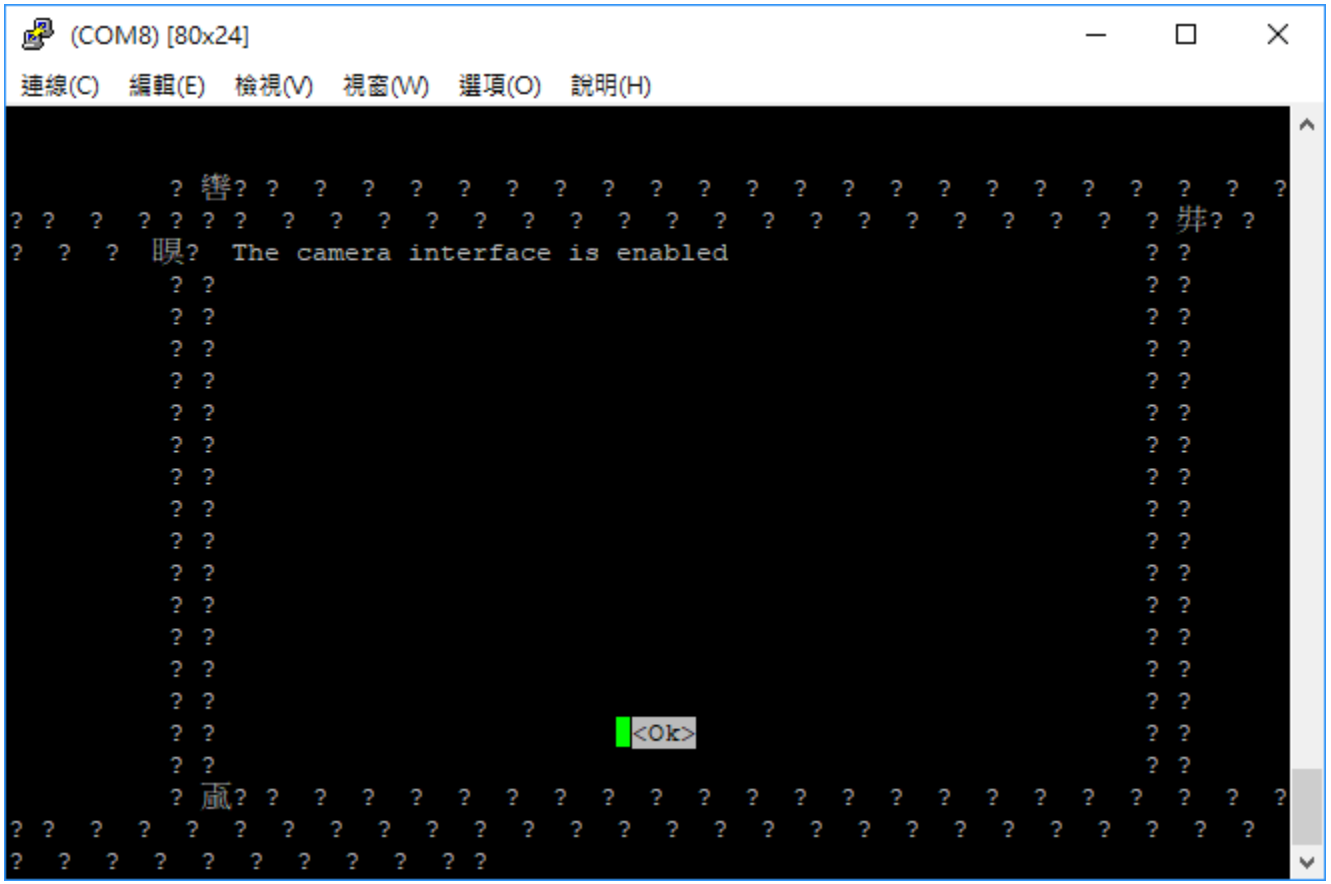
# Enable PI camera

- 啟用PI camera



# Enable PI camera

- 啟用PI camera



# Enable PI camera

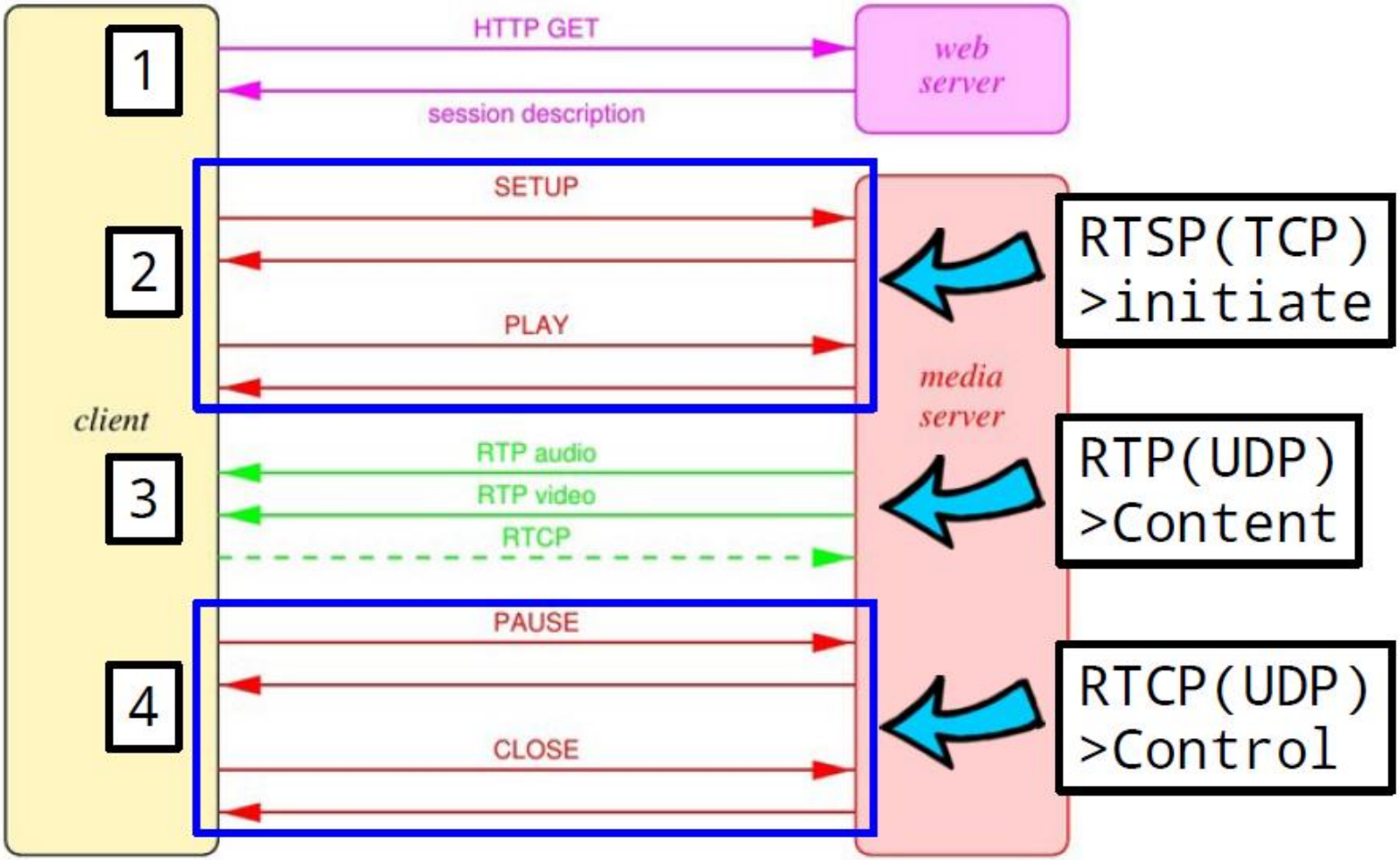
- 啟用PI camera

# Camera 應用

- 拍照/錄影測試 (**需要用圖形介面!!**)
  - 只預覽2秒(-t),不存檔
    - raspistill -t 2000
  - 3秒後拍照,並編碼成png格式(-e),長640x寬480
    - raspistill -t 3000 -o test.png -e png -w 640 -h 480
  - 錄5秒的1080p30影片, 長640x寬480
    - raspivid -t 5000 -w 640 -h 480 -o video.h264

- IP Camera
  - 自行架設網路攝影機, 使用 RTSP + H.264

# RTSP protocol




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

# RTSP Server

- 在Raspberry Pi執行下列指令
  - raspivid -o - -t 0 -hf -w 320 –h 240 -fps 15 | cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554}' :demux=h264

- 電腦/手機端可用VLC觀看畫面
  - rtsp://192.168.1.48:8554/

最後面的/不要忘記

# 其他camera的應用教學

- https://www.slideshare.net/raspberrypi-tw/raspberry-pi-camera-python-opencv-day1
- 影像串流
  - 使用 RTSP + H.264
  - 使用 HTTP + MJPG

- https://www.slideshare.net/raspberrypi-tw/raspberry-pi-camera-and-opencv-day2
- 用Python + OpenCV做影像處理
- 機器學習應用與綜合練習
  - 人臉偵測
  - 圖形分類

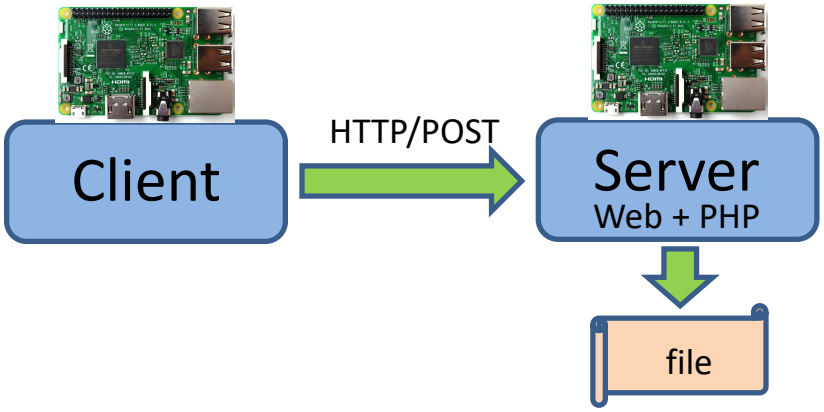
- https://github.com/CiscoBlockChain/YOLO-Pi
  - Real Time Object Detection on Raspberry Pi using YOLO

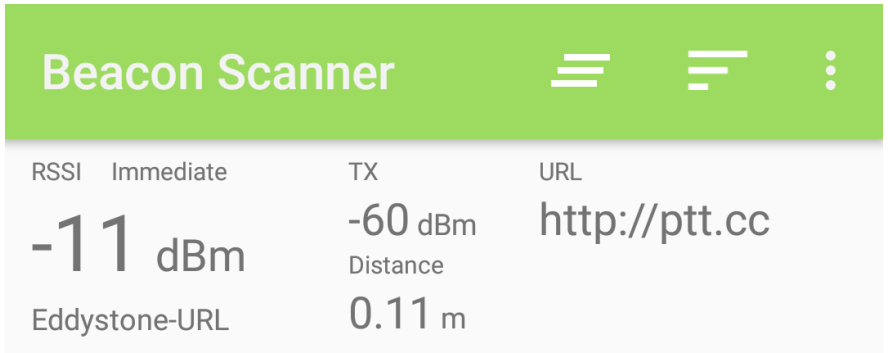
# 5. Networking and Communications

National Chiao Tung University

# Outline

- 1. 架設web server
  - 提供php範例, 利用HTTP/POST送資料至server端
- 2. BLE 實驗
  - 修正PI3的藍牙功能
  - 藍牙BLE介紹
  - 實作BLE廣告



# Web Server

- Web Server:
  - 軟體(LAMP)
    - Linux – Raspbian
    - Apache – webserver (http) software
    - MySQL – database server
    - PHP or Perl or Python – Web Programming

# 建立 Web Server

- 安裝網頁伺服器apache2:
  - sudo apt-get install apache2

```
pi@raspberrypi ~ $ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version.
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libdrm-nouveau2 libdrm-radeon1 libelf1 libllvm3.5
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

# 建立 Web Server 網站

- 安裝PHP程式語言:
  - sudo apt-get install php5 libapache2-mod-php5

```
pi@raspberrypi ~ $ sudo apt-get install php5 libapache2-mod-php5
Reading package lists... Done
Building dependency tree
Reading state information... Done
libapache2-mod-php5 is already the newest version.
php5 is already the newest version.
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libdrm-nouveau2 libdrm-radeon1 libelf1 libllvm3.5
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

- 重新啟動服務 (Web Server):
  - sudo service apache2 restart

```
pi@raspberrypi ~ $ sudo service apache2 restart
pi@raspberrypi ~ $
```

# 建立 Web Server 網站

- 確認IP位置&開網頁測試:
  - ifconfig

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:01:1b
          inet6 addr: fe80::8ce:da36:d61c:fce/64 Sco
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:425 errors:0 dropped:0 overruns:0 frame:0
          TX packets:425 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:944288 (922.1 KiB)  TX bytes:944288 (922.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 80:1f:02:fd:ad:65
          inet addr:192.168.1.101  Bcast:255.255.255.255  Mask:255.255.255.0
          inet6 addr: fe80::b115:3431:56a1:d164/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:197 errors:0 dropped:34 overruns:0 frame:0
          TX packets:236 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24995 (24.4 KiB)  TX bytes:28644 (27.9 KiB)

pi@raspberrypi ~ $
```

# 建立 Web Server 網站

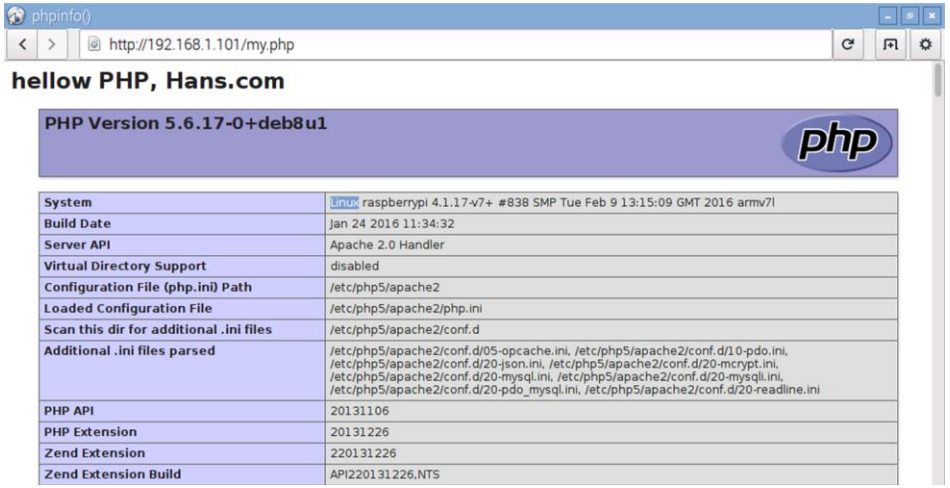
- 撰寫簡單的PHP程式:
  - sudo nano /var/www/html/my.php

```
<h1>hellow PHP, XD </h1>
<?php
   phpinfo();
?>
```

程式解說:
第1行: 使用標準HTML顯示一段文字
第2,4行: 範圍內的文字是php程式語言
第3行: 用來顯示機器上安裝php程式語言的相關資料

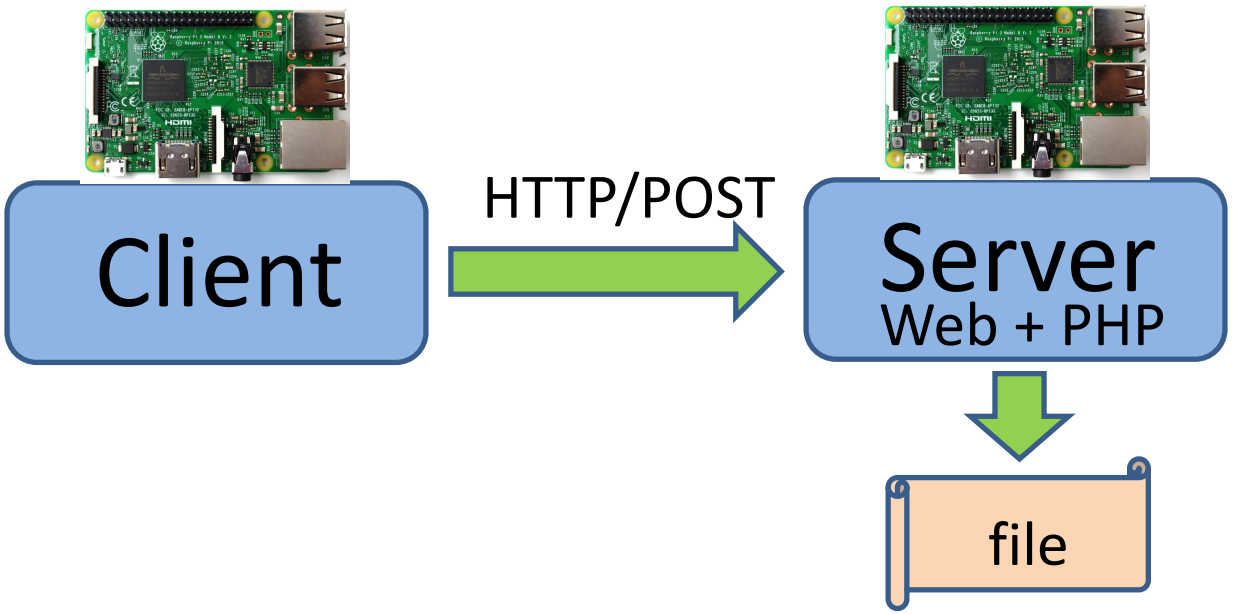
開啟網頁的畫面

http://192.168.1.101/my.php



# Web Server & HTTP/POST

- 結合php語法, 利用HTTP/POST傳送資料並存檔



# HTTP protocol

- HTTP
  - 超文字傳輸協定, HyperText Transfer Protocol
  - HTTP/1.0: define the GET, POST and HEAD methods
  - HTTP/1.1: add 5 new methods: OPTIONS, PUT, DELETE, TRACE and CONNECT.
- Two commonly methods for a request-response
  - GET: Requests data from a specified resource
  - POST: Submits data to be processed to a specified resource (ex: 送出表單資料)

https://www.w3schools.com/tags/ref_httpmethods.asp

# HTTP protocol

- HTTP/GET
  - only retrieve data and should have no other effect
- HTTP/POST
  - requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI
  - often used when uploading a file or when submitting a completed web form

Name: **Gareth Wylie**
Age: **24**
Formula: **a + b == 13%!**

the key-value pairs are encoded as follows

Name=**Gareth+Wylie**&Age=**24**&Formula=**a+%2B+b+%3D%3D+13%25%21**

https://en.wikipedia.org/wiki/POST_(HTTP)

# Web Server & HTTP/POST

- 在網頁根目錄放置一個 index.php 的檔案, 內容如下
  - 根目錄 /var/www/html/

```
<?php
        header("Content-Type:text/html; charset=utf-8");
        /* Temp -> server */
        // ex: curl -d "sensor=1&Temp=28.9" http://IP_addr
        $Temperature=$_POST[Temp];
        $SensorID=$_POST[sensor];

        echo 'Temperature:'.$Temperature. endl;

if ($SensorID==1) {
        $fp = fopen('/home/pi/www-data/temp.txt', 'w');
        fwrite($fp, $Temperature);
        fclose($fp);
}
?>
```

送來的Temp與Sensor值,
存到Temperature與SensorID

當SensorID為1
將Temperature數值存成該檔案

# Web Server & HTTP/POST

- 實驗相關指令

```
cd ~
mkdir www-data
chown www-data:pi www-data/

cd /var/www/html
mv index.htm index2.htm
nano index.php

ifconfig
```

先在家目錄建立資料夾 (與PHP裡面的路徑一致)
設定該資料夾權限, 使web server具有存取能力

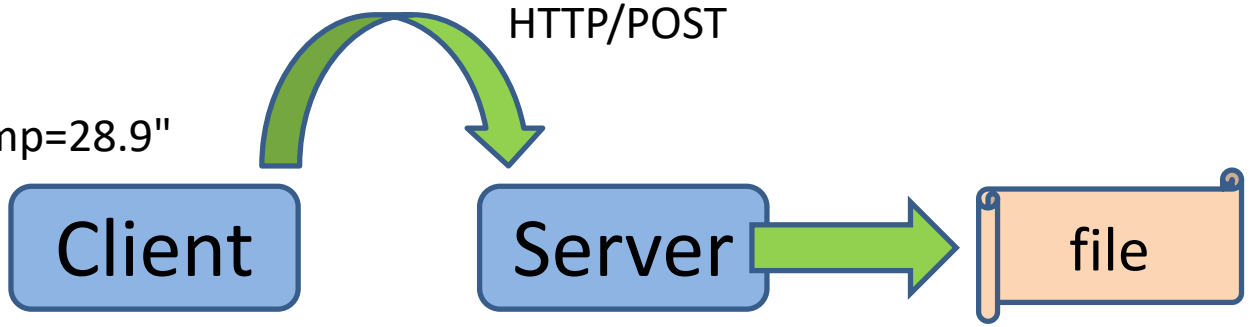
進入網頁根目錄
將原本的首頁改檔名
新建index.php檔案, 內容如前頁

使用ifconfig檢查目前使用IP

# Web Server & HTTP/POST

- 自己傳送post資料給自己
  - curl -d "sensor=1&Temp=28.9" http://127.0.0.1

curl -d "sensor=1&Temp=28.9"
http://127.0.0.1

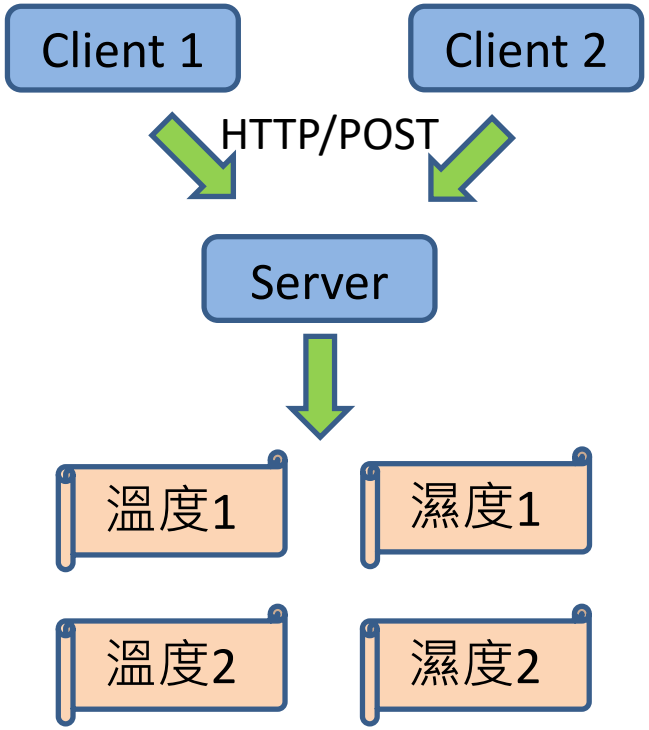


# Web Server & HTTP/POST

- Quiz1: 同時有兩個client傳送各2筆資料, 修改php使server將資料存成不同檔案 (Ex: client 1 & 2同時傳送溫濕度資訊)

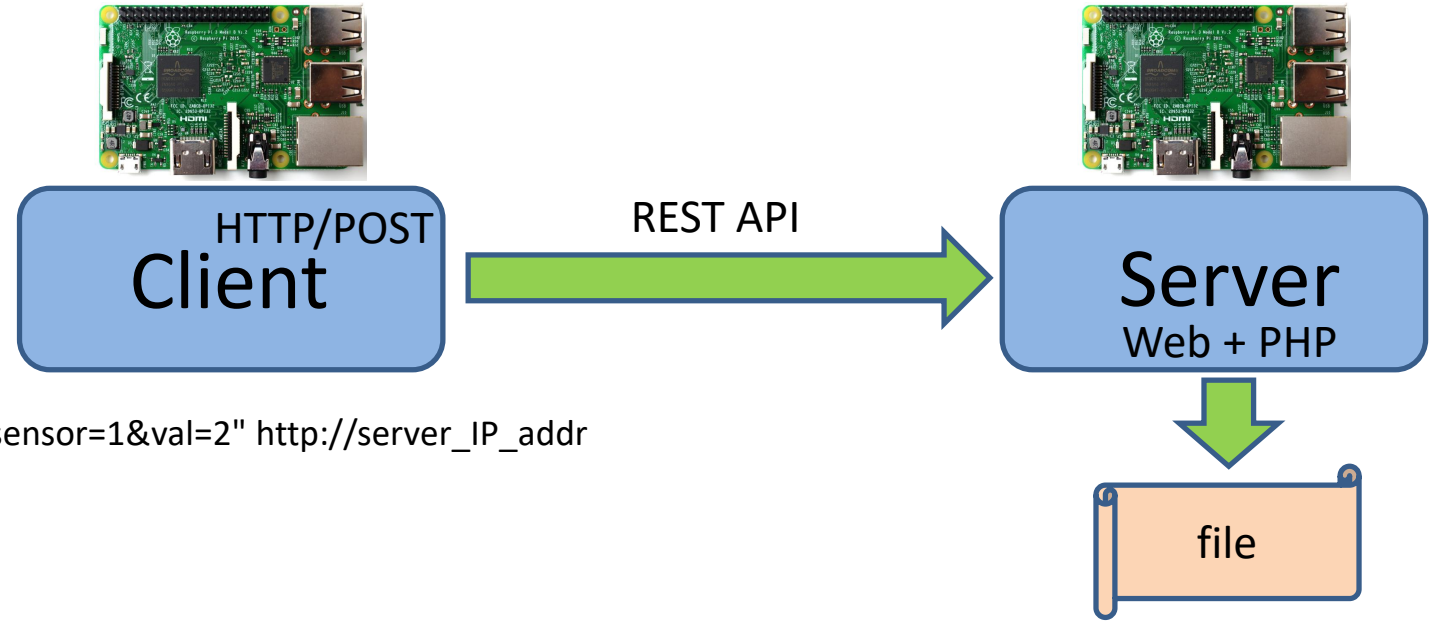
curl -d "sensor=1&Temp=28.9&humi=60" http://127.0.0.1

```
<?php
        header("Content-Type:text/html; charset=utf-8");
        /* Temp -> server */
        // ex: curl -d "sensor=1&Temp=28.9" http://IP_addr
        $Temperature=$_POST[Temp];
        $SensorID=$_POST[sensor];
        echo 'Temperature:'.$Temperature. endl;

if ($SensorID==1) {
        $fp = fopen('/home/pi/www-data/temp.txt', 'w');
        fwrite($fp, $Temperature);
        fclose($fp);
}
?>
```



# Web Server & HTTP/POST

- Quiz2: 使用兩個PI, 其中 Client 傳送資料 至 Server
  - 一人當client, 一人當server
  - Server需開啟檔案確認有成功接收到資料




curl -d "sensor=1&val=2" http://server_IP_addr

# Bonus

- 基於Quiz2 (使用兩個PI), Client 安裝溫濕度sensor, 將實際的溫濕度資訊傳送 至 Server

載入 subprocess

```
import subprocess

server = 'http://127.0.0.1/recv_acc.php'
data = 'sensor=1&X={0:.4f}&Y={1:.4f}'.format(x, y)
subprocess.call(['curl','-d', data , server ],shell=False);
```

傳送下列指令:

curl -d "sensor=1&X=0.1234&Y=1.2345" http://127.0.0.1/recv_acc.php

# 2. 藍牙 Bluetooth

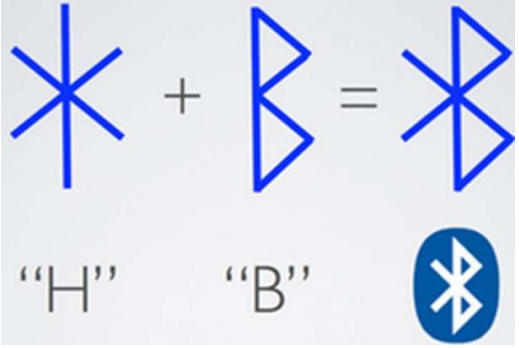
- 目的
  - 為了解決電腦與電器設備之間的傳輸問題
- 特色
  - 短距離無線技術 (10 - 100m)
  - 使用 2.4 至 2.485 GHz 的 ISM 頻段
- Bluetooth Classic: 802.15
- Bluetooth 4.0 Low Energy (BLE): 802.15.1

https://zh.wikipedia.org/zh-tw/%E8%97%8D%E7%89%99

# 藍牙起源

- 歷史
  - 十世紀國王的名字 (Harald Blåtand)
    - 統一了因宗教戰爭和領土爭議而分裂的挪威與丹麥而聞名於世
    - 喜歡吃藍莓，因此牙齒都變成藍色 (Blue tooth)
    - 另一說，他的牙齒很差，看起來像藍色(blue, dark, black)
    - 他喜歡穿藍色的服飾，當時的藍色有昂貴、尊爵、不凡的意思
  - 由 Ericsson 在 1994 年創製 ,希望為裝置間的通訊創造一組統一規則（標準化協定），以解決用戶間互不相容的移動電子裝置

不要寫成藍**芽**喔!




https://zh.wikipedia.org/zh-tw/%E8%97%8D%E7%89%99
https://en.wikipedia.org/wiki/Harald_Bluetooth

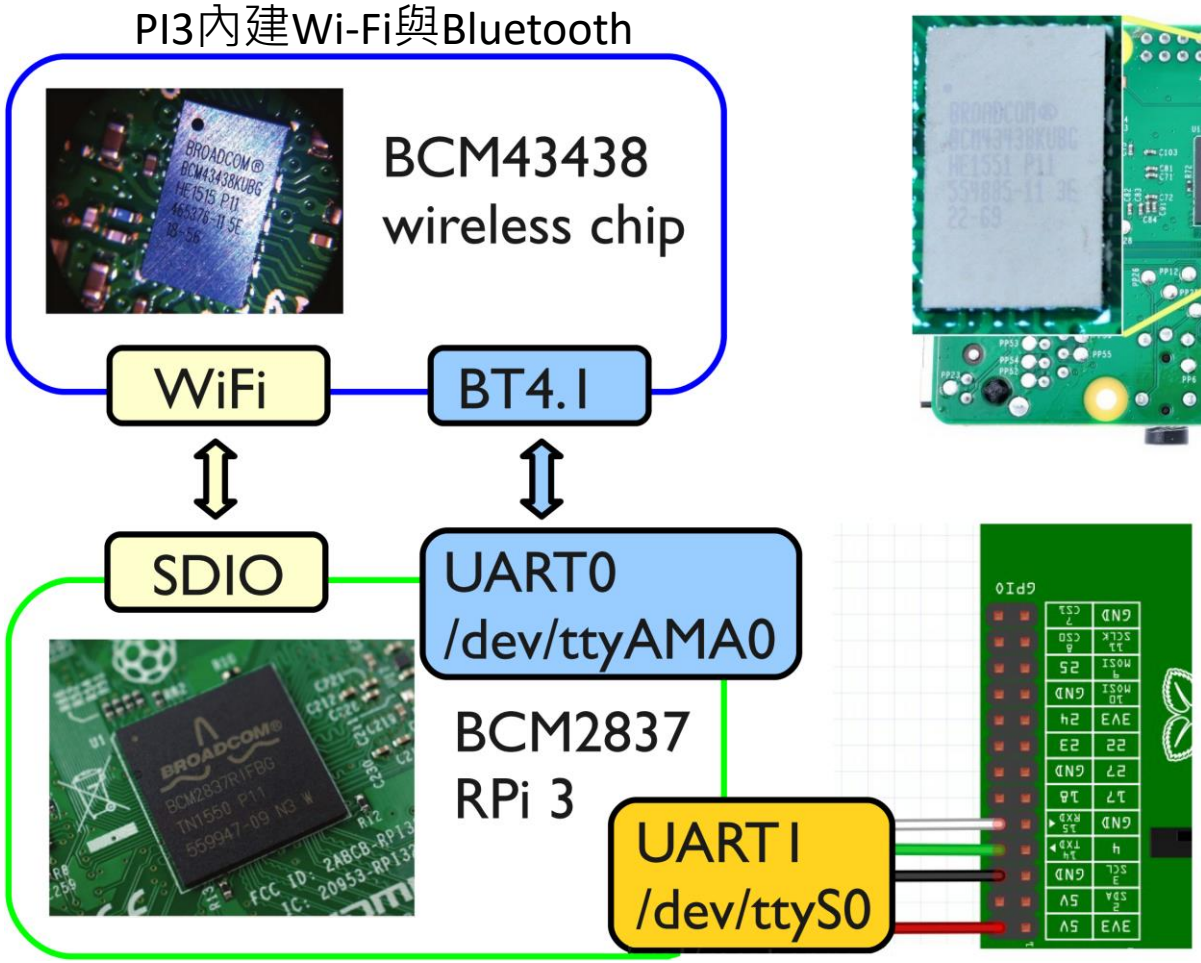
# Bluetooth Low Energy (BLE)

- 一種無線個人區域網路 (Wireless PAN) 的技術
- 出現目的 : 低成本 , 低耗電 (CR2032 電池可用 1 年 )
- BT4 分 Classic(BR/EDR), High Speed(HS), Low Energy

| | Classic | BLE |
|---|---|---|
| Throughput | 2 ~ 3 Mbps | 0.2 Mbps |
| Range | 50 ~ 300 m | 10 ~ 30 m |
| Power consumption | 1 W | 0.01 ~ 0.5 W |
| Connection time | 5 s | 0.1 s |

Bluetooth™ 4.0 Low Energy

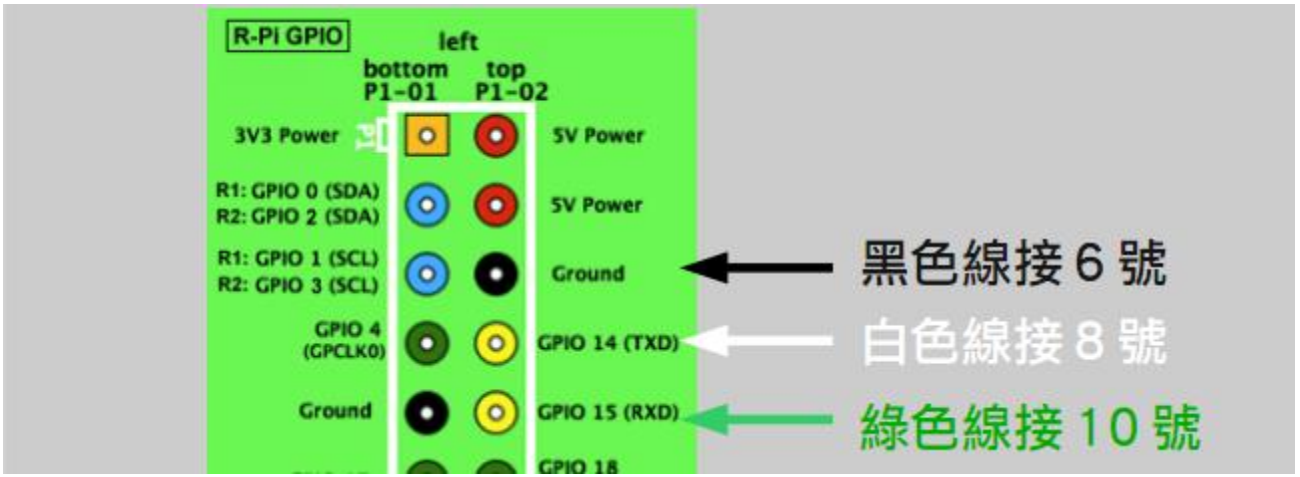
# Hardware Architecture of Pi 3



http://www.slideshare.net/yeokm1/raspberry-pi-3-uartbluetooth-issues

# 誰會受影響？

- 在PI3上有**使用 TX/RX 腳位**的裝置
  - Serial debug console
  - RS232 模組, GSM 模組, GPS 模組
  - 裝置之間透過 UART 通訊
  - 內建藍牙裝置



# 解決方法

- 需求：
  - Full UART 和 Slow Bluetooth 或是 BLE
- 想法：
  - 使用 Device Tree Overlay(DTO) 重新映射腳位
- 作法
  - 修改開機設定檔: sudo nano /boot/config.txt
  - 重開機執行 hciconfig -a 查詢藍牙裝置

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

新增

# 解決方法

- 還是找不到藍牙裝置? -> 用USB的藍牙網卡
  - 連接裝置後, 執行 hciconfig -a 查詢

```
pi@raspberrypi:~$ hciconfig -a
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING
        RX bytes:887 acl:0 sco:0 events:39 errors:0
        TX bytes:951 acl:0 sco:0 commands:39 errors:0
        Features: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'raspberrypi'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.0 (0x6)  Revision: 0x22bb
        LMP Version: 4.0 (0x6)  Subversion: 0x22bb
        Manufacturer: Cambridge Silicon Radio (10)
```

# Bluetooth 常用工具

- bluetoothctl - bluetooth control tool
- hciconfig - configure Bluetooth devices
- hcitool - configure Bluetooth connections
- l2ping - Send L2CAP echo request and receive answer
- btmon - Bluetooth monitor
- gatttool - GATT tool

# 藍牙 + BLE

- 確認藍牙裝置是否有支援BLE功能
  - hciconfig -a hci0 features (尋找 **LE support**)

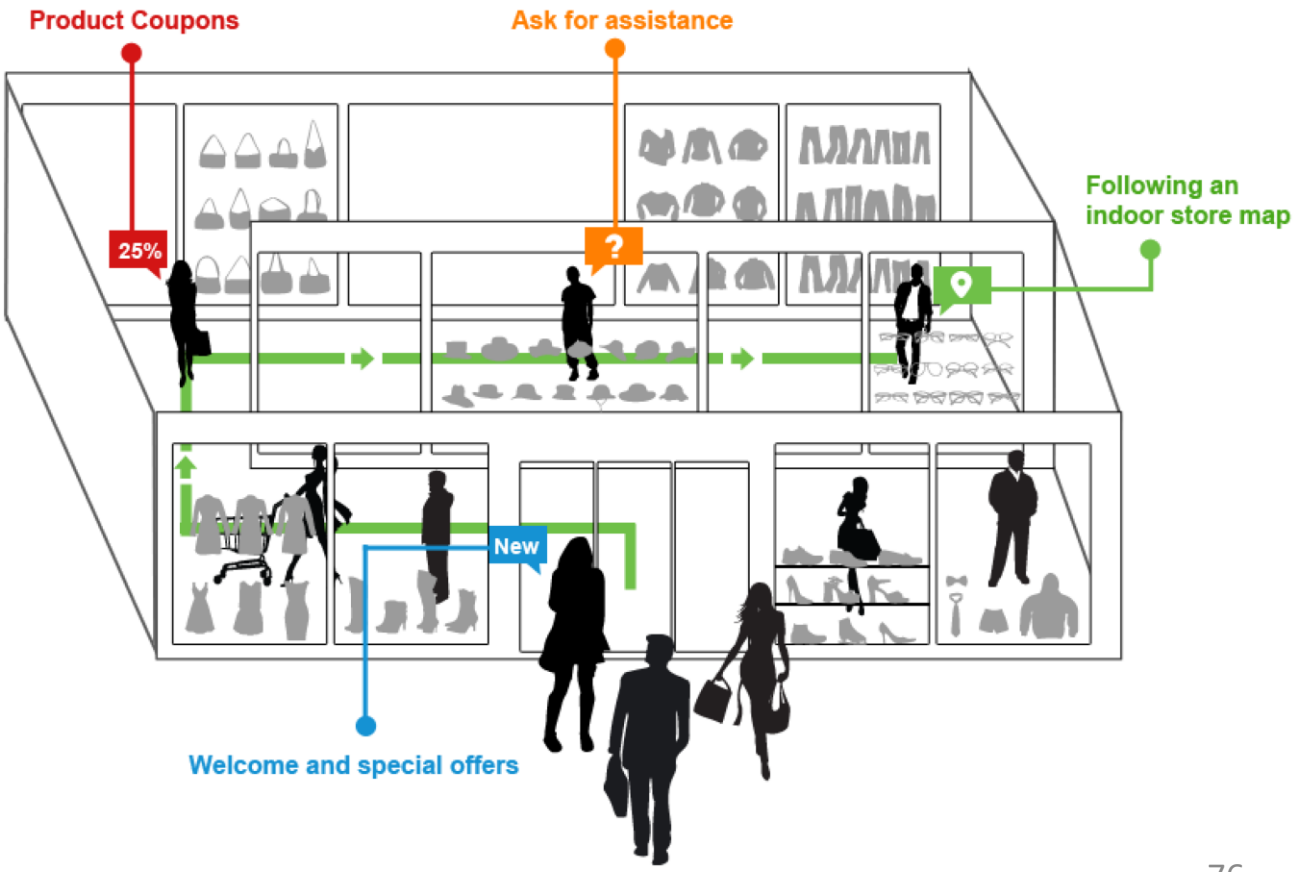
```
pi@raspberrypi:~$ hciconfig -a hci0 features
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 310:10  SCO MTU: 64:8
        Features page 0: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87
                <3-slot packets> <5-slot packets> <encryption> <slot offset>
                <timing accuracy> <role switch> <hold mode> <sniff mode>
                <park state> <RSSI> <channel quality> <SCO link> <HV2 packets>
                <HV3 packets> <u-law log> <A-law log> <CVSD> <paging scheme>
                <power control> <transparent SCO> <broadcast encrypt>
                <EDR ACL 2 Mbps> <EDR ACL 3 Mbps> <enhanced iscan>
                <interlaced iscan> <interlaced pscan> <inquiry with RSSI>
                <extended SCO> <EV4 packets> <EV5 packets> <AFH cap. slave>
                <AFH class. slave> <LE support> <3-slot EDR ACL>
                <5-slot EDR ACL> <sniff subrating> <pause encryption>
                <AFH cap. master> <AFH class. master> <EDR eSCO 2 Mbps>
                <EDR eSCO 3 Mbps> <3-slot EDR eSCO> <extended inquiry>
                <LE and BR/EDR> <simple pairing> <encapsulated PDU>
                <non-flush flag> <LSTO> <inquiry TX power> <EPC>
                <extended features>
        Features page 1: 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

# 支援 BLE 的平台

- iOS5+ (iOS7+ preferred)
- Android 4.3+ (numerous bug fixes in 4.4+)
- Apple OS X 10.6+
- Windows 8 (XP, Vista and 7 only support Bluetooth 2.1)
- GNU/Linux Vanilla BlueZ 4.93+

# BLE 的應用

- 微型定位服務
- 推播訊息




https://learn.adafruit.com/pibeacon-ibeacon-with-a-raspberry-pi/overview

# BLE 的產品

- 物品防丟

**產品規格**

藍牙版本：藍牙4.0

兼容設備：蘋果IOS5.0及以上版本系統和安卓4.3以上版本系統

工作距離：≤25m（無障礙環境）

電　　池：CR 2032鋰電池，可工作半年左右

# BLE的應用 – LINE beacon

LINE Developers

Log in

Products

Documents

Document top

LINE Login

LINE SDK for iOS

LINE SDK for Android

Social API v2.1

Messaging API

Top

Quickstart

Overview

Getting started

Building a sample bot

Building a bot

Guides

LINE Developers > Document top > Messaging API > Using beacons

## Using beacons

With ***LINE Beacon***, your bot can receive beacon webhook events whenever a LINE user enters or leaves the proximity of a beacon. Using beacons, you can customize your bot to interact with users in specific contexts.

★ Note: LINE Beacon is available in Japan, Taiwan, and Thailand.

Table of contents


Getting a beacon

Linking your bot with a beacon

Receiving webhook events

Beacon banner


### Getting a beacon

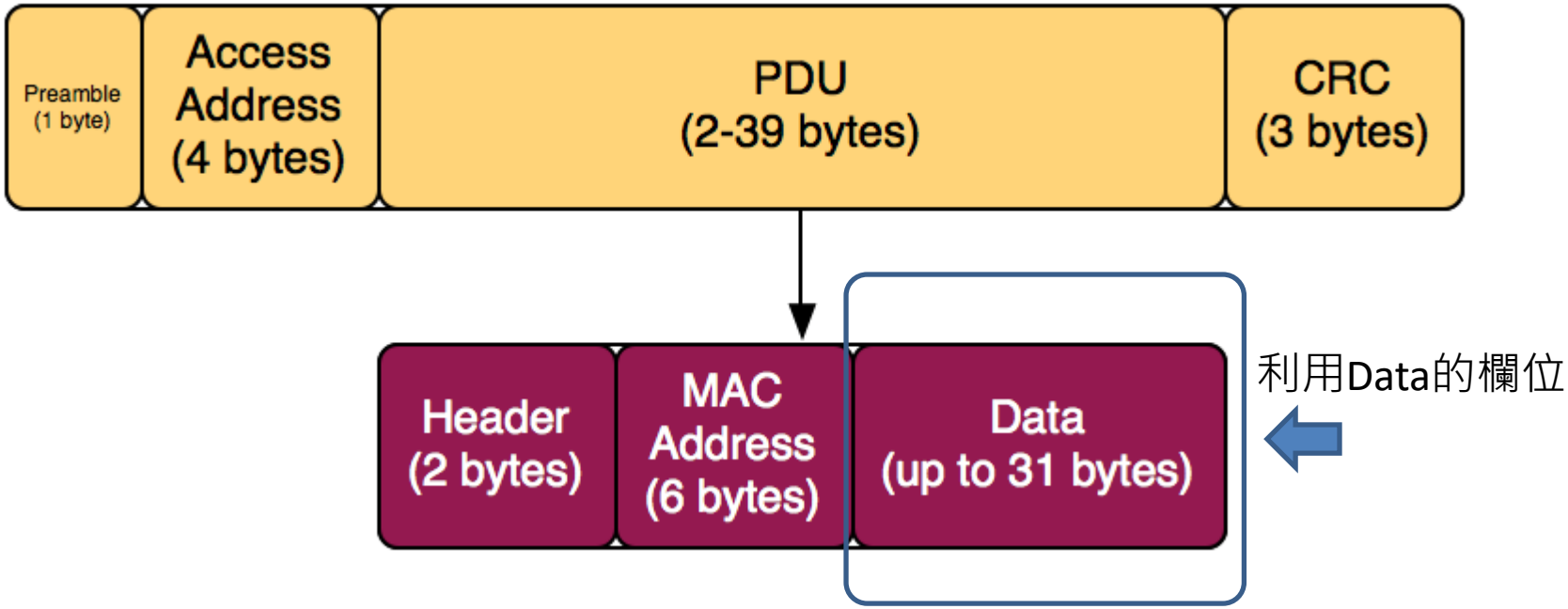
To use LINE Beacon, you'll first need a Bluetooth® Low Energy beacon to link with your bot. You can link one of the following types of devices with your bot.

- A beacon device that supports LINE Beacon. Supported devices are available here. Note: These devices are only supported in Japan.
- A Bluetooth Low Energy device that uses the LINE Simple Beacon specification.

https://developers.line.biz/en/docs/messaging-api/using-beacons/

# BLE frame format

- 1 byte preamble
- 4 byte access address
- 2-39 bytes advertising channel PDU
- 3 bytes CRC




http://www.warski.org/blog/2014/01/how-ibeacons-work/

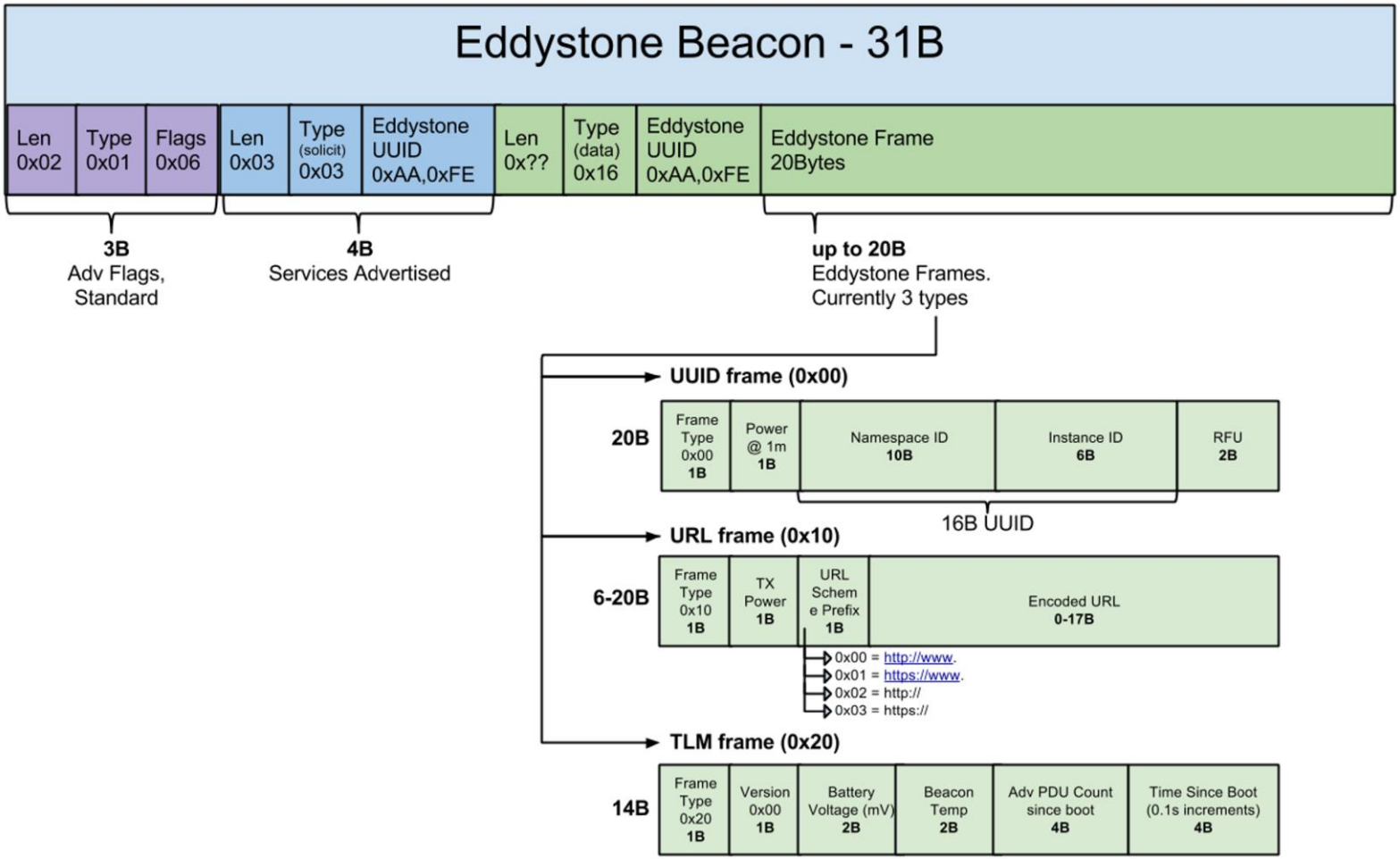
# BLE 的應用 (Eddystone)

- Eddystone is a protocol specification that defines a Bluetooth low energy (BLE) message format for proximity beacon messages.

- Design Goals
  - Works well with Android and iOS Bluetooth developer APIs
  - Straightforward implementation on a wide range of existing BLE devices
  - Flexible architecture permitting development of new frame types
  - Fully compliant with the Bluetooth Core Specification

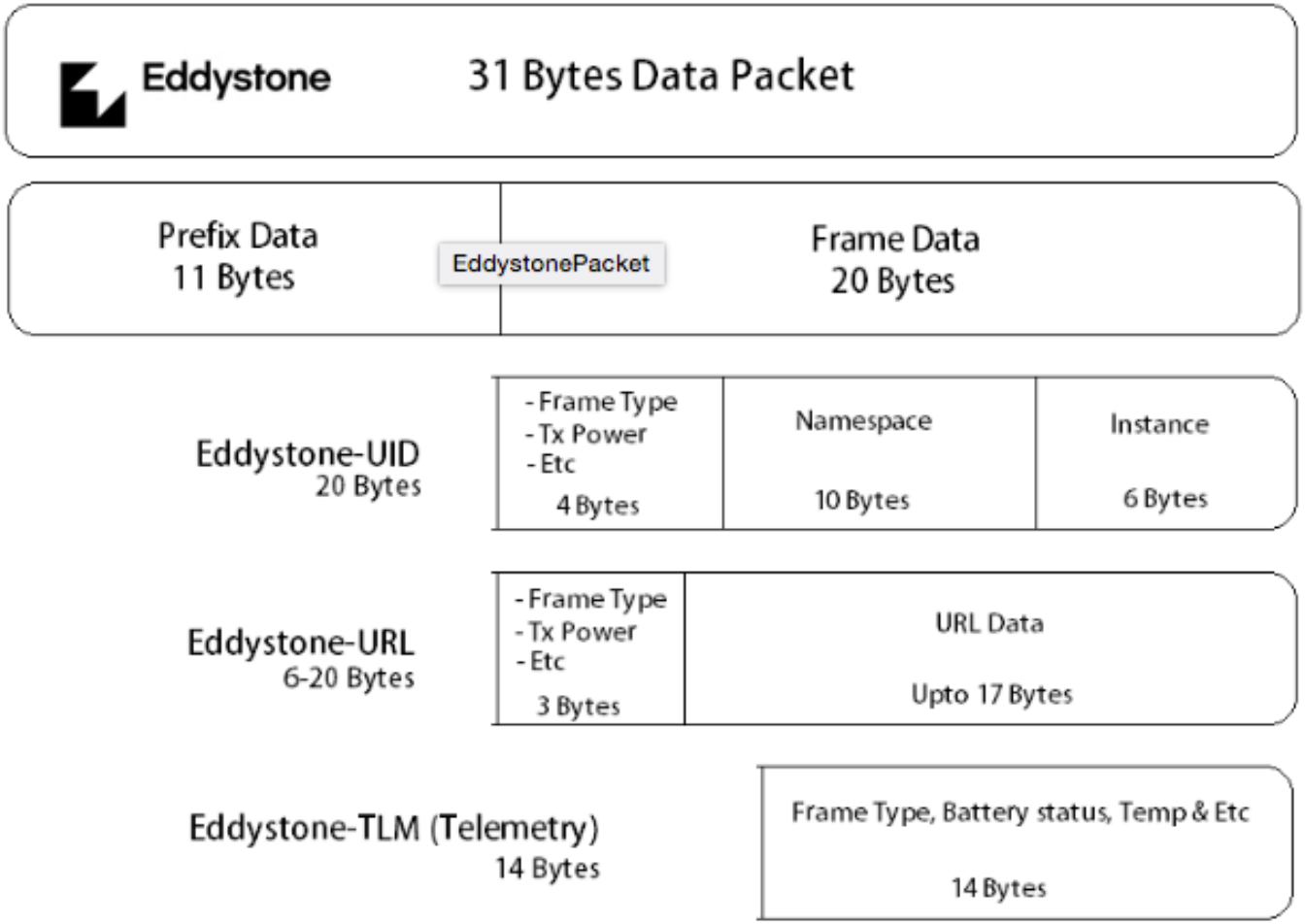
https://github.com/google/eddystone

# Eddystone Frame format




https://developer.mbed.org/teams/Bluetooth-Low-Energy/code/BLE_EddystoneBeacon_Service/file/dfb7fb5a971b/Eddystone.h

# Eddystone Frame format

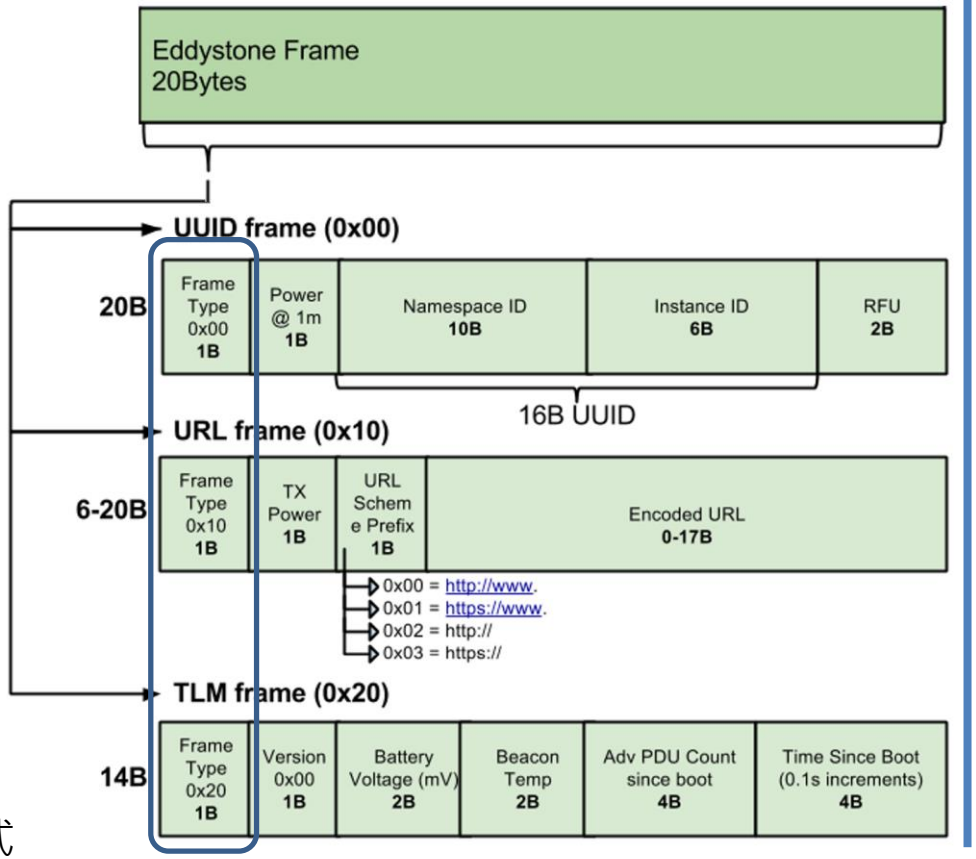



http://iosandi.blogspot.tw/2015/12/in-addition-to-my-previous-post-i-have.html

# Eddystone Frame format

- Eddystone Protocol Specification

| Frame Type | High-Order 4 bits | Byte Value |
|---|---|---|
| UID | 0000 | 0x00 |
| URL | 0001 | 0x10 |
| TLM | 0010 | 0x20 |
| EID | 0011 | 0x30 |
| RESERVED | 0100 | 0x40 |




ps. 這張圖沒有畫出Eddystone-EID frame的格式

https://github.com/google/eddystone/blob/master/protocol-specification.md

# Eddystone訊息

- Goal:
  - 利用PI的BLE, 產生Eddystone的廣告訊息
  - 依照網址建立一個符合Eddystone格式的frame

  - Eddystone網址格式
    - https://github.com/google/eddystone/tree/master/eddystone-url
  - ACSII table
    - https://zh.wikipedia.org/wiki/ASCII

# Eddystone訊息

- 將網址轉換為Eddystone格式
  - Ex: http://ptt.cc
  - 利用ASCII table查詢數值

| 數值 (16進位) | 網址 |
|---|---|
| 02 | http:// |
| 70 | p |
| 74 | t |
| 74 | t |
| 2e | . |
| 63 | c |
| 63 | c |

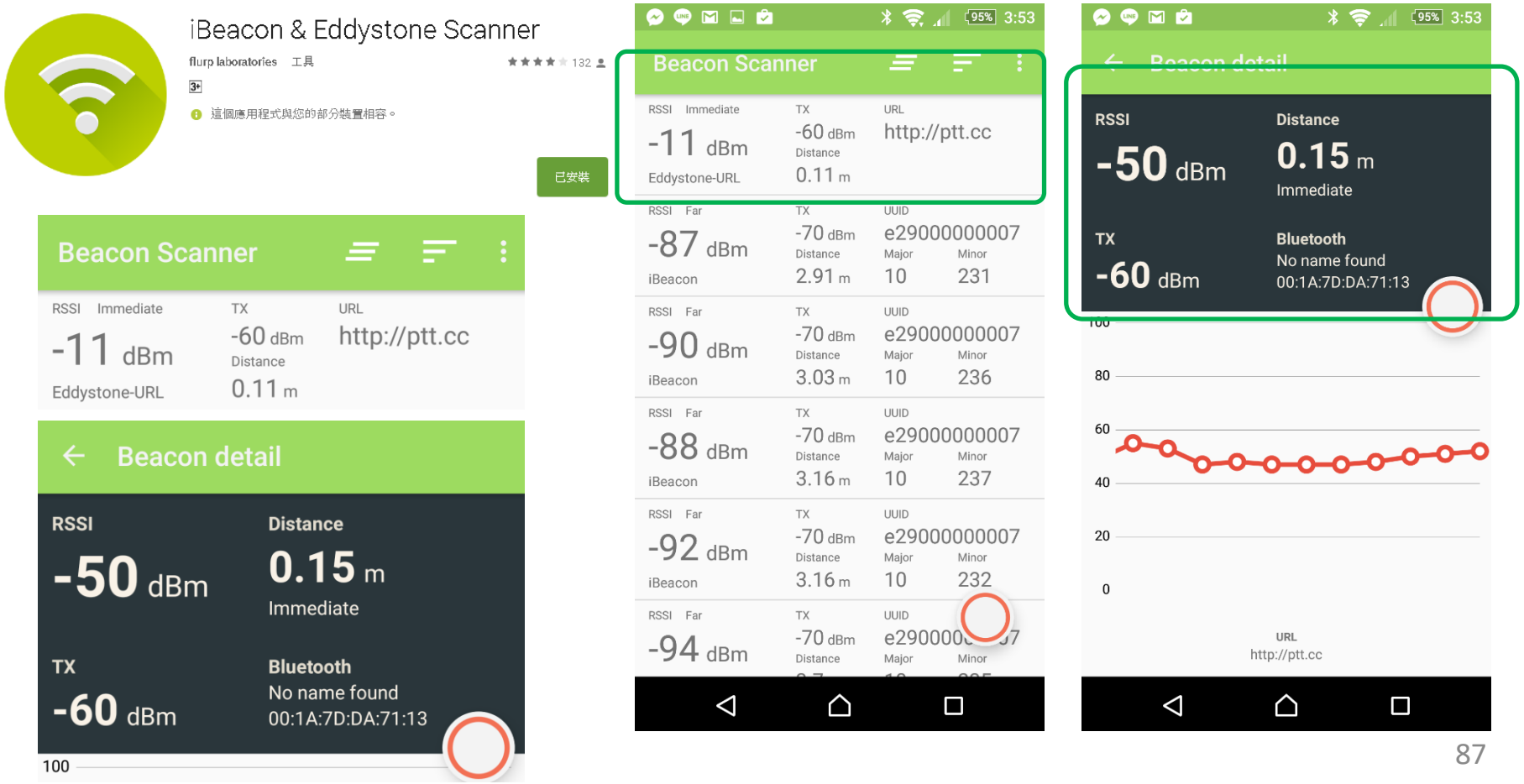
| Decimal | Hex | Expansion |
|---|---|---|
| 0 | 0x00 | http://www. |
| 1 | 0x01 | https://www. |
| 2 | 0x02 | http:// |
| 3 | 0x03 | https:// |

# Eddystone訊息

- 使用 advertise-url 來傳送網址廣播
  - Source code
    - https://github.com/google/eddystone/blob/master/eddystone-url/implementations/linux/advertise-url
  - 下載後
    - chmod +x advertise-url (新增執行權限)
    - sudo ./advertise-url -u http://ptt.cc (開始廣播)
    - sudo ./advertise-url –s (停止廣播)

# Eddystone訊息

- 手機端可安裝app查看Eddystone訊息



# Eddystone訊息

- 也可以使用bluetooth工具傳送網址廣播

啟用藍牙的低耗能廣告(LE advertising)模式，並關閉掃描功能

```
sudo hciconfig hci0 leadv 3
sudo hciconfig hci0 noscan
```

```
sudo hcitool -i hci0 cmd 0x08 0x0008 14 02 01 1a 03 03 aa fe 0c 16
aa fe 10 ed 02 70 74 74 2e 63 63 00 00 00 00 00 00 00 00 00 00 00
```

http://ptt.cc

傳送Eddystone frame

```
sudo hciconfig hci0 noleadv
```

停止廣告

# Eddystone訊息

sudo hcitool -i hci0 cmd 0x08 0x0008 14 02 01 1a 03 03 aa fe 0c 16 aa fe 10 ed **02 70 74 74 2e 63 63** 00 00 00 00 00 00 00 00 00 00 00

- 0x08 0x0008:  set the ad package
  - #OGF = Operation Group Field = Bluetooth Command Group = 0x08
  - #OCF = Operation Command Field = HCI_LE_Set_Advertising_Data = 0x0008
- 14: the ENTIRE following data packet in bytes (16進位的14 = 20 byte)

20 byte

- 02 01 1a:  Eddystone Adv Flags
  - 0x06 - The device is BLE only. The full Bluetooth stack is not supported.
  - 0x1A - The device can be used as BLE as well as full Bluetooth Controller/Host simultaneously.
- 03 03 aa fe: Eddystone service adv
- 0c: length (12 byte)
- 16: type (data)
- aa fe: Eddystone UUID
- 10: URL frame type
- ed: TX power
- 02 70 74 74 2e 63 63: http://ptt.cc,共 7 byte

12 byte

- 00 00 00 00 00 00 00 00 00 00 00: 共 10 byte

# Eddystone訊息

- Quiz3:
  - 產生Eddystone的廣告訊息, 網址為google首頁
    - https://www.google.com
  - 網址轉換後的數值?
  - 手機抓取Eddystone截圖

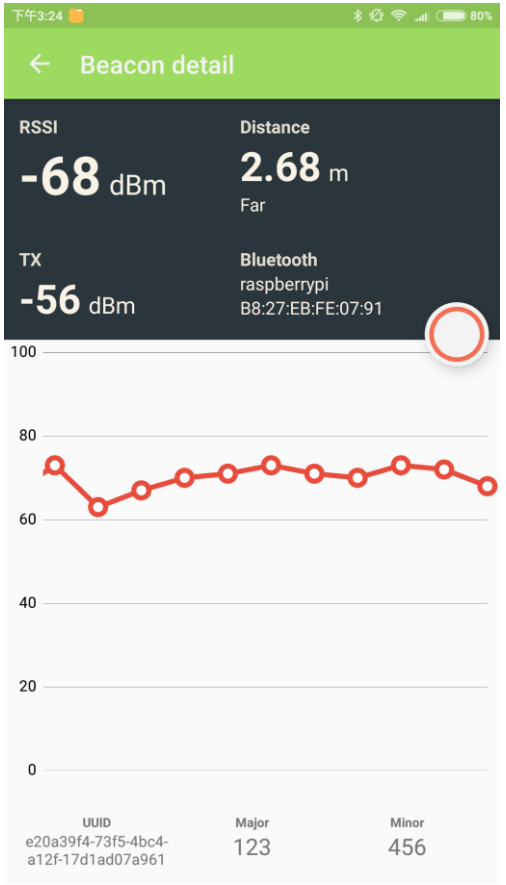
| 數值 (16進位) | 網址 |
|---|---|
| ??? | ??? |
| ??? | ??? |
| ??? | ??? |
| ??? | ??? |
| ??? | ??? |
| ??? | ??? |
| ??? | ??? |

# 補充: iBeacon

- Python script that creates an iBeacon-compatible Bluetooth LE beacon using Linux and a Bluetooth LE adapter
  - https://github.com/dburr/linux-ibeacon

使用方式:

**sudo python ibeacon **
**--uuid=E20A39F473F54BC4A12F17D1AD07A961 **
**--major=123 --minor=456**



# Summary

- Practice Lab (PHP與HTTP/POST, BLE實驗)

- Write code for Quiz 1 - 3, then demonstrate it to TAs
  - Quiz1: 同時傳送2筆資料, 修改php將資料存成不同檔案
  - Quiz2: 使用兩個PI, 其中 Client 傳送資料 至 Server
  - Quiz3: 產生Eddystone的廣告訊息, 網址為google首頁
  - Bonus: 在Quiz2傳送真實的感測資訊

# 5. Communicate with Arduino

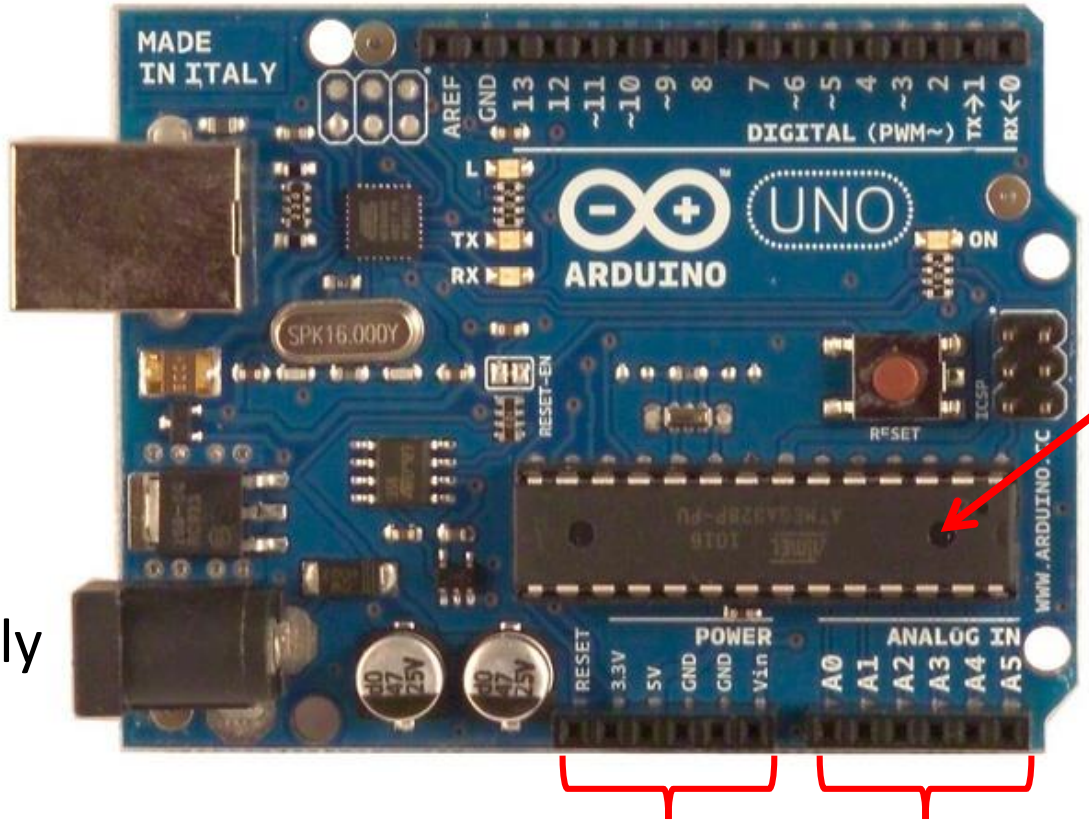
National Chiao Tung University

# Outline

- 1. Arduino
  - Introduction
  - Analog read
  - Digital read
- 2. Communication Methods
  - USB console (Arduino IDE)
  - Wi-Fi
  - I2C
  - TTL

# Arduino intro.

## Arduino UNO




Digital In/Out Pins

USB Port

Atmega328p

Power Supply

Power Pins

Analog Input Pins

# Arduino UNO Spec

**AVR Arduino microcontroller**

- Microcontroller — ATmega328P
- Operating Voltage — 5V
- Digital I/O Pins — 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins — 6
- Analog Input Pins — 6
- DC Current per I/O Pin — 20 mA
- DC Current for 3.3V Pin — 50 mA
- Flash Memory — 32 KB (ATmega328P)
  - which 0.5 KB used by bootloader
- Clock Speed — 16 MHz
- LED_BUILTIN — 13

# Before using Arduino…

## Download the Arduino IDE

**ARDUINO 1.8.5**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

**Windows** Installer
**Windows** ZIP file for non admin install

**Windows app** Get

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits
**Linux** 64 bits
**Linux** ARM

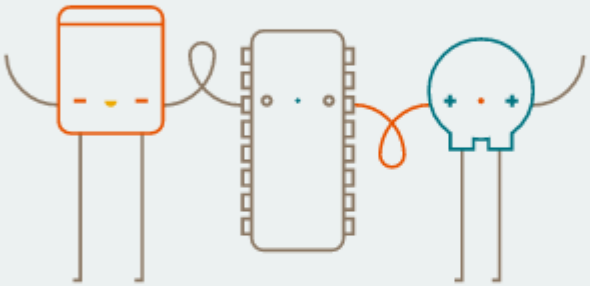
Release Notes
Source Code
Checksums (sha512)

**[Highly recommend] Bring your laptop to this class.**

# Download the Arduino Software (IDE)

## Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



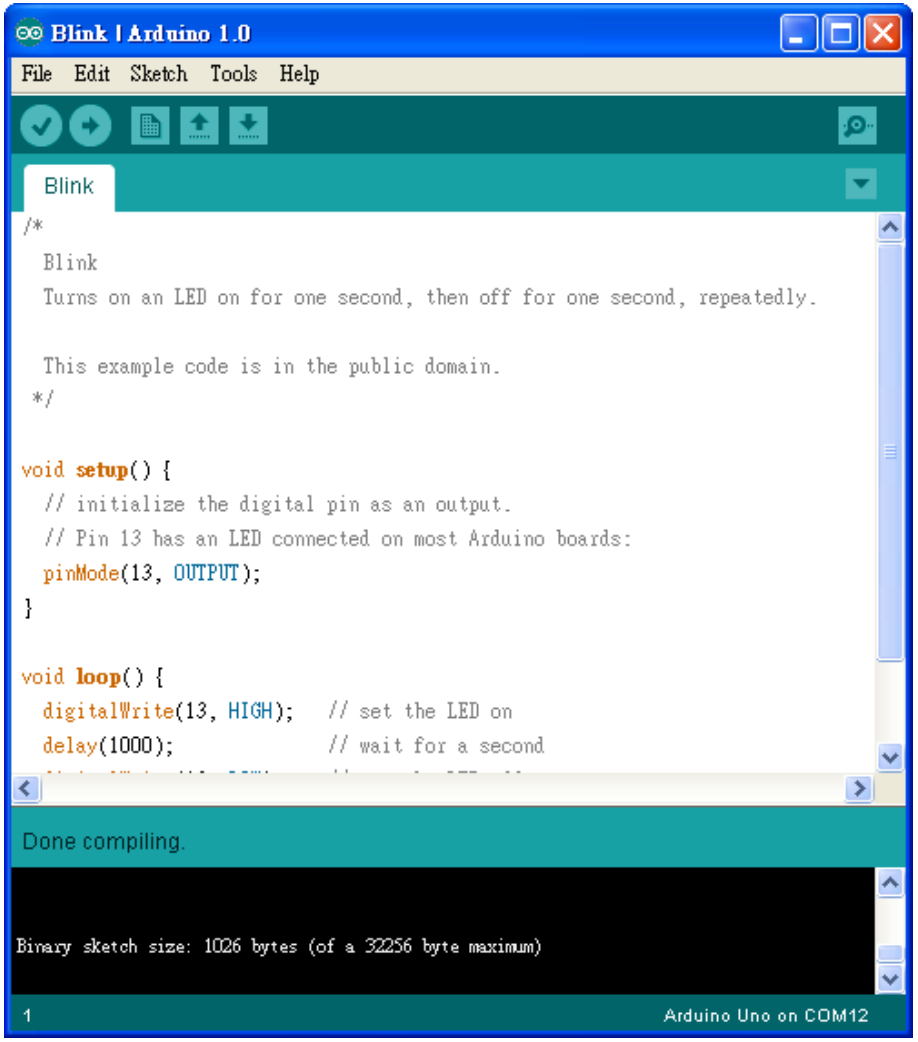
SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 10,059,224 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

$3 $5 $10 $25 $50 OTHER

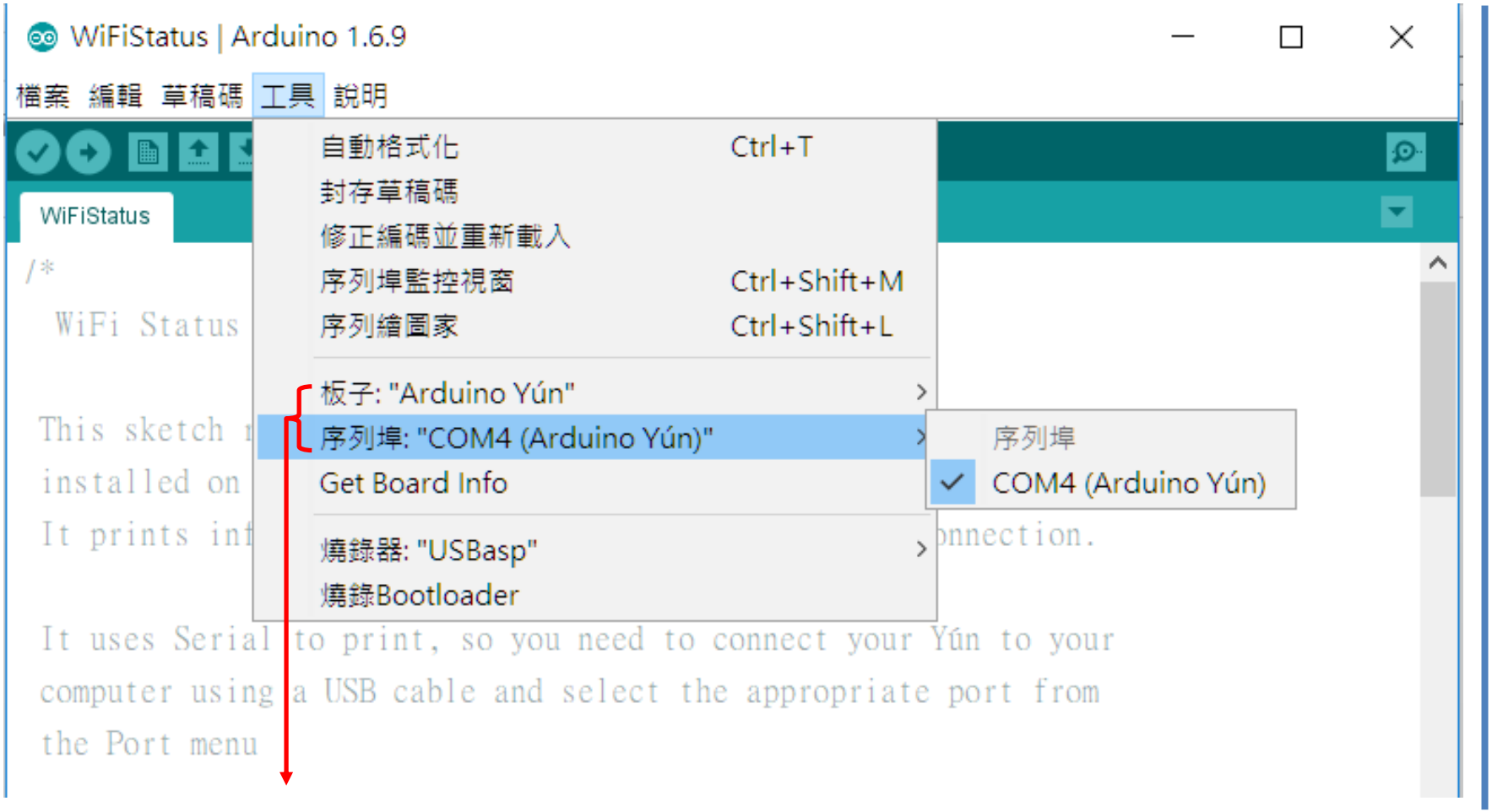
JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

# Arduino IDE

- Open Source
- Cross Platform
  - Windows
  - Mac OS X
  - Linux
- Simple UI
  (Easy to Use)



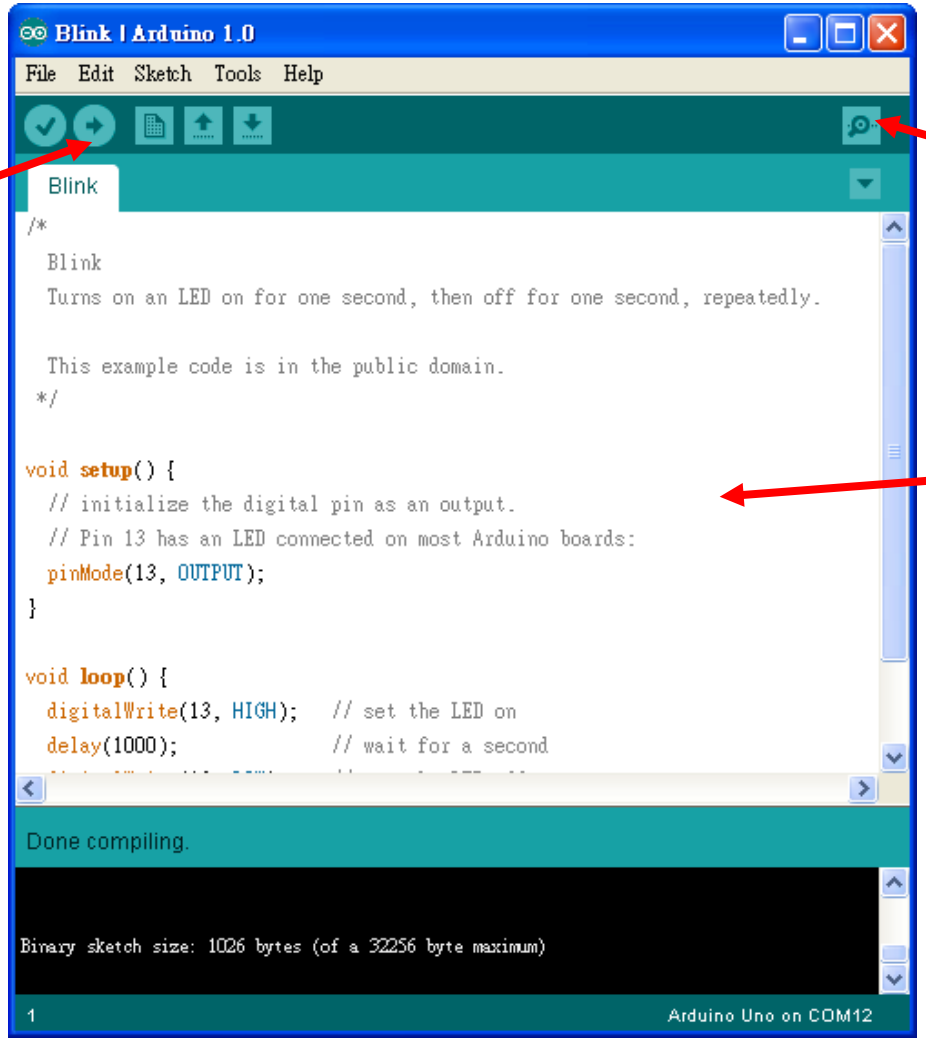
# Arduino IDE




Set “Arduino Yun”!! (It uses Yun here)

# Arduino IDE

Upload code

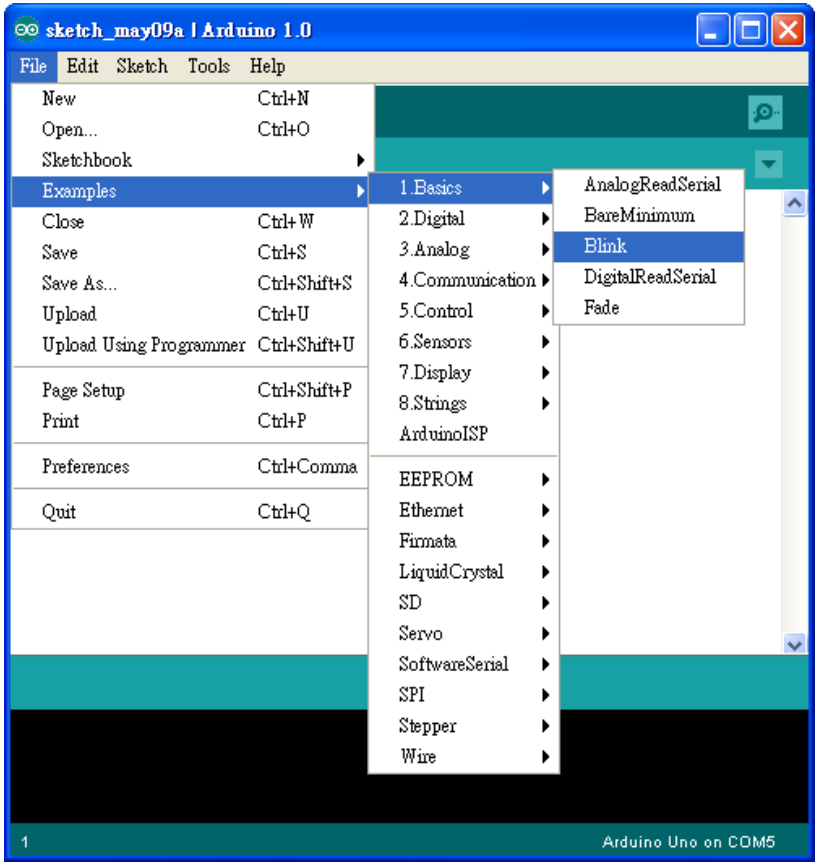



Open terminal
(Serial Monitor)

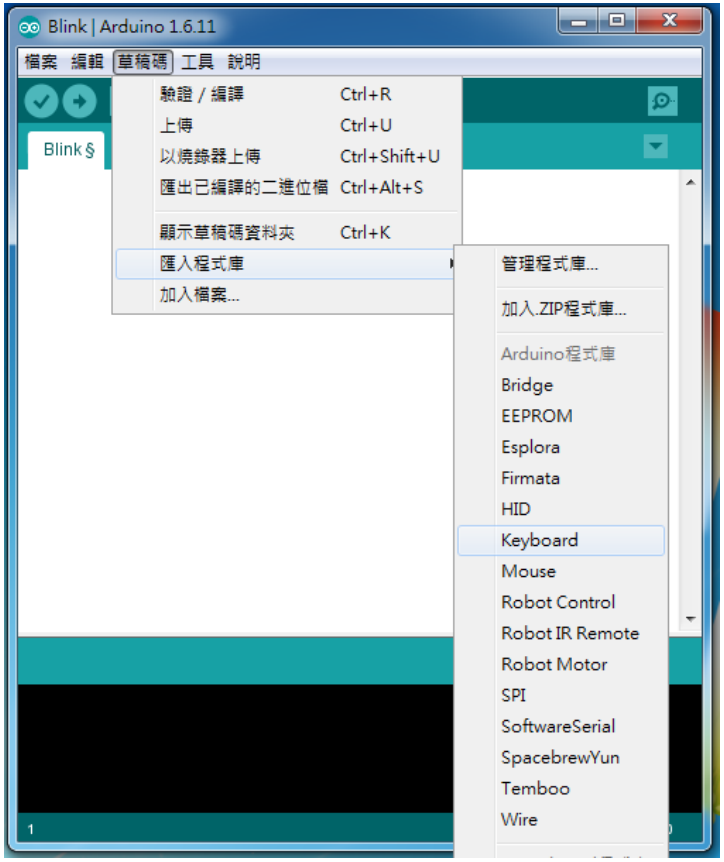
Your code

# Arduino IDE

Build-in examples:
File -> Examples -> …

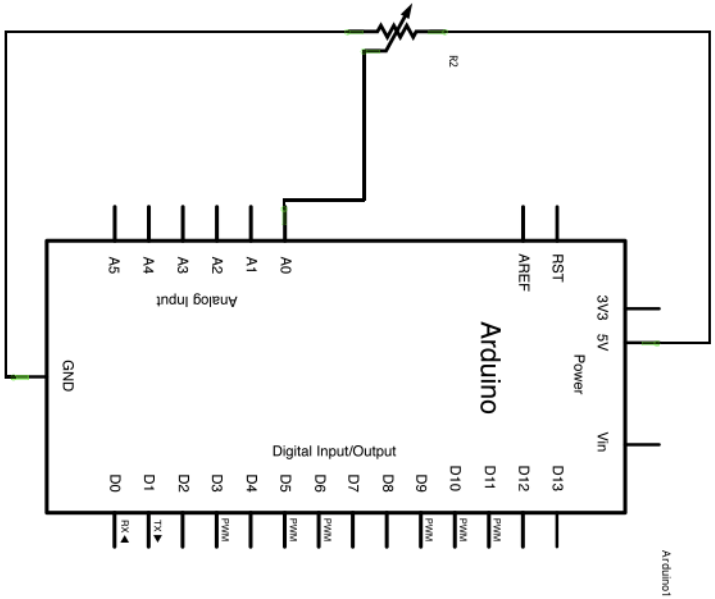


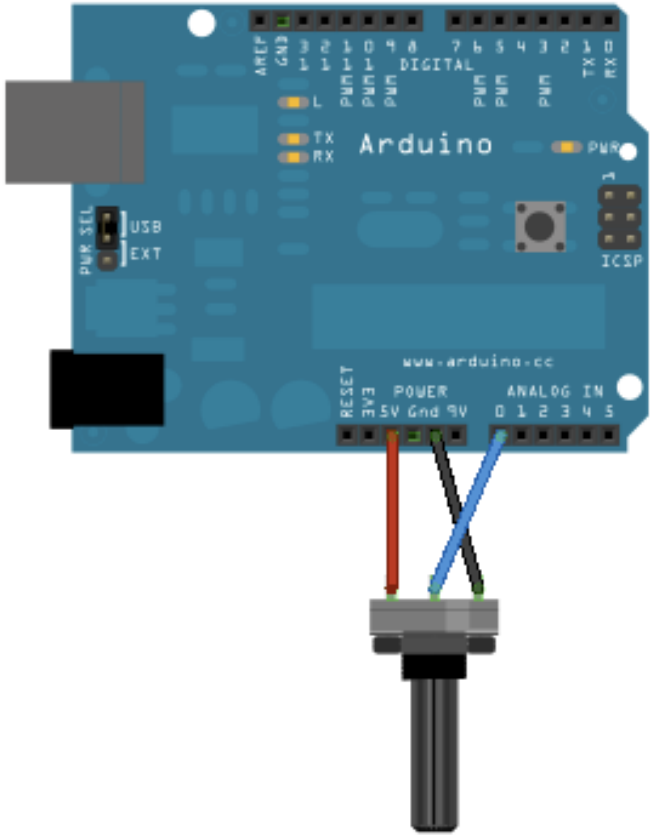

Import library



# 1. Analog Read Serial

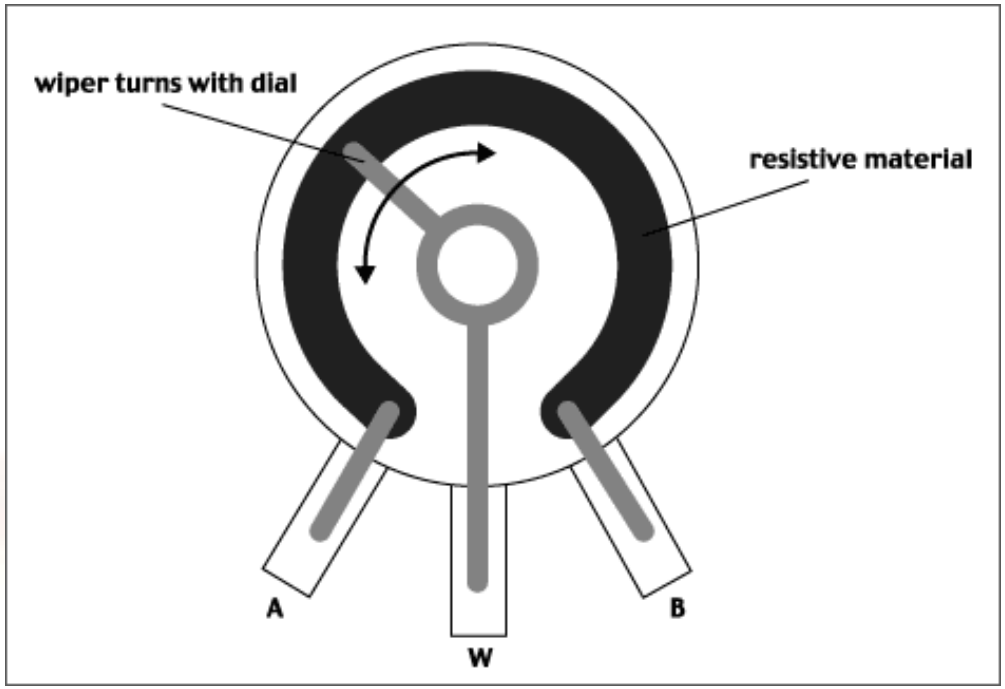
- Goal: shows you how to read analog input from the physical world using a potentiometer.
- Hardware Required
  - Arduino Board
  - 10-kilohm Potentiometer



# 1. Analog Read Serial

- Potentiometer



# 1. Analog Read Serial

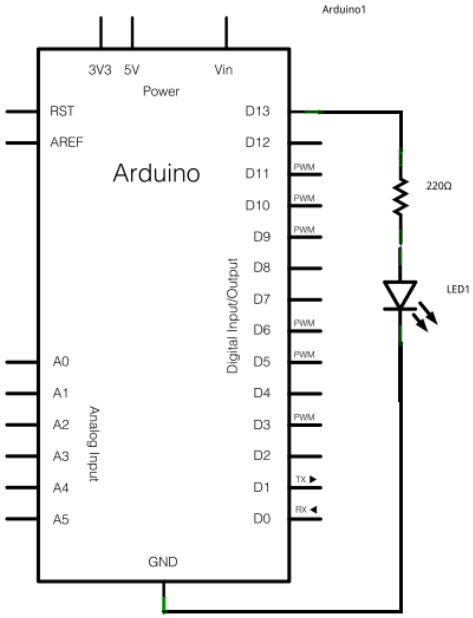
```
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}

 void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // print out the value you read:
 Serial.println(sensorValue);
 delay(1);        // delay in between reads for stability
}
```

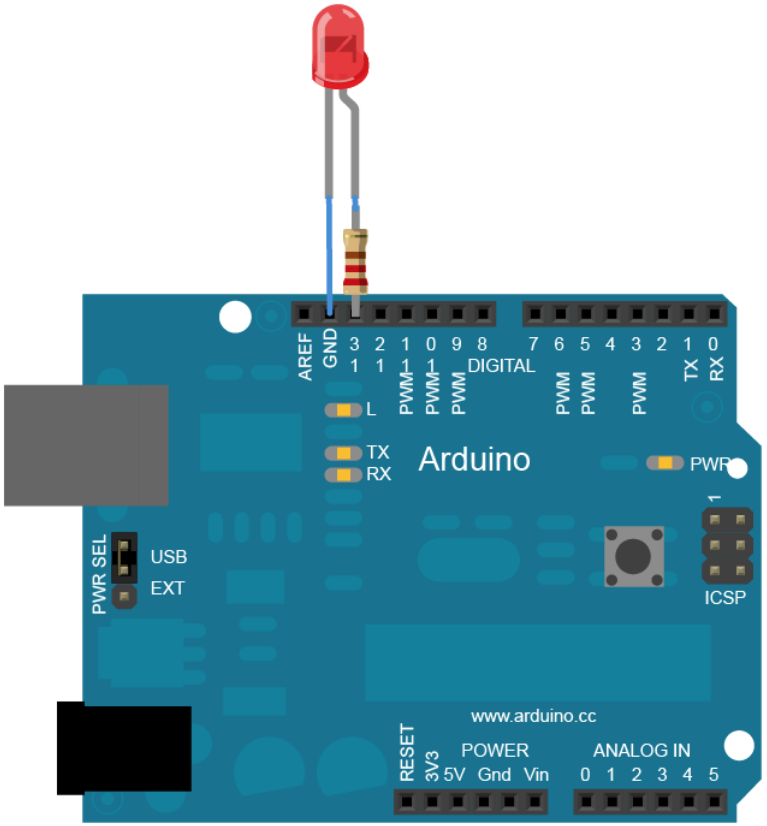
# 1. Analog Read Serial

- Syntax
  - analogRead(pin)
- Description
  - 10-bit analog to digital converter.
  - Map input voltages 0~5 volts ---> 0~1023.
  - Resolution: 5 volts / 1024 or 0.0049 volts per unit.
- Returns
  - int (0 to 1023)
- Example
  - value = analogRead(3);

# 2. Digital Write

- Goal: shows the simplest thing you can do with an Arduino to see physical output: it blinks an LED.
- Hardware Required
  - Arduino Board
  - LED



# 2. Digital Write

```
int led = 13;

// the setup routine runs once when you press reset:
  void setup() {
  pinMode(led, OUTPUT);    // initialize the digital pin as an output.
}

// the loop routine runs over and over again forever:
  void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

# 2. Digital Write

- Syntax
  - pinMode(pin, mode)
- Parameters
  - pin: the number of the pin whose mode you wish to set
  - mode: INPUT, OUTPUT, or INPUT_PULLUP
- Example
  - pinMode(13, OUTPUT);

# 2. Digital Write

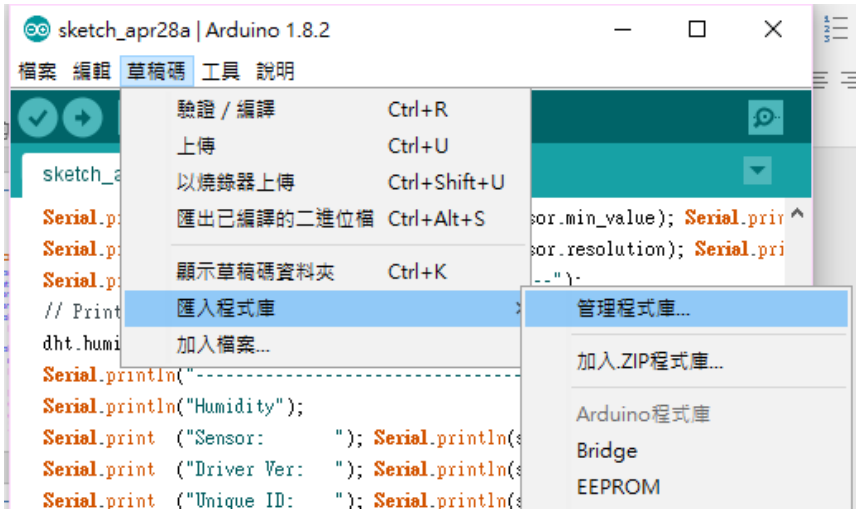
- Syntax
  - digitalWrite(pin, value)
- Parameters
  - pin: the pin number
  - value: HIGH or LOW
- Example
  - digitalWrite(13, HIGH);
  - digitalWrite(13, LOW);

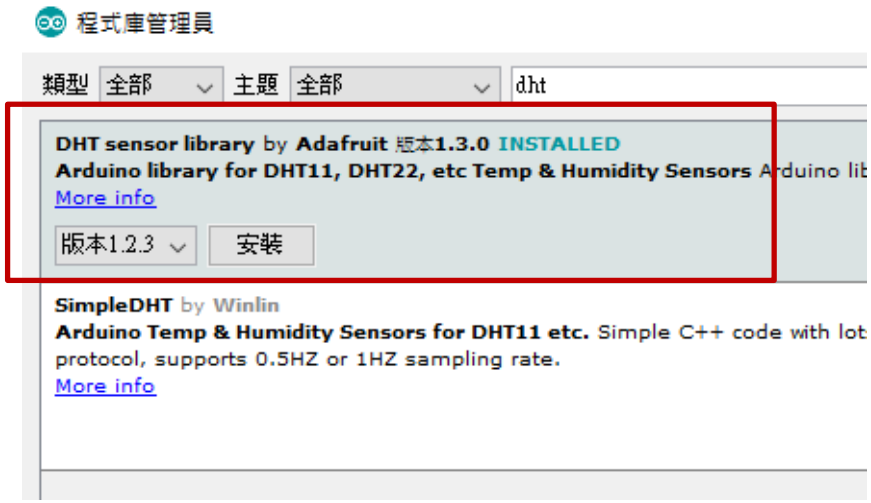
# Quiz 1

- **讓Arduino同時顯示多種感測器的數值**
- 用Arduino接收溫溼度sensor (DHT11)、以及土壤濕度sensor，將數值同時顯示在Serial Monitor上
  - DHT11的使用方法請參閱後面投影片
- 格式: "Temp: ___, Humi: ____, Mois: ____"

# (1) DHT11 - Install Library

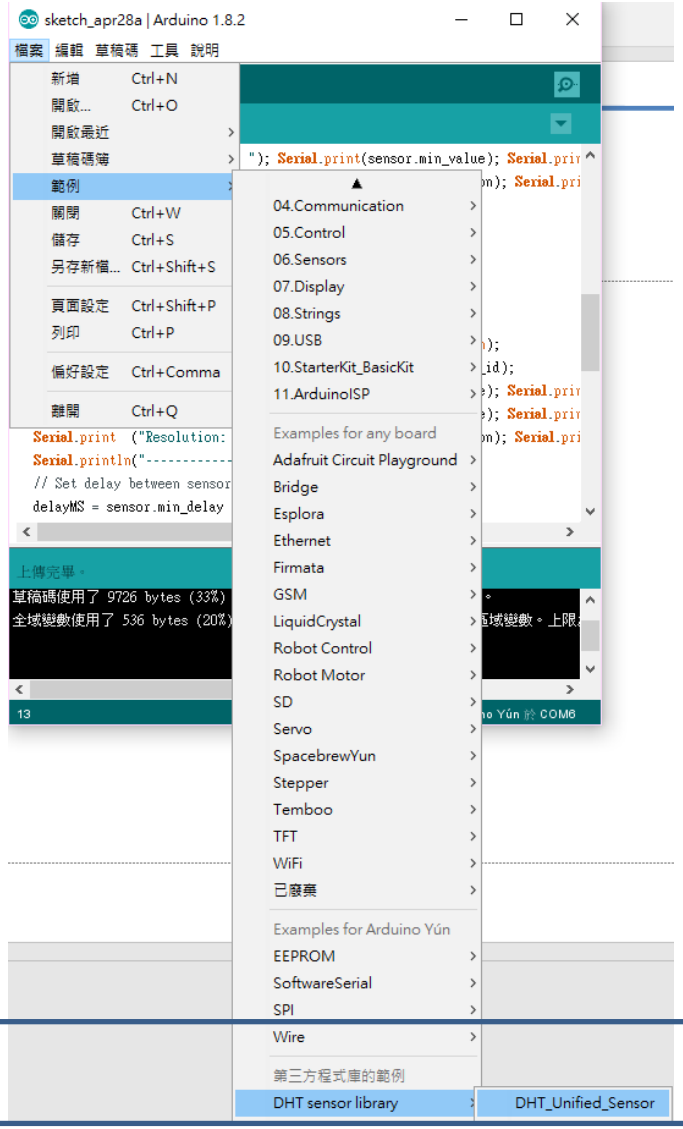
- 下載DHT11的Library:




- 或是直接從[https://github.com/adafruit/DHT-sensor-library](https://github.com/adafruit/DHT-sensor-library)下載，將資料夾重新命名成DHT後放至:
  - Windows: <user>\Documents\Arduino\libraries
  - Pi: /home/pi/sketchbook/libraries

# (1) Select Sketch



# (1) DHT11

sketch_apr28a | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明

sketch_apr28a §

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN            2         // Pin which is connected to the DHT senso

// Uncomment the type of sensor in use:
#define DHTTYPE           DHT11     // DHT 11
//#define DHTTYPE           DHT22     // DHT 22 (AM2302)
//#define DHTTYPE           DHT21     // DHT 21 (AM2301)

// See guide for details on sensor wiring and usage:
//   https://learn.adafruit.com/dht/overview

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup() {
```

上傳完畢。

草稿碼使用了 9726 bytes (33%) 的程式儲存空間。上限為 28672 bytes。

全域變數使用了 536 bytes (20%) 的動態記憶體，剩餘 2024 bytes 給區域變數。上限為 2560 bytes 。

9 Arduino Yún 於 COM6

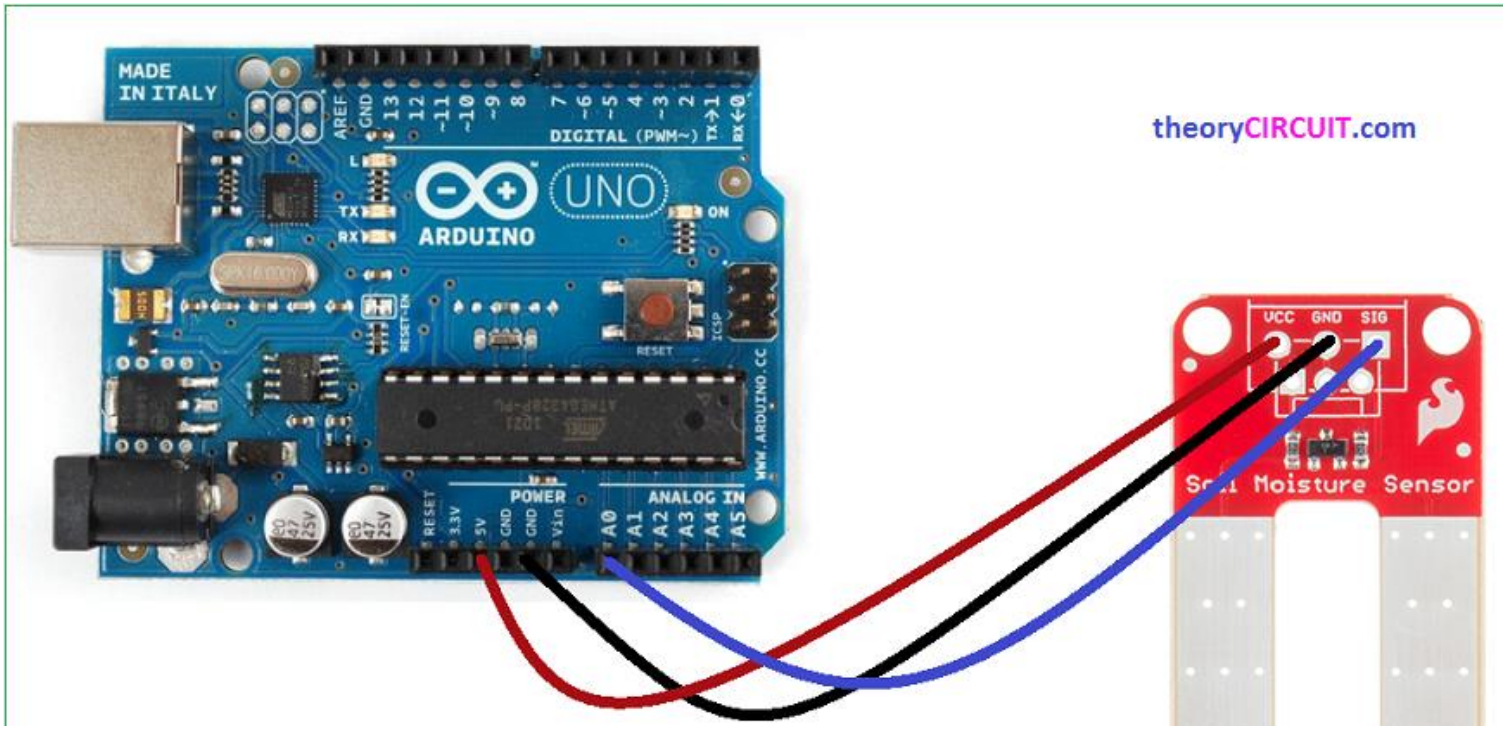
COM6 (Arduino Yún)

傳送

```
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
Temperature: 26.00 *C
Humidity: 18.00%
```

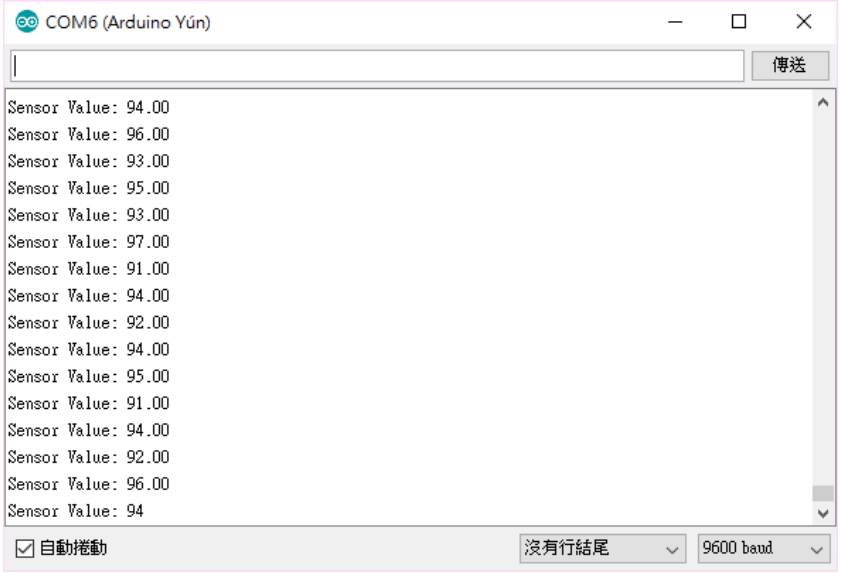
自動捲動 沒有行結尾 9600 baud

# (2) Soil Moisture Sensor



# (2) Soil Moisture Sensor

```
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}

 void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // print out the value you read:
 Serial.println(sensorValue);
 delay(1);
 // delay in between reads for stability
}
```




將上述兩個Sensor的資料一起顯示在serial monitor上

# 2. Communication Methods

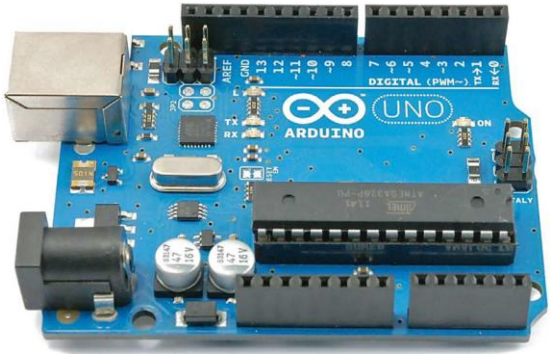
A. USB console
   - 安裝Arduino IDE在PI上面

B. Wi-Fi
   - Arduino做為client, 發送HTTP/POST資料至server端
     (搭配上周的php程式)
   - 使用Arduino YUN或ESP8266模組

C. $I^2C$

D. Software serial

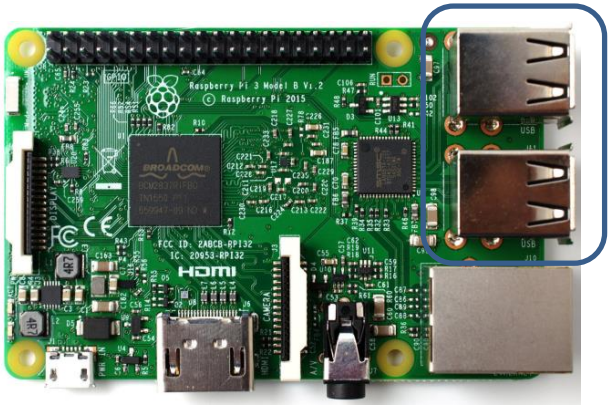
E. TTL

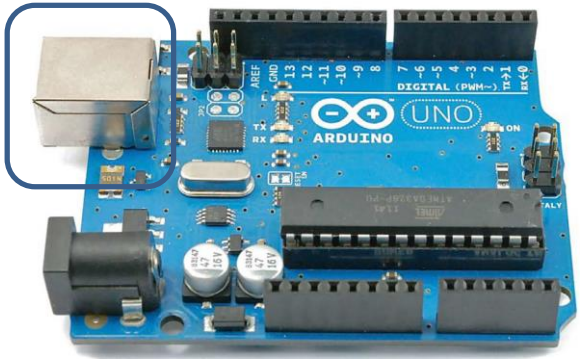


參考用

# 2(A) USB console

- sudo apt-get update
- sudo apt-get install arduino
  - 接著就能在 PI 的桌面找到 Programmings / Arduino.
    點選後就會開啟 Arduino IDE
  - 需要使用圖形化介面

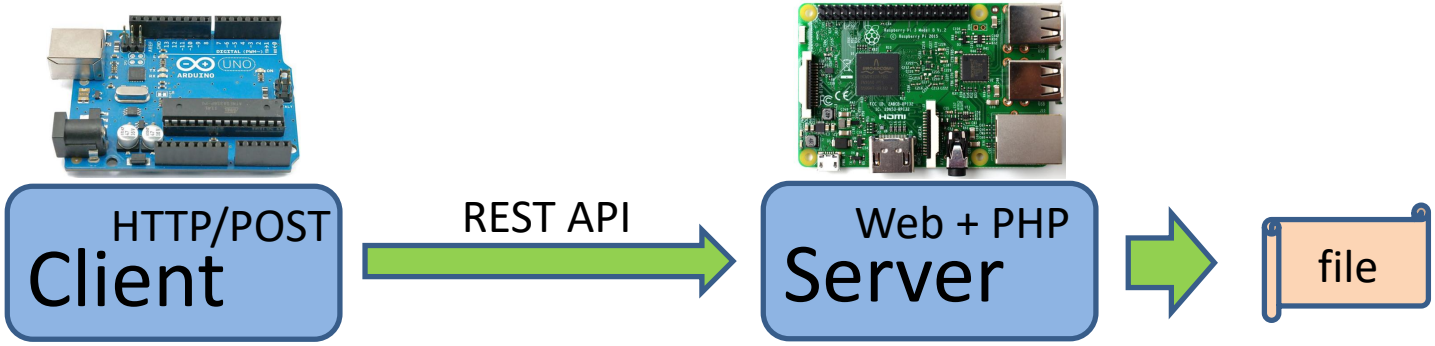




參考用

# 2(B) Wi-Fi

- 0. 確認Wi-Fi狀態
- 1. 設定Wi-Fi & 開啟REST API
- 2. 連接sensor & 送出HTTP/POST
- 3. server端(PI)接收到POST後, 存成檔案




- PS. 理論上行的通, 實際上會遇到很多奇怪的問題 -.-
  - Arduino YUN記憶體很小, 有可能會跑不太動, 連帶影響無法使用Wi-Fi傳資料
  - 大家同時啟動Wi-Fi, 會有嚴重的碰撞問題

參考用

Arduino YUN的範例

# 2(B) Wi-Fi

- Sample code: WiFi Status
- Goal: This lab runs a script called "pretty-wifi-info.lua" installed on your Yún in the folder /usr/bin. It prints out the status of your WiFi connection.
- Hardware Required
  - Arduino Yun

參考用

Arduino YUN的範例

# 2(B) Wi-Fi

```
#include <Process.h>

void setup() {
  Serial.begin(9600);  // initialize serial communication
  while (!Serial);        // do nothing until the serial monitor is opened

  Serial.println("Starting bridge...\n");
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
  Bridge.begin();                // make contact with the linux processor
  digitalWrite(13, HIGH);  // Led on pin 13 turns on when the bridge is
ready

  delay(2000);  // wait 2 seconds
}
```

參考用

Arduino YUN的範例

# 2(B) Wi-Fi

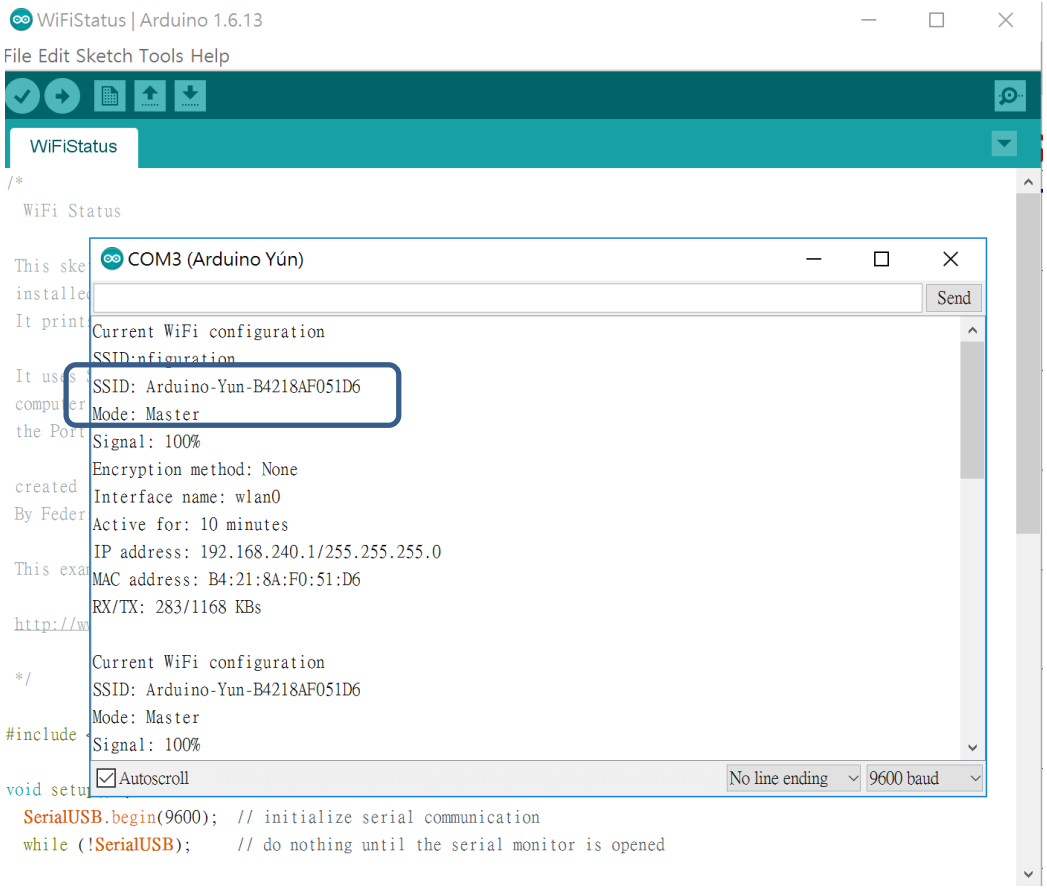
```
void loop() {
  Process wifiCheck;  // initialize a new process

  wifiCheck.runShellCommand("/usr/bin/pretty-wifi-info.lua");
                                        // command you want to run

  // while there's any characters coming back from the
  // process, print them to the serial monitor:
  while (wifiCheck.available() > 0) {
    char c = wifiCheck.read();
    Serial.print(c);
  }
  Serial.println();
  delay(5000);
}
```

參考用

Arduino YUN的範例

# Connect and Configure Yun's WiFi

1. Connect to Arduino Yun (identify by MAC)



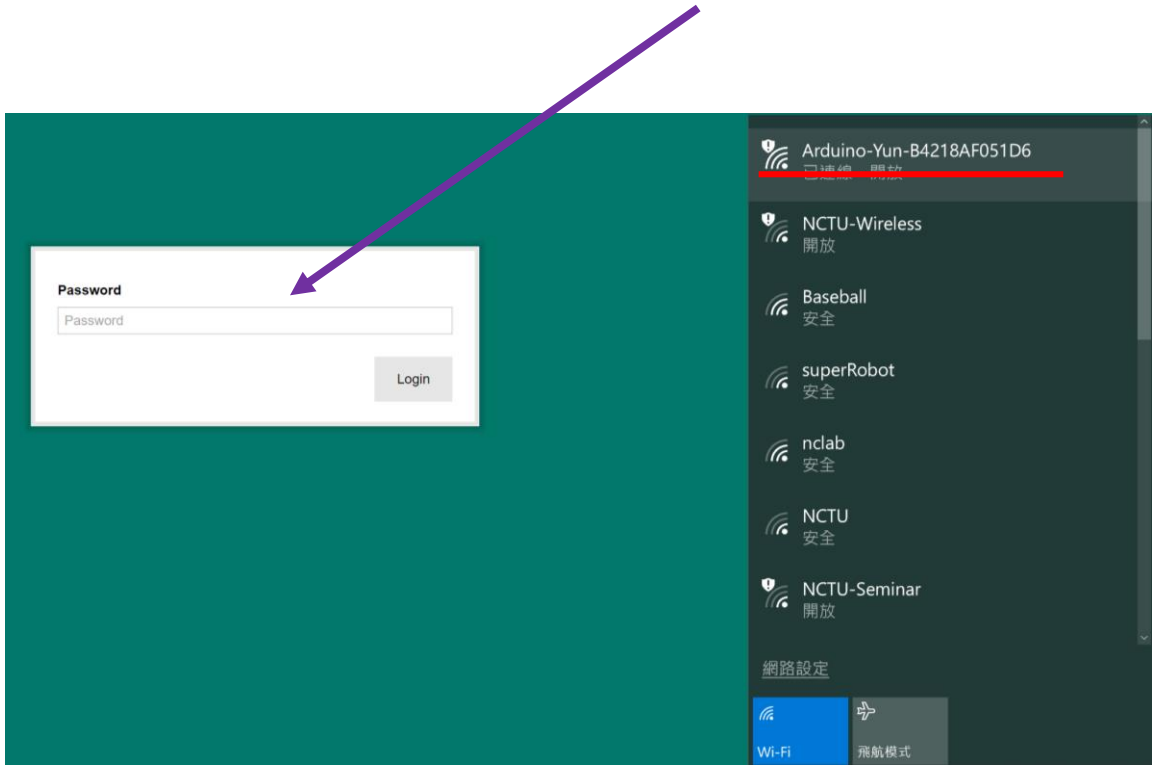

Mode: Master -> 尚未設定Wi-Fi
Mode: Slave -> 已設定過Wi-Fi

參考用

Arduino YUN的範例

# Connect and Configure Yun's WiFi

2. Enter http://arduino.local or http://192.168.240.1
3. Enter password (default: arduino)

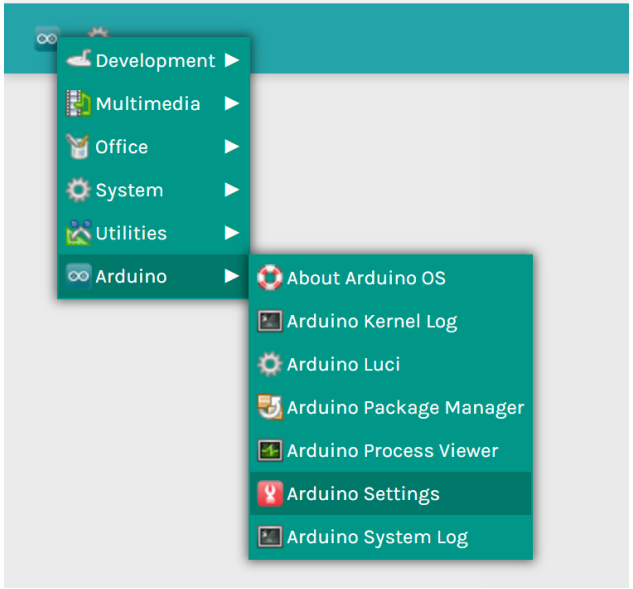


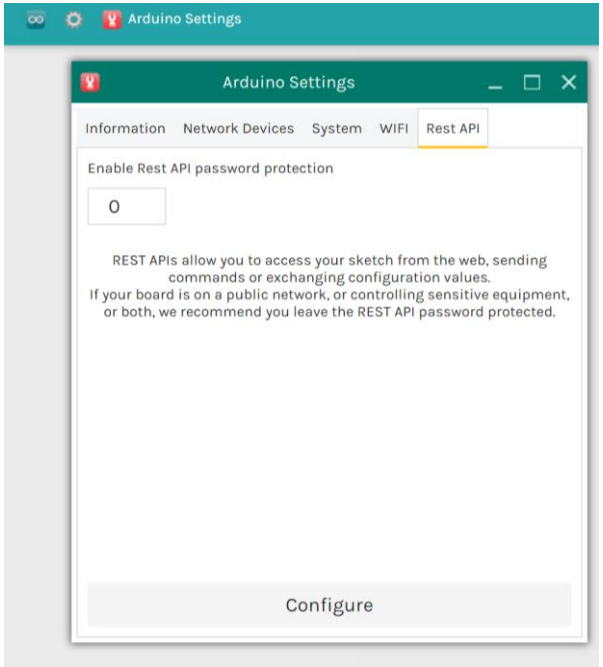
參考用

Arduino YUN的範例

# Connect and Configure Yun's WiFi

- In configuration
  - **Enable** REST API ACCESS
- Press "Configure"



參考用

Arduino YUN的範例

# HTTP/Post – part 1

Arduino sample code

```
#include <Process.h>

int state = 0;
int lastdata = 0;
int digital_sensor = 13; //ex: button

void setup() {
  // put your setup code here, to run once:
  Bridge.begin();
  Serial.begin(9600);
  pinMode(digital_sensor, INPUT);
}
```

參考用

Arduino YUN的範例

# HTTP/Post – part 2

Arduino sample code

```
void loop() {
  // put your main code here, to run repeatedly:
  int data= digitalRead(MotionButton);
  Serial.println(data);

  if(data==1||lastdata==data){
     if(data!=state){
       sendData(data);
       state = data;
     }
  }
  lastdata = data;
  if(data)
    delay(10000);
  else
    delay(500);
}
```

參考用

Arduino YUN的範例

# HTTP/Post – part 3

Arduino sample code

```
void sendData(int data) {
  // curl -d "Motion=3&Name=5566" http://server_ip_address
  Process p;
  p.runShellCommand("curl -d \"Motion="+String(data)+"&Name=5566\"
http:// server_ip_address ");
  Serial.print("send:");
  Serial.println(data);
}
```

利用YUN裡面的Linux系統, 執行curl送出HTTP POST
等同於執行 curl -d "Motion=3&Name=5566" http://server_ip_address

# Custom Communication with an Arduino

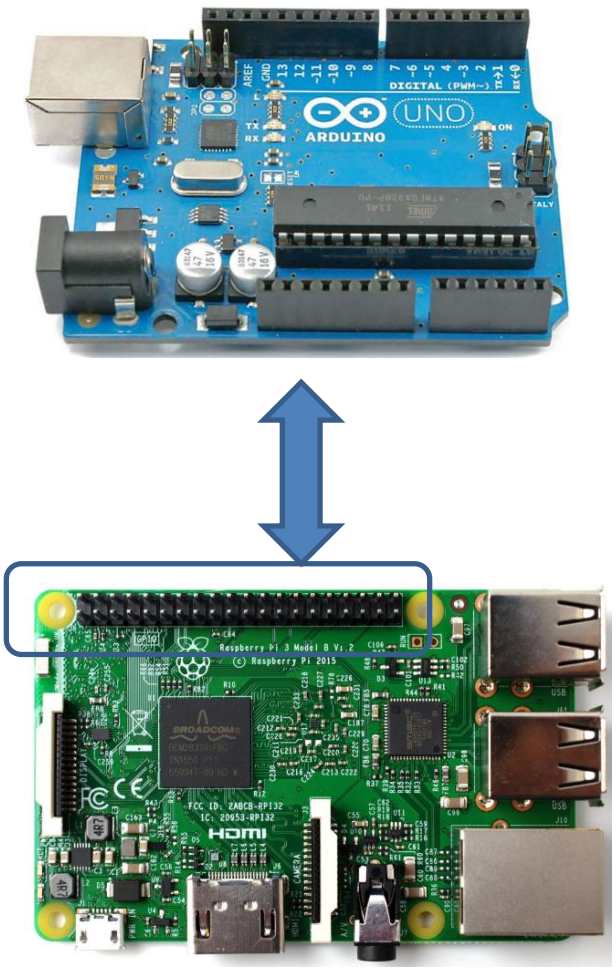
- 試試看用GPIO的線路來通訊!?

C. **over I2C**

D. over software serial

E. over TTL Serial

  - 需要將PI的TTL腳位連到Arduino
    無法連接電腦, 需外接螢幕



# 2(c) Over I2C

- Custom Communication with an Arduino over I2C

SDA
SCL
GND

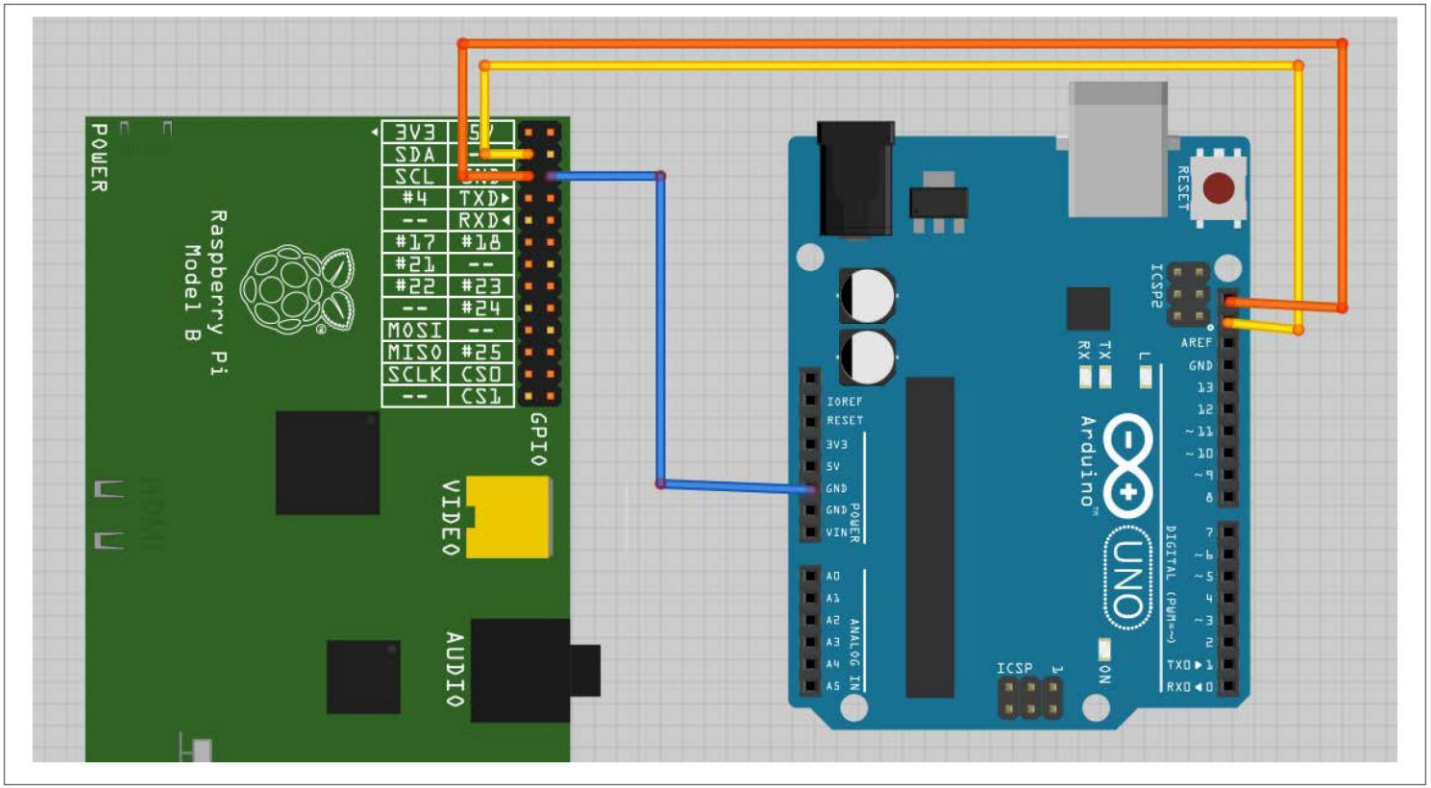



*Figure 14-16. Wiring diagram for I2C communication with an Arduino*

# 2(c) Over I2C

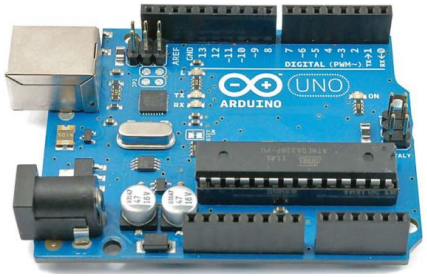
- exchange data by I2C interfaces
  - The I2C bus has the concept of master and slave devices.
  - The master device controls the bus and a number of slave devices can be connected to one master, with each slave having its own address.
- Arduino: I2C slave device
- Raspberry PI: I2C master
  - use the SMBus Python library to write an I2C program

```
sudo apt-get install python-smbus
```

# 2(c) Over I2C

Source code for Arduino: (part 1/3)
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoI2C/ArduinoI2C.ino

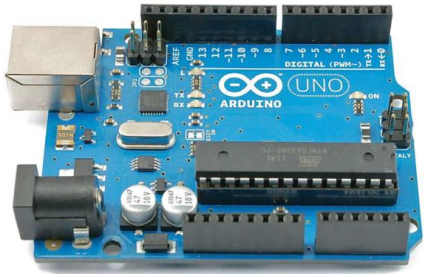
```
#include <Wire.h>
int SLAVE_ADDRESS = 0x04;
int ledPin = 13;
int analogPin = A0;
boolean ledOn = false;

void setup()
{
   pinMode(ledPin, OUTPUT);
   Wire.begin(SLAVE_ADDRESS);
   Wire.onReceive(processMessage);
   Wire.onRequest(sendAnalogReading);
   Serial.begin(9600);
}
```



# 2(c) Over I2C

Source code for Arduino:  (part 2/3)
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoI2C/ArduinoI2C.ino

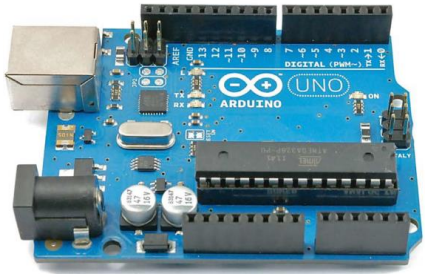
```
void loop()
{
}

void processMessage(int n)
{
  Serial.println("In processMessage");
    char ch = Wire.read();
    if (ch == 'l')
    {
      toggleLED();
    }
}
```



# 2(c) Over I2C

Source code for Arduino: (part 3/3)
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoI2C/ArduinoI2C.ino

```
void toggleLED()
{
  ledOn = ! ledOn;
  digitalWrite(ledPin, ledOn);
}

void sendAnalogReading()
{
  Serial.println("In sendAnalogReading");
  int reading = analogRead(analogPin);
  Wire.write(reading >> 2);
}
```



# 2(c) Over I2C

Source code for Raspberry PI:
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ardu_pi_i2c.py

```
import smbus
import time
bus = smbus.SMBus(1)

# This must match in the Arduino Sketch
SLAVE_ADDRESS = 0x04

def request_reading():
    reading = int(bus.read_byte(SLAVE_ADDRESS))
    print(reading)

while True:
    command = raw_input("Enter command: l - toggle LED, r - read A0 ")
    if command == 'l' :
        bus.write_byte(SLAVE_ADDRESS, ord('l'))
    elif command == 'r' :
        request_reading()
```

# 2(c) Over I2C

- It's designed to show communication in both directions.

- Entering the **l** command
  - toggles the Arduino's built-in LED on and off
- Entering the **r** command
  - reports a reading from the Arduino's A0 analog input

```
if command == 'l' :
    bus.write_byte(SLAVE_ADDRESS, ord('l'))
  elif command == 'r' :
    request_reading()
```

參考用

# Custom Communication with an Arduino

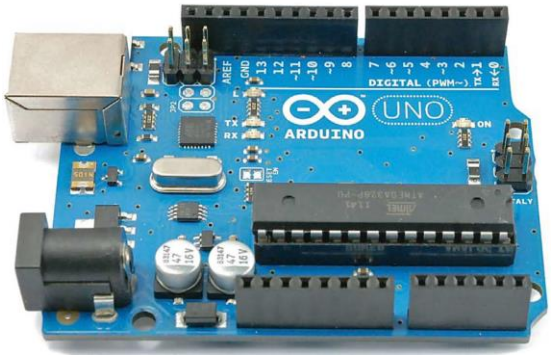
- 試試看用GPIO的線路來通訊!?

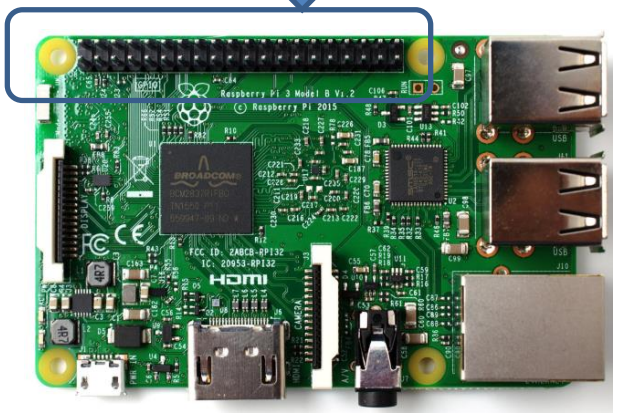
C. over I2C

D. **over software serial**

E. over TTL Serial

  - PI的TTL腳位連到Arduino
    無法連接電腦, 需外接螢幕



參考用

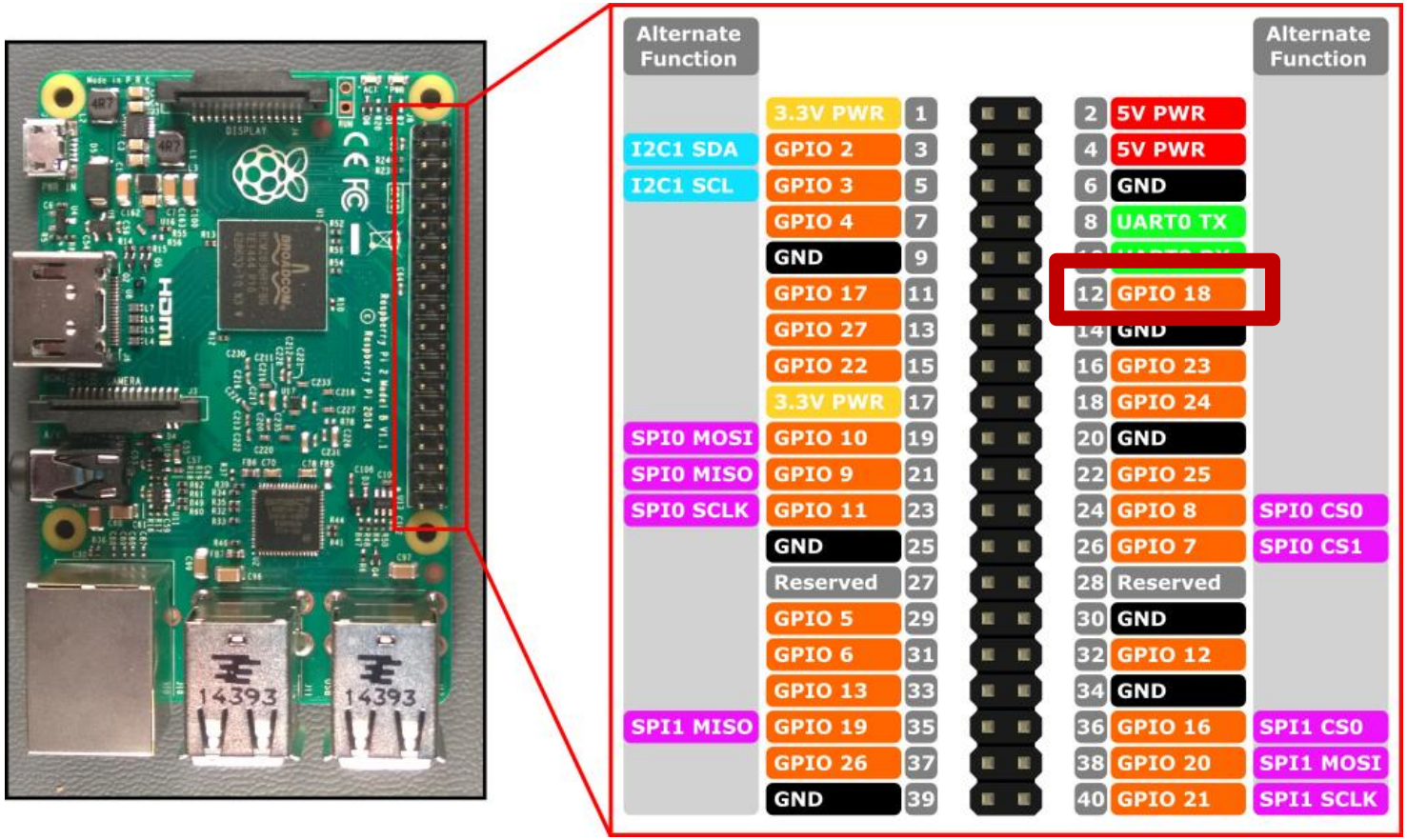
# 2(D) over software serial

- 將Arduino的PIN 11接到Pi的GPIO 18上
- 先在Raspberry Pi 上面下載library: pigpio
  http://abyz.co.uk/rpi/pigpio/download.html
  - 在PI的terminal輸入以下指令

```
wget http://abyz.co.uk/rpi/pigpio/pigpio.zip
unzip pigpio.zip
cd PIGPIO
make -j4
sudo make install
sudo ./pigpiod
```

参考用

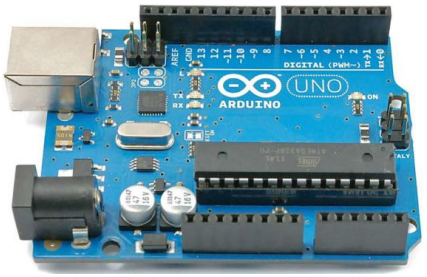
# 2(D) over software serial



参考用

# 2(D) over software serial

Code for Arduino:




```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

void setup() {
  Serial.begin(57600);
  while (!Serial);

  mySerial.begin(9600);
  mySerial.println("Hello, world?");
}

void loop() {
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}
```

參考用

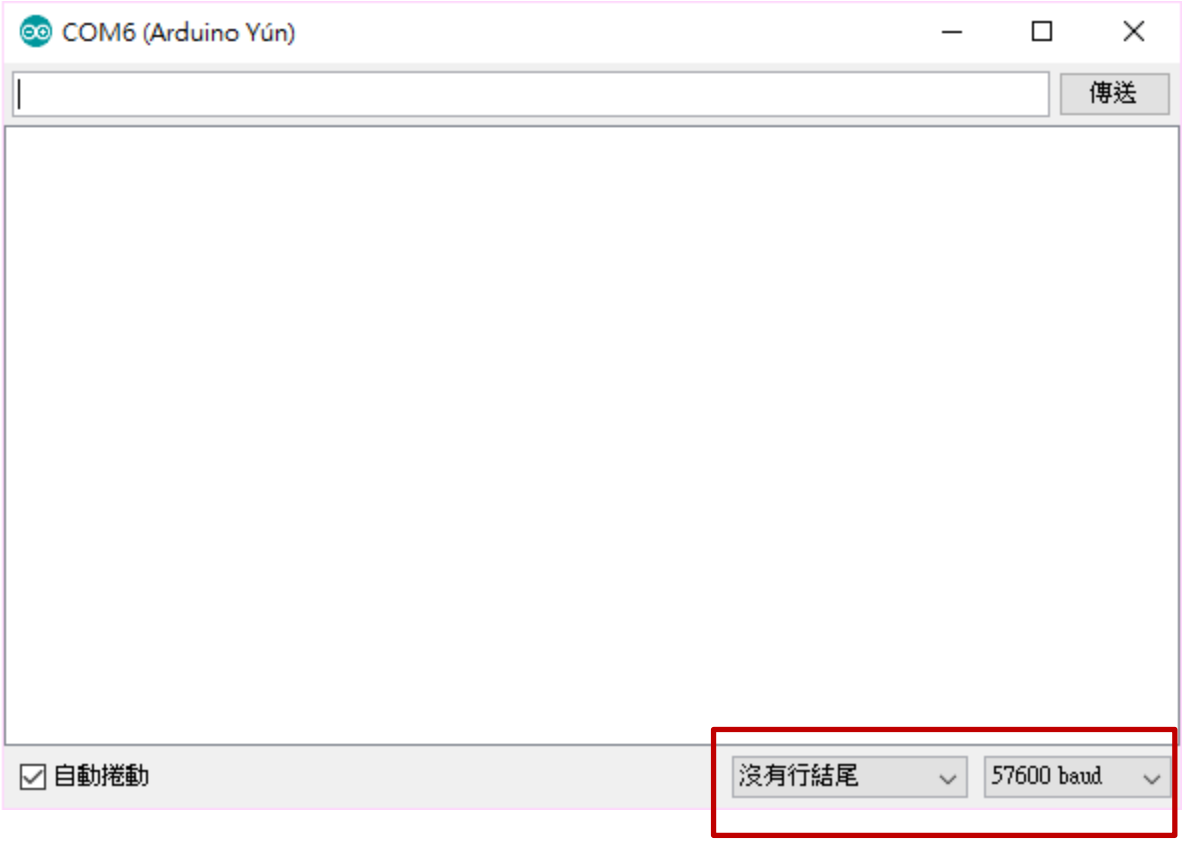
# 2(D) over software serial

Code for Raspberry Pi:

```
#!/usr/bin/env python
import sys
import time
import difflib
import pigpio
RX = 18

try:
        pi = pigpio.pi()
        pi.set_mode(RX, pigpio.INPUT)
        pi.bb_serial_read_open(RX, 9600, 8)

        print "Data from software serial: "
        while 1:
                (count, data) = pi.bb_serial_read(RX)
                if count
                        print data
                time.sleep(1)
except:
        print "Error: ", sys.exc_info()[0]
        pi.bb_serial_read_close(RX)
        pi.stop()
```

參考用

# 2(D) over software serial




在Arduino終端機輸入訊息 -> 傳送到PI, 並顯示在PI的終端機

參考用

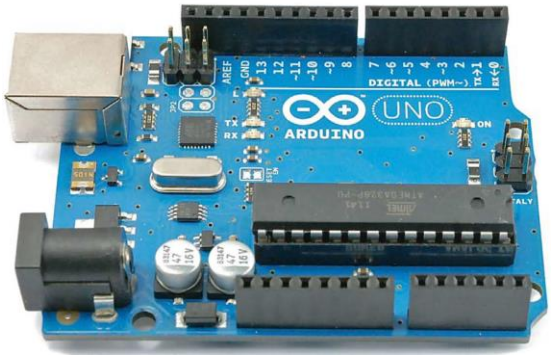
# Custom Communication with an Arduino

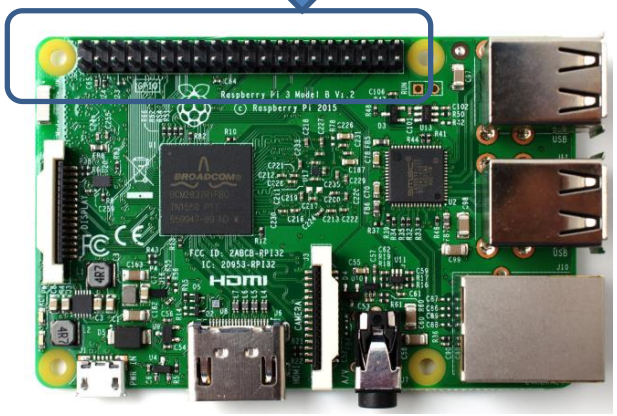
- 試試看用GPIO的線路來通訊!?

C. over I2C

D. over software serial

E. **over TTL Serial (參考用)**
   - **PI的TTL腳位連到Arduino 無法連接電腦, 需外接螢幕**



参考用

# 2(E) over TTL Serial

- Custom Communication with an Arduino over TTL Serial

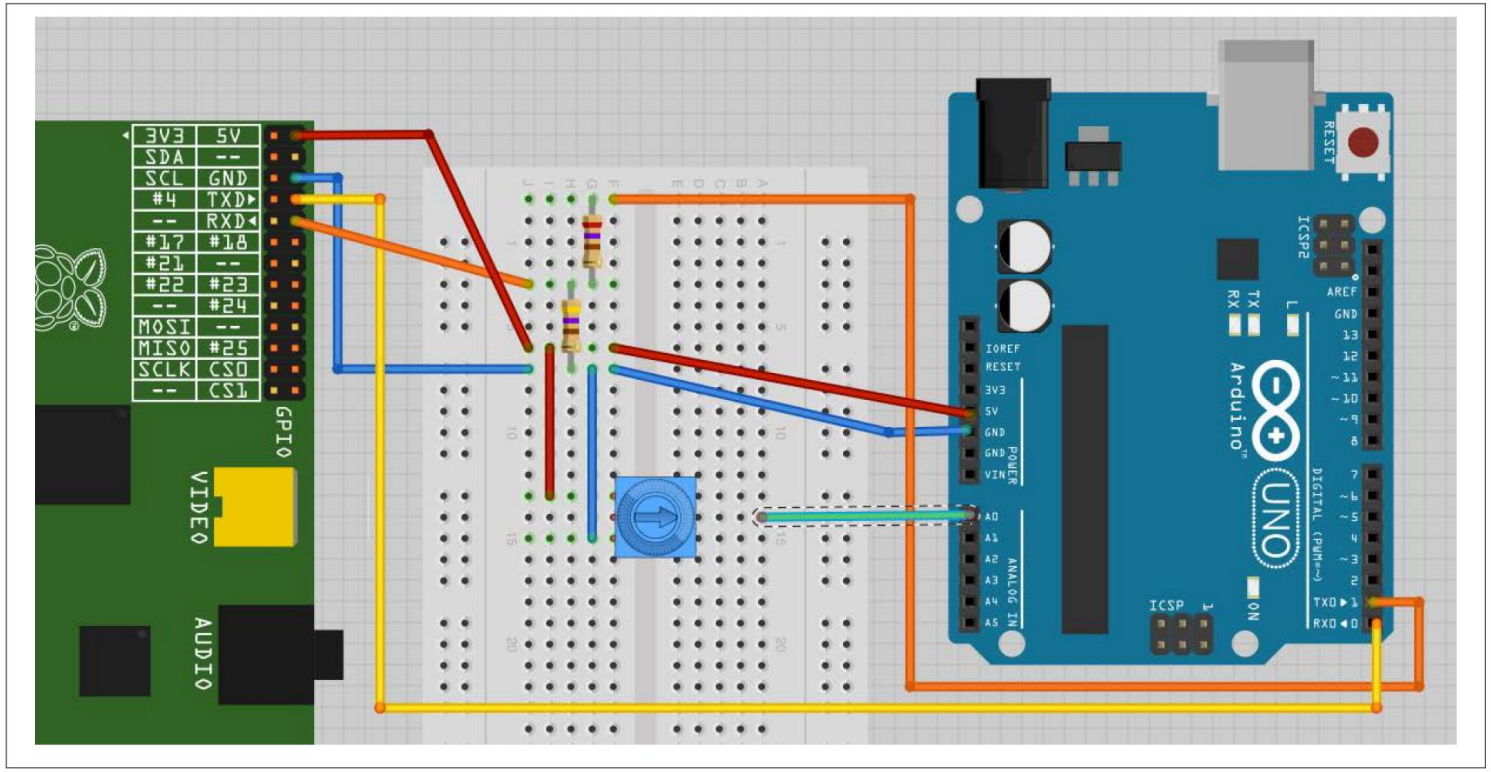



*Figure 14-15. Wiring diagram for serial communication with an Arduino (resistors)*

參考用

# 2(E) over TTL Serial

- Communication with an Arduino over TTL Serial
  - exchange data with an Arduino by using the serial interfaces of both devices
- Equip
  - 270Ω and 470Ω resistors
    - The Arduino Rx input is fine with just 3.3V from the Raspberry Pi TXD pin
    - the 5V coming from the Arduino Tx pin must be dropped to the 3V expected by the Raspberry Pi.
  - 10kΩ trimpot (可變電阻)

参考用

# 2(E) over TTL Serial

- Python script (ardu_pi_serial.py)
  - enter a command of either l or r
  - If you enter l,
    - then the built-in Arduino LED will toggle on and off.
  - If you enter r
    - the Arduino will read the analog value from its analog input A0
    - Then send back a number between 0 and 1023 as the reading.

参考用

# 2(E) over TTL Serial

Source code for Raspberry PI:
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ardu_pi_serial.py

```
import serial

ser = serial.Serial('/dev/ttyAMA0', 9600)

while True:
    command = raw_input("Enter command: l - toggle LED, r - read A0 ")
    if command == 'l' :
        ser.write('l')
    elif command == 'r' :
        ser.write('r')
        print(ser.readline())
```

ardu_pi_serial.py

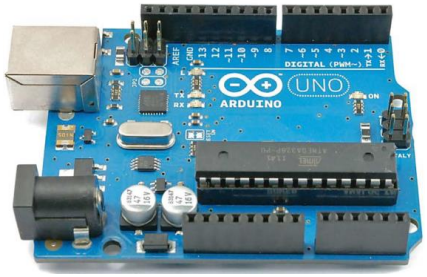
参考用

# 2(E) over TTL Serial

Source code for Arduino:  (part 1/3)

https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoSerial/ArduinoSerial.ino

```
#include "SoftwareSerial.h"

int ledPin = 13;
int analogPin = A0;

SoftwareSerial ser(8, 9); // RX, TX

boolean ledOn = false;

void setup()
{
  ser.begin(9600);
  pinMode(ledPin, OUTPUT);
}
```



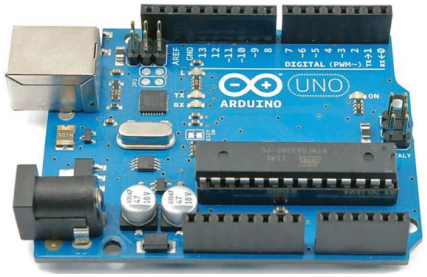
参考用

# 2(E) over TTL Serial

Source code for Arduino: (part 2/3)

https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoSerial/ArduinoSerial.ino

```
# cont.
void loop()
{
  if (ser.available())
  {
    char ch = ser.read();
    if (ch == 'l')
    {
      toggleLED();
    }
    if (ch == 'r')
    {
      sendAnalogReading();
    }
  }
}
```

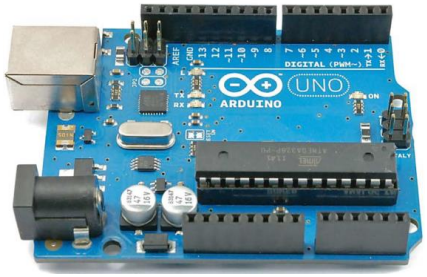


参考用

# 2(E) over TTL Serial

Source code for Arduino: (part 3/3)
https://github.com/simonmonk/raspberrypi_cookbook/blob/master/code/ArduinoSerial/ArduinoSerial.ino

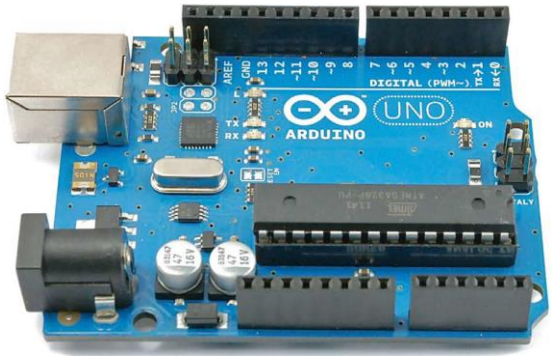
```
$ cont.
void toggleLED()
{
  ledOn = ! ledOn;
  digitalWrite(ledPin, ledOn);
}

void sendAnalogReading()
{
  int reading = analogRead(analogPin);
  ser.println(reading);
}
```



# Quiz 2

- 讓PI接收到Arduino的訊息, 並顯示出來
  - 基於Quiz1, 在Arduino同時顯示多感測器的資訊
  - 格式: "Temp: ___, Humi: ____, Mois: ____"

"Temp: ___, Humi: ____, Mois: ____"



# Bonus

- 當Sensor數值大於臨界值時, 回傳訊息給Arduino
  - Ex: Moisture > 50%的時候，Raspberry Pi回傳訊息給Arduino讓上面的LED(PIN 13)亮，反之則讓他暗

# Summary

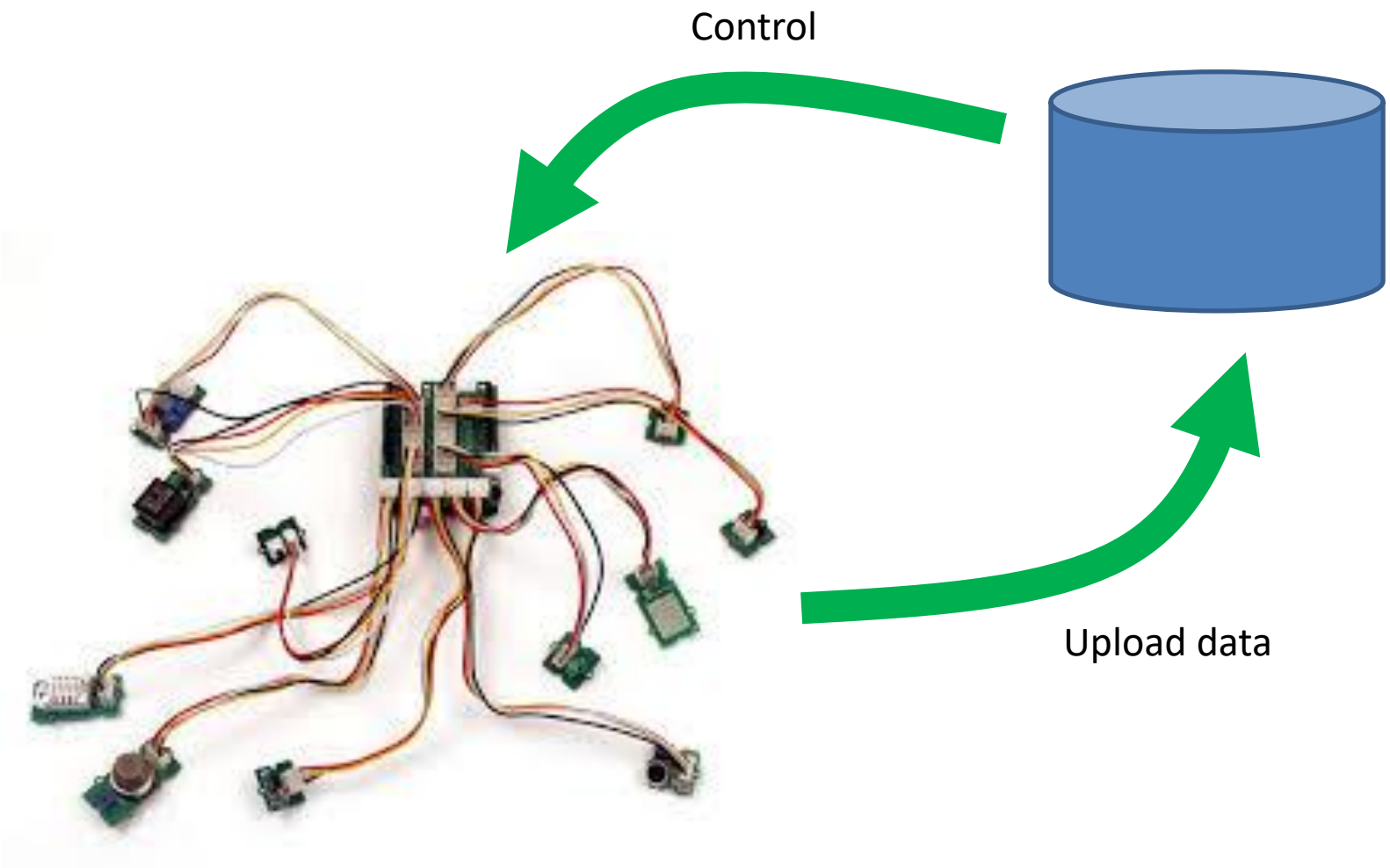
- Practice Lab (Arduino基本操作, PI與Arduino之間的通訊)

- Write code for Quiz 1 - 2, then demonstrate it to TAs
  - Quiz1: 讓Arduino同時顯示多種感測器的數值
  - Quiz2: 讓PI接收到Arduino的訊息, 並顯示出來
  - Bonus: 當Sensor數值大於臨界值時, 回傳訊息給Arduino

# 6. Cloud platform

National Chiao Tung University

# IoT Platform

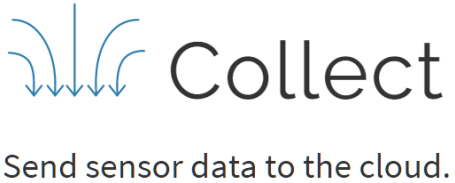


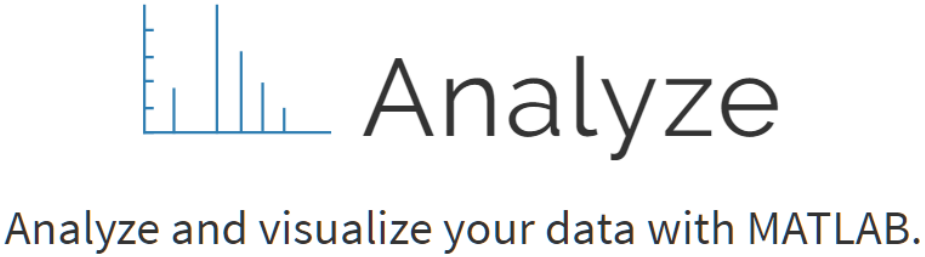
# Content

- Thingspeak
  - 1. ThingSpeak -> Device (talkback)
  - 2. Device -> ThingSpeak (upload data)

# ThingSpeak Introduction

- ThingSpeak Features
  - Data collection
  - Open API
  - Alerts
  - Event scheduling
  - MATLAB® analytics and visualizations
  - MQTT publish support
  - App integra
  - Worldwide



# ThingSpeak – public channels

## Public Channels

### ENS210 @ HTC

Channel ID: **249563**
Author: maartenpe...
ENS210 relative humidity and temperature sensor by ams

ENS210, relative humidity, temperature, AMS

### my_house

Channel ID: **9**
Author: iothans
Netduino Plus connected to sensors around the house

temp, light, sensor

### Neo Home sensor

Channel ID: **243783**
Author: meaverick
Address:ZHENLI AVE HEDONG TIANJIN CHINA

AQI, PMS5003, PM1.0, PM10, BMP180, PM2.5, DHT11

### PFC Temperatura

Channel ID: **104729**
Author: dez131266
temperatura do quarto

### Kamery Rzeszów - Pom...

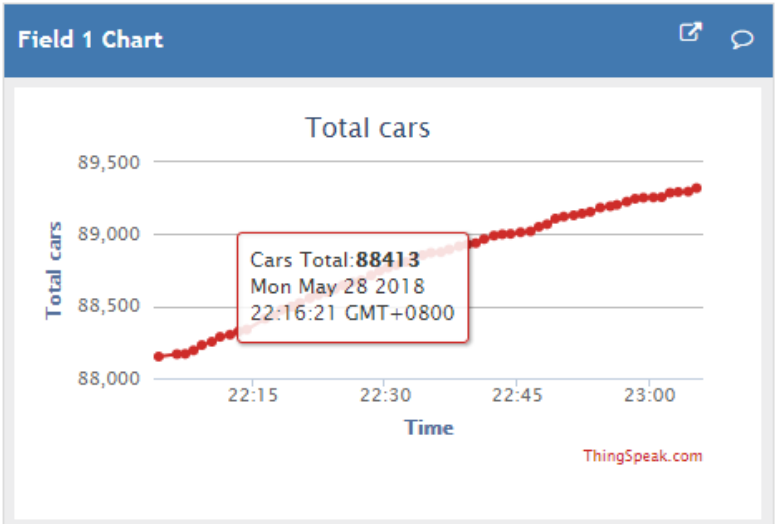
Channel ID: **418835**
Author: tvrzeszow
Pomiar czystości powietrza, temperatury, wilgotności oraz ciśnienia w Rzeszowie os Kielanówka.

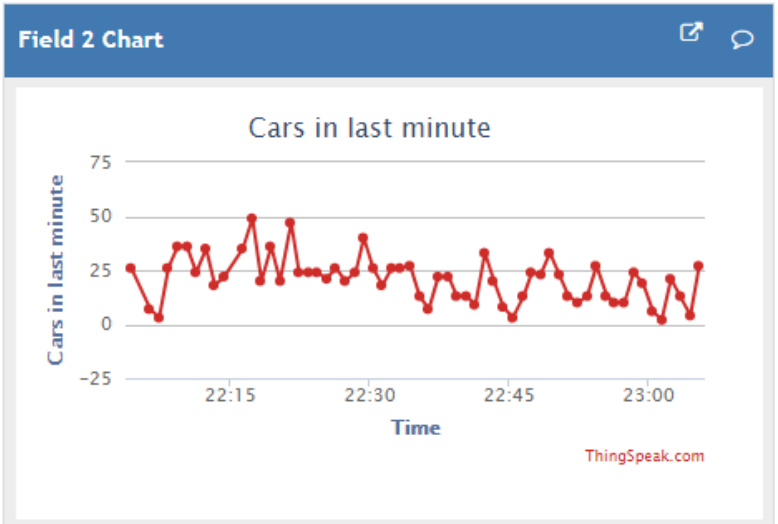
### Kamery Rzeszów - Pom...

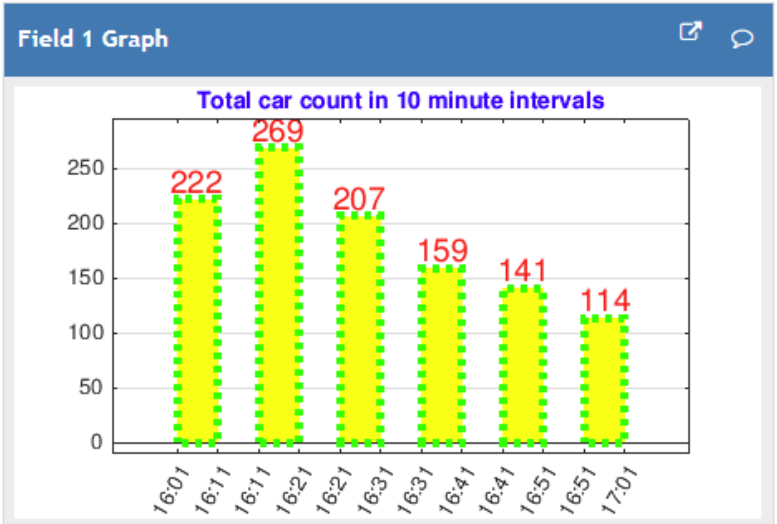
Channel ID: **377943**
Author: itprojekt
Pomiar czystości powietrza, temperatury, wilgotności oraz ciśnienia w Rzeszowie os Drabiniaka.

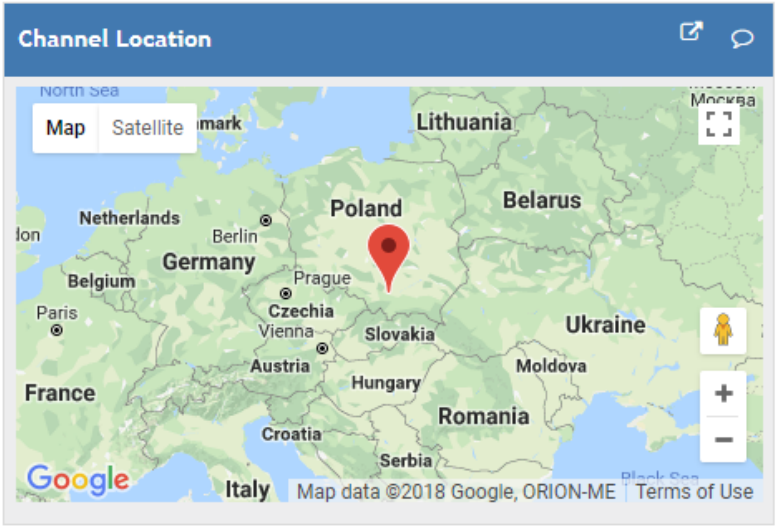
https://thingspeak.com/channels/438049

# Example - Traffic Intensity



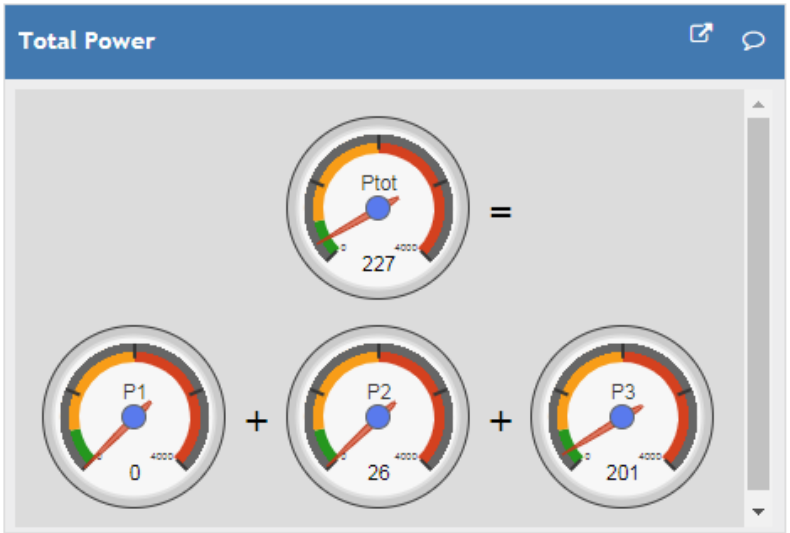



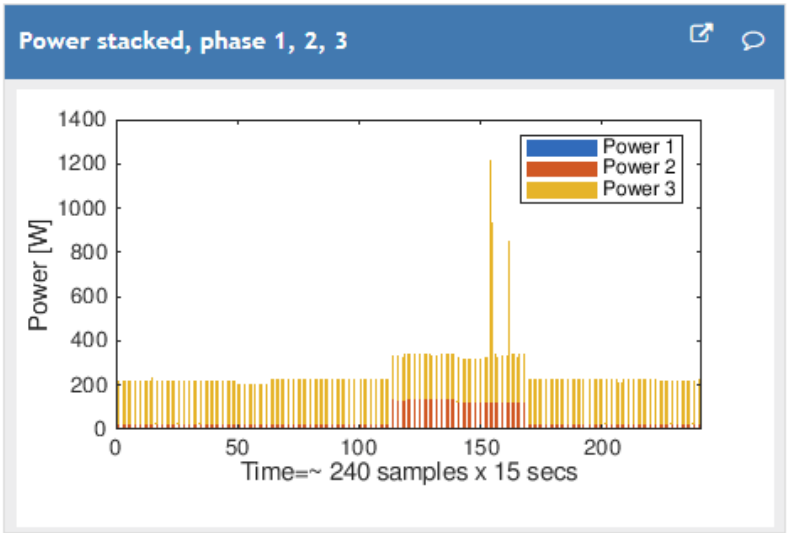



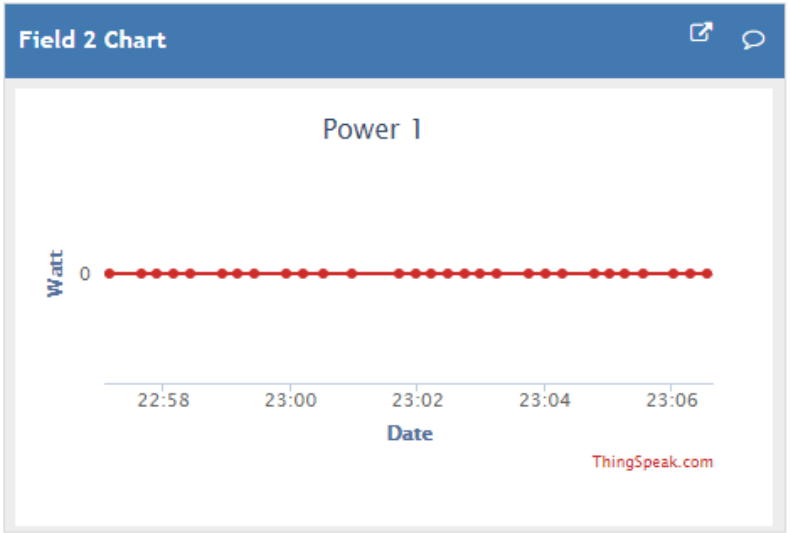



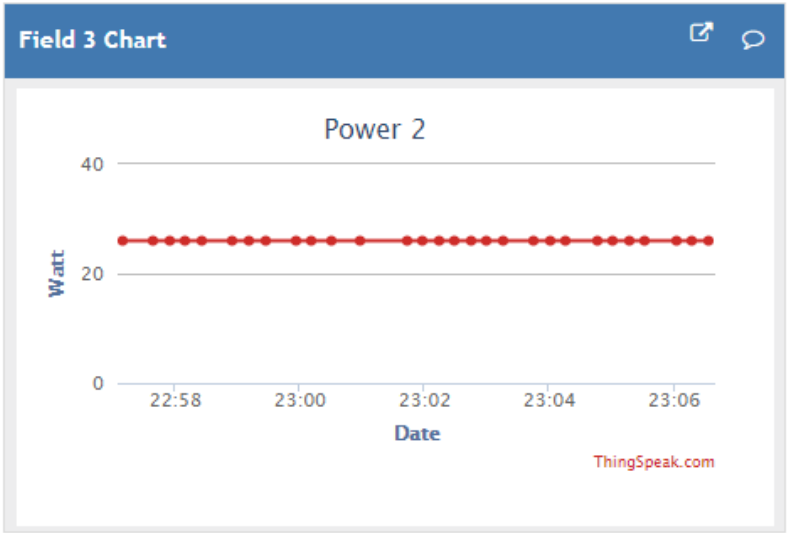
https://thingspeak.com/channels/45572

# Example - Energy monitoring



# ThingSpeak limitation!?

| | FREE<br>For small non-commercial projects | ACADEMIC<br>For academic use by faculty, staff, or researchers at degree-granting institutions[1] |
|---|---|---|
| Scalable for larger projects | ✖<br>No. Annual usage is capped. | ✔ |
| Number of messages | 3 million/year<br>(~8,200/day)[2] | 33 million/year per unit<br>(~90,000/day per unit)[2] |
| Message update interval limit | Every 15 seconds | Every second |
| MATLAB Compute Timeout | 20 seconds | 60 seconds |
| Number of simultaneous MQTT subscriptions | Limited to 3 | 50 per unit |
| Private channel sharing | Limited to 3 shares | Unlimited |
| Technical Support | Forum | Standard MathWorks support |

https://thingspeak.com/prices/thingspeak_academic

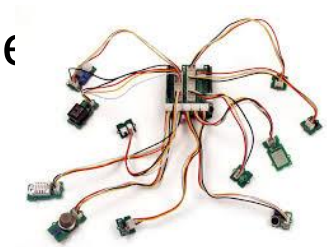
# ThingSpeak Introduction

Create account

- ThingSpeak
  - https://thingspeak.com/

# 1. Thingspeak

- Goal: This example shows you how to **control sensor by Thingspeak** with a string.
  Your device will connect to that channel and get command periodically.
- Hardware Required
  - Raspberry PI
  - Internet (a wired or wireless network)
- Software required
  - ThingSpe



with a Channel ID and API key

# 1. Thingspeak

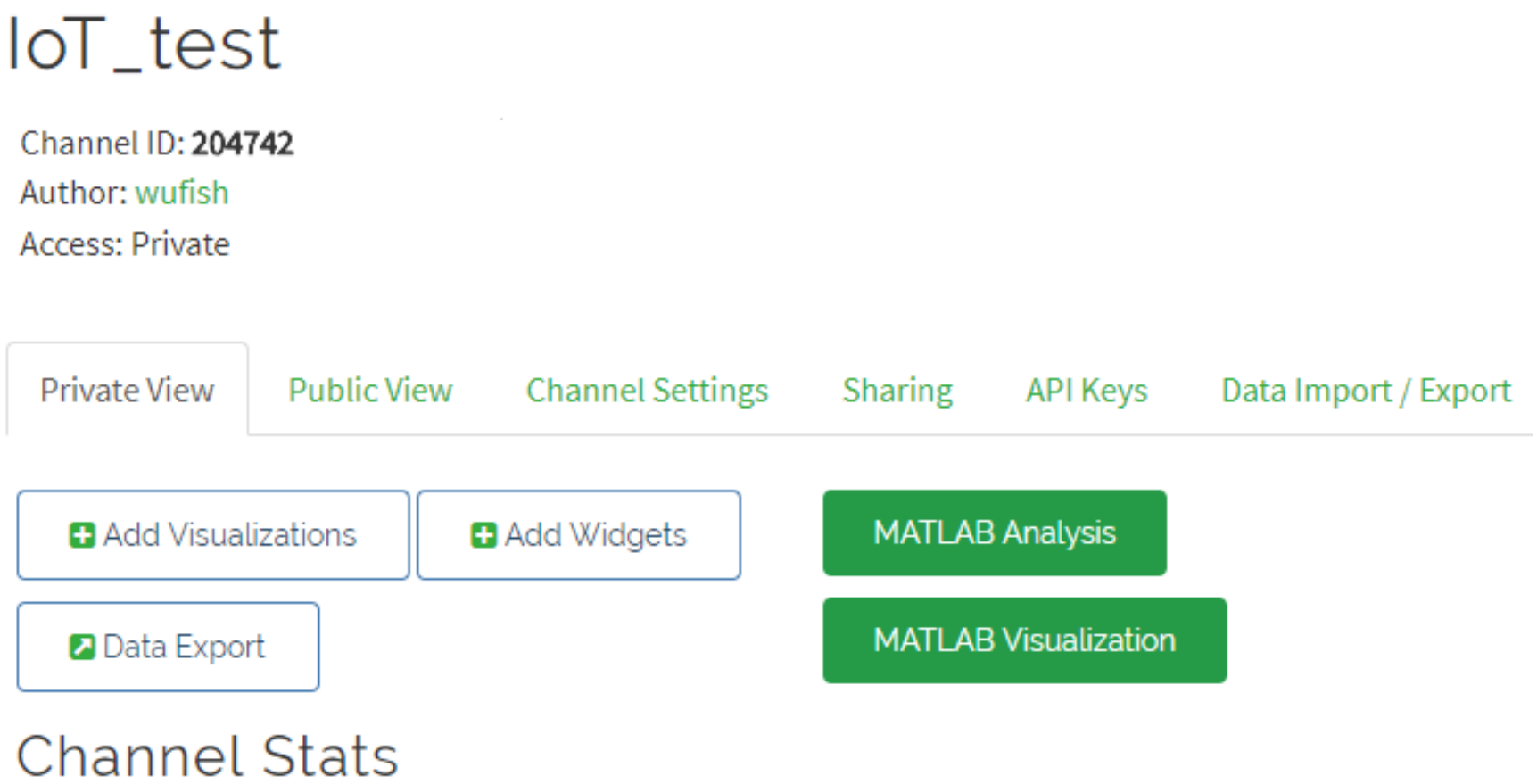
- Steps
  - Sign Up for New User Account https://www.thingspeak.com/users/new
  - Create a **new Channel**
    - selecting Channels and then Create New Channel
  - Create a **TalkBack**
    - selecting Apps, TalkBack, and New TalkBack
  - Enter the TalkBack **API Key**
    - in this sketch under "ThingSpeak Settings"
  - Enter the TalkBack **ID**
    - in this sketch under "ThingSpeak Settings"

# 1. Thingspeak

Select “Channel”, create “New Channel”

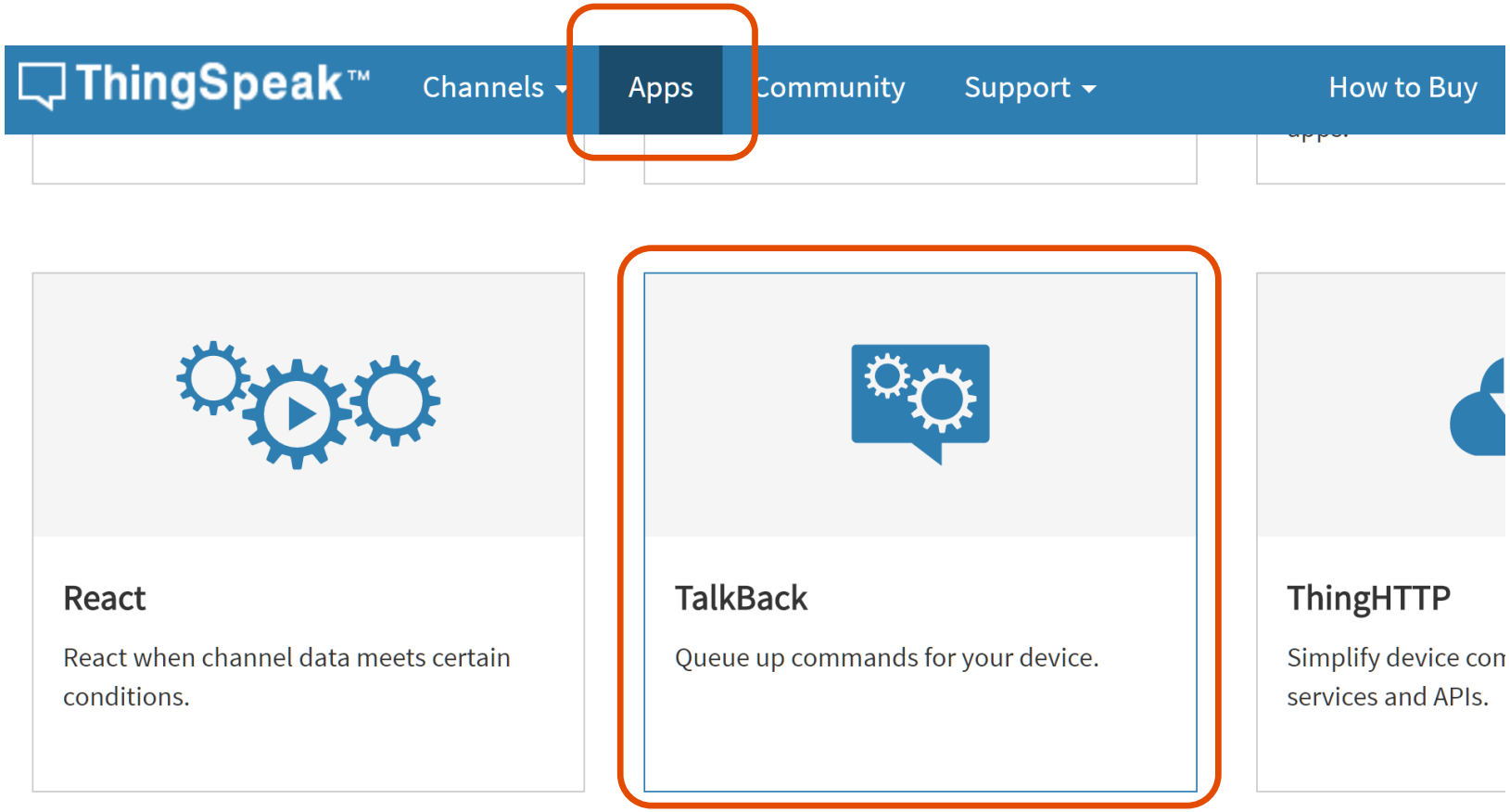
# 1. Thingspeak

A new channel is created:



# 1. TalkBack (Thingspeak)

Select “App”, choose “TalkBack”



# 1. TalkBack (Thingspeak)

In TalkBack:

Apps / TalkBack / LED_on_off / Edit

Name: LED_on_off

API Key: O7FN66PKSX9R3JW4

Log to Channel: IoT_test (204742)

Commands: Add a new command

Save TalkBack

Set “Name”

Connect to “Channel”

# 1. TalkBack (Thingspeak)

In TalkBack:

Apps / TalkBack / LED_on_off

Edit TalkBack

| | |
|---|---|
| Name: | LED_on_off |
| TalkBack ID: | 12315 |
| API Key: | O7FN66PKSX9R3JW4 |
| | Regenerate API Key |
| Created: | 2016-12-19 2:30 pm |
| Logged to Channel: | IoT_test |

Commands

Add a new command

Commands

| Position | Command ID | Command string |
|---|---|---|
| 1 | 12862314 | ON |

| Position | Command string | |
|---|---|---|
| | ON | Save |

Commands

| Position | Command string | |
|---|---|---|
| | ON | Save |

Command = ON or OFF

# 1. TalkBack (Thingspeak)

## Execute the Next TalkBack Command

```
POST https://api.thingspeak.com/talkbacks/12315/commands/execute.json
     api_key=BV76T8WDCJLQCQF7
```

TalkBack ID: 12315
API Key: O7FN66PKSX9R3JW4
**(use your own ID and key)**

Apps / TalkBack / LED_on_off

Edit TalkBack

| | |
|---|---|
| Name: | LED_on_off |
| TalkBack ID: | 12315 |
| API Key: | O7FN66PKSX9R3JW4 |
| | Regenerate API Key |
| Created: | 2016-12-19 2:30 pm |
| Logged to Channel: | IoT_test |

# 1. Check TalkBack

Commands

| Position | Command ID | Command string |
|---|---|---|
| 1 | 12858155 | OFF |

- Check Command by POSTMAN (Chrome

POST https://api.thingspeak.com/talkbacks/12315/commands/execute.json Params Send

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

| Key | Value | Description |
|---|---|---|
| ☑ api_key | BV76T8WDCJLQCQF7 | |
| New key | Value | Description |

Body Cookies Headers (16) Test Results Status: 200 OK

Pretty Raw Preview JSON


```
{
    "id": 12858155,
    "command_string": "OFF",
    "position": null,
    "executed_at": "2018-05-24T15:51:57Z",
    "created_at": "2018-05-24T15:51:47Z"
}
```


**Response message**

# 1. Python for TalkBack

HTTP Methods
- GET
- POST
- PUT
- HEAD
- DELETE
- PATCH
- OPTIONS

- Install python library: **requests**
  - sudo apt-get install python-pip
  - sudo pip install requests

- Usage:
  - r = requests.get('https://api.github.com/events')
  - r = requests.post('http://httpbin.org/post', data = {'key':'value'})
  - r = requests.put('http://httpbin.org/put', data = {'key':'value'})
  - r = requests.delete('http://httpbin.org/delete')
  - r = requests.head('http://httpbin.org/get')
  - r = requests.options('http://httpbin.org/get')

http://docs.python-requests.org/zh_CN/latest/user/quickstart.html
https://www.w3schools.com/tags/ref_httpmethods.asp

# 1. Python for TalkBack

Execute the Next TalkBack Command

```
POST https://api.thingspeak.com/talkbacks/12315/commands/execute.json
     api_key=BV76T8WDCJLQCQF7
```

```
import requests

# Execute the Next TalkBack Command
response = requests.post(\
'https://api.thingspeak.com/talkbacks/12315/commands/execute.json', \
data = {'api_key':'BV76T8WDCJLQCQF7'})
response = response.text

pos = -1
for i in range(2):
    pos = response.find(":", pos+1)

start = pos + 2
end = response.find('"', start)
command = response[start:end]

print(command)    # only print command
```

{"id":12862195,"command_string":"**ON**","position":null,"executed_at":"2018-05-25T01:42:02Z","created_at":"2018-05-25T01:41:58Z"}

Response message

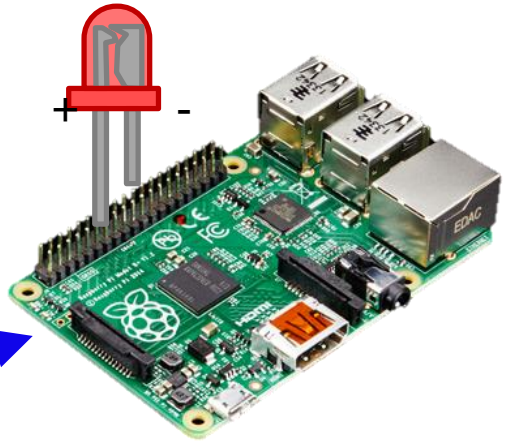
# Quiz 1

- Use TalkBack to turn on/off the LED on Raspberry PI

Apps / TalkBack / LED_on_off

Edit TalkBack

| | |
|---|---|
| Name: | LED_on_off |
| TalkBack ID: | 12315 |
| API Key: | O7FN66PKSX9R3JW4 |
| | Regenerate API Key |
| Created: | 2016-12-19 2:30 pm |
| Logged to Channel: | IoT_test |



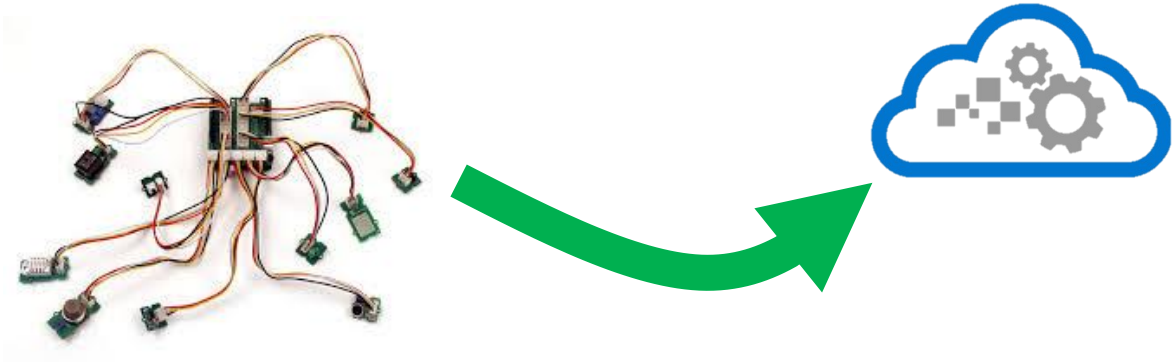

Commands

Add a new command

Set LED state

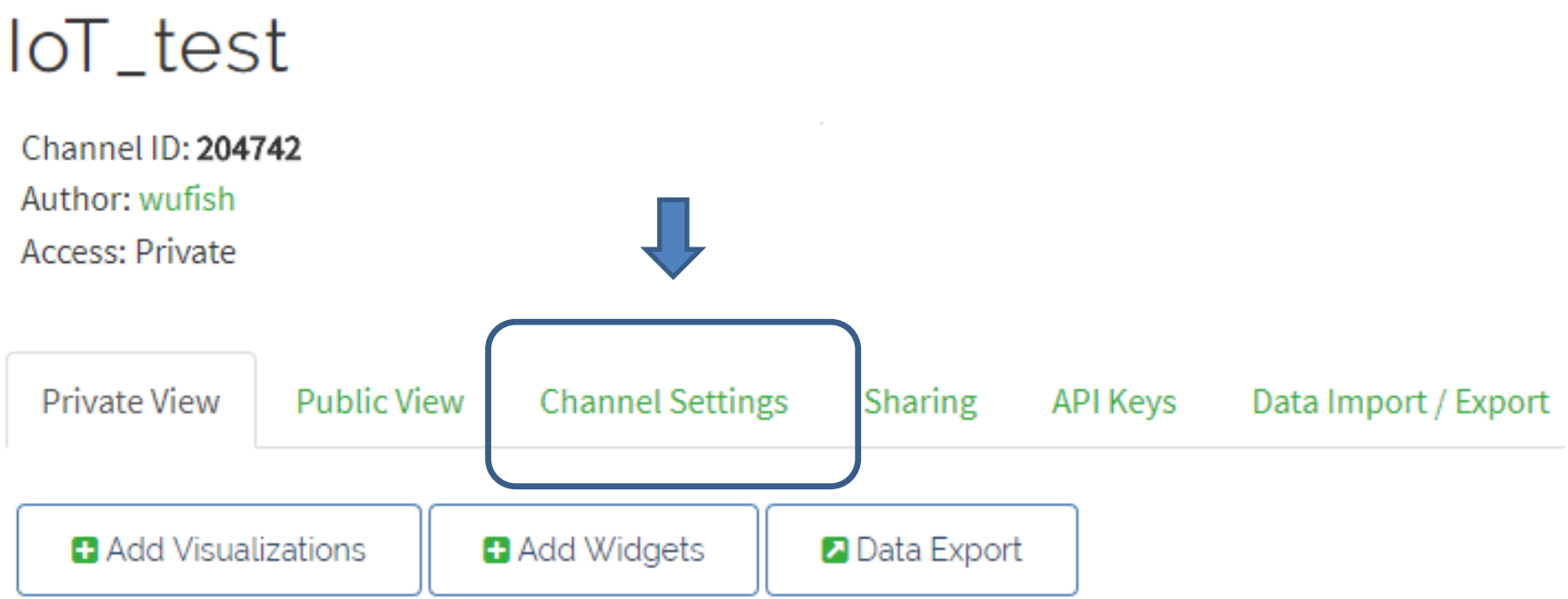
# 2. Device -> ThingSpeak

- Goal: This example shows you how to **upload data to Thingspeak**. Your device will connect to that channel and send data.
- Hardware Required
  - Raspberry PI
  - Internet (a wired or wireless network)
- Software required
  - ThingSpeak account, with a Channel ID and API key



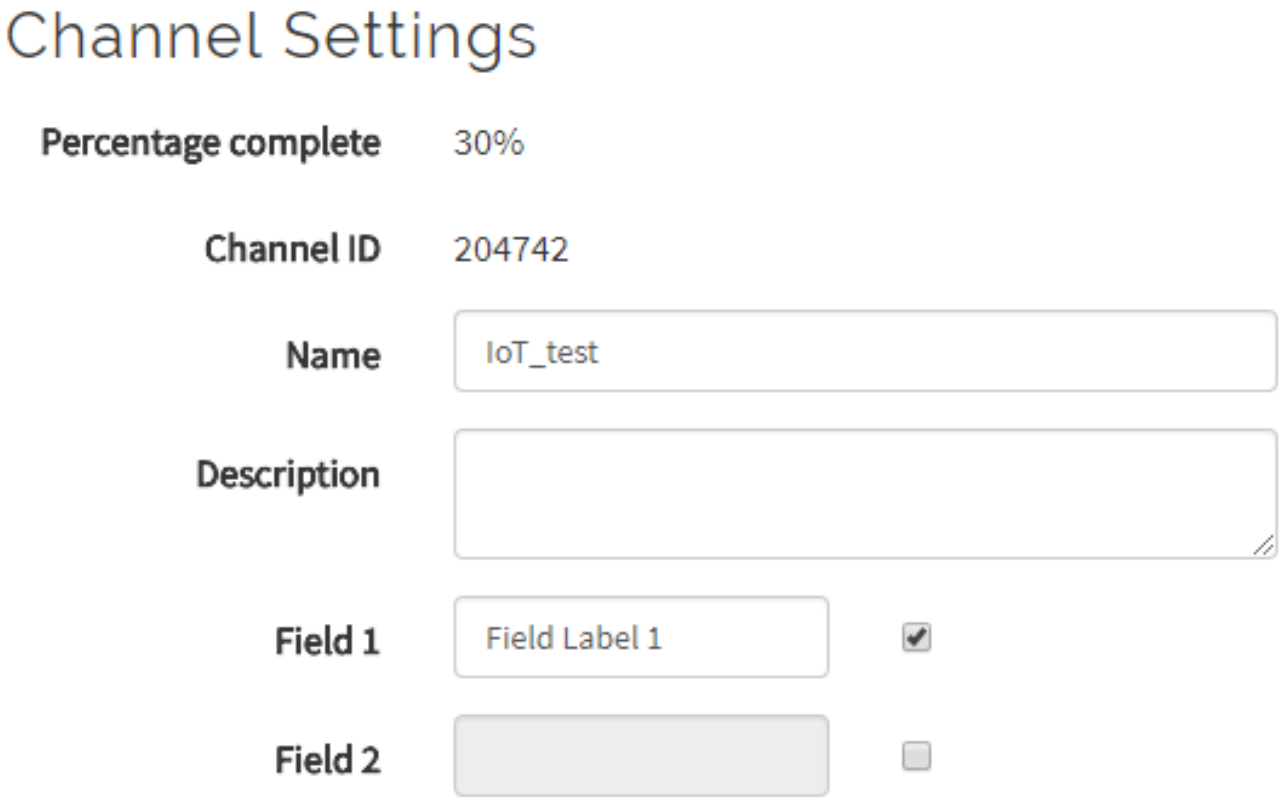
# 2. Device -> ThingSpeak

- Go to “Channel Settings”



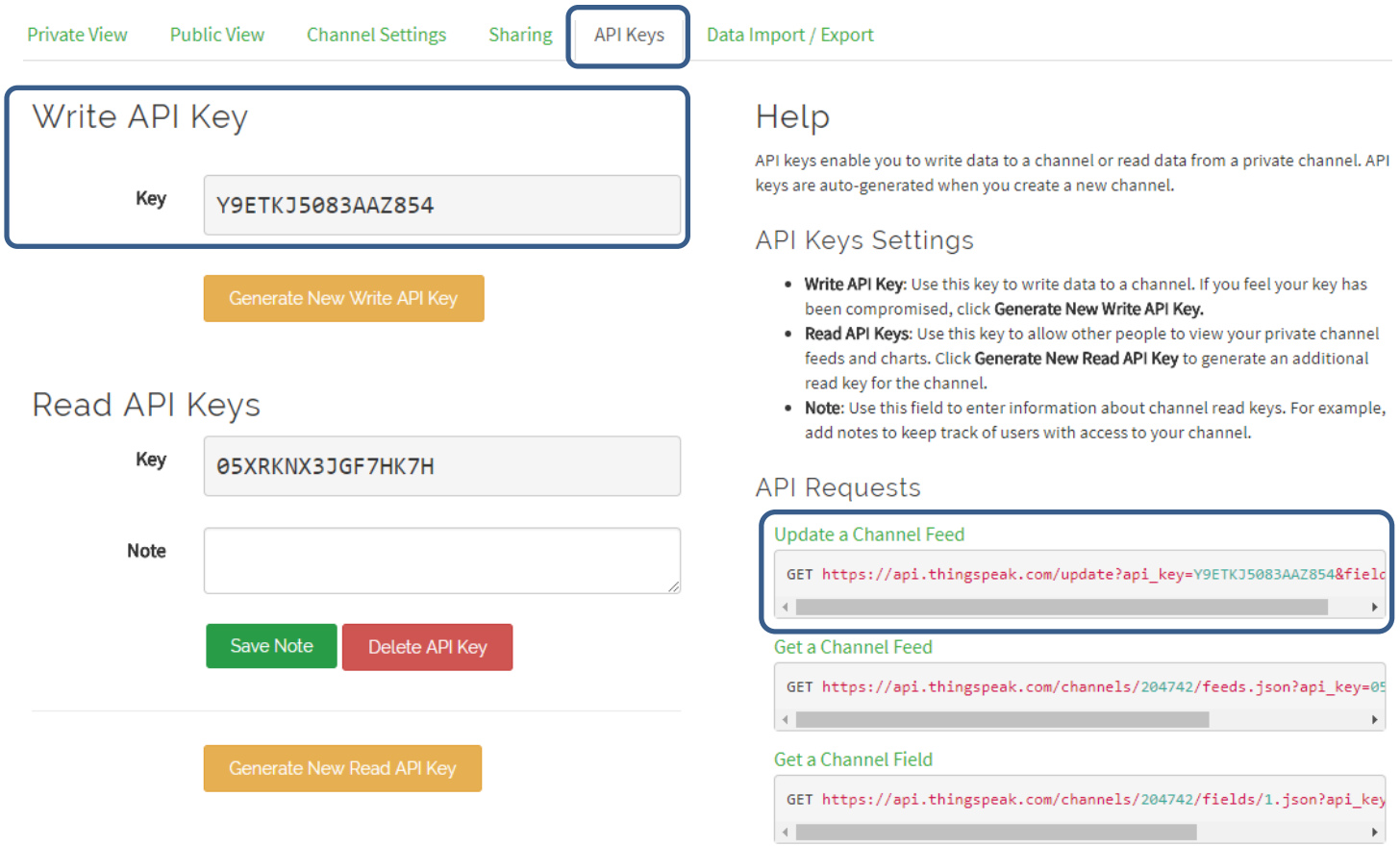
# 2. Device -> ThingSpeak

- Set the number of filed and their name
  - Ex: Filed 1 = Temperature



# 2. Device -> ThingSpeak

- Use your own "**Write API Key**"
  - It is different from Talkback API key



# 2. Python for uploading

```
import requests

sensor_value=30

url =
"https://api.thingspeak.com/update?api_key=Y9ETKJ5083AAZ854&field1
="+str(sensor_value)

response = requests.request("GET", url)

print(response.text)    # print response message
```

Update a Channel Feed

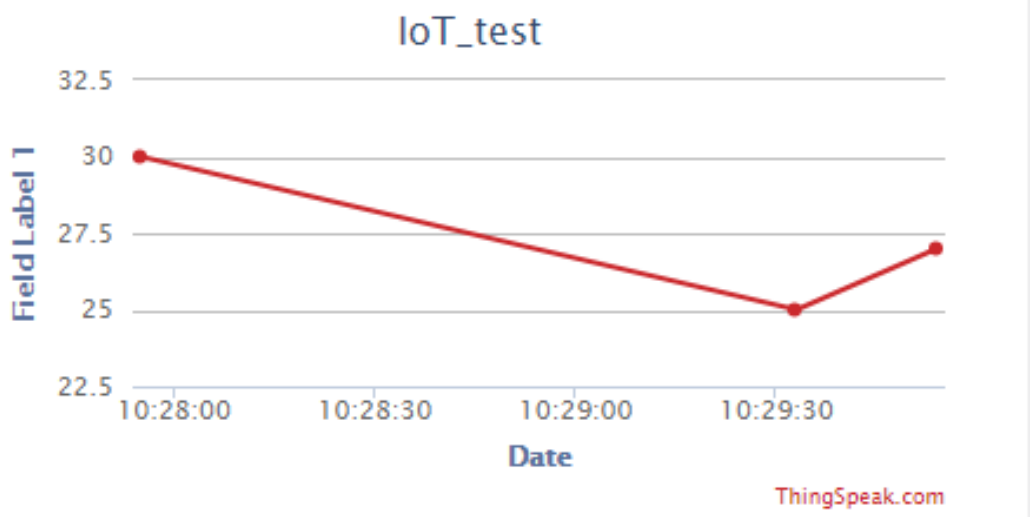
GET https://api.thingspeak.com/update?api_key=**Y9ETKJ5083AAZ854**&field1=**0**

# 2. ThingSpeak

## Channel Stats

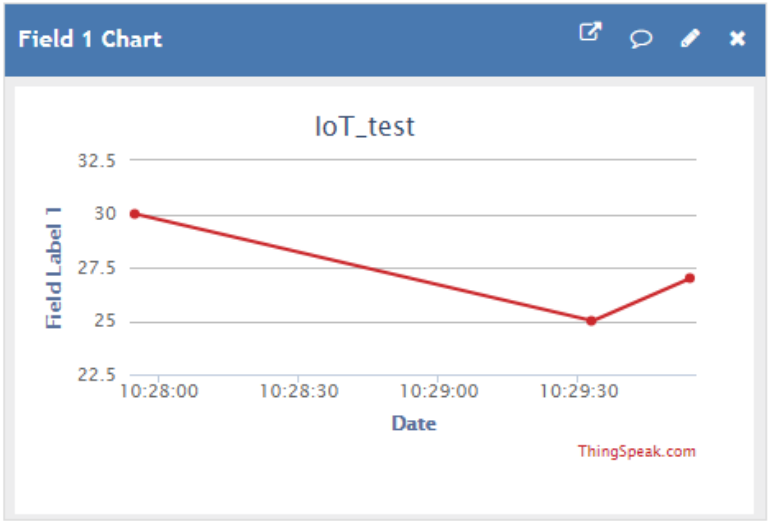
Created: about a year ago
Updated: less than a minute ago
Last entry: less than a minute ago
Entries: 3

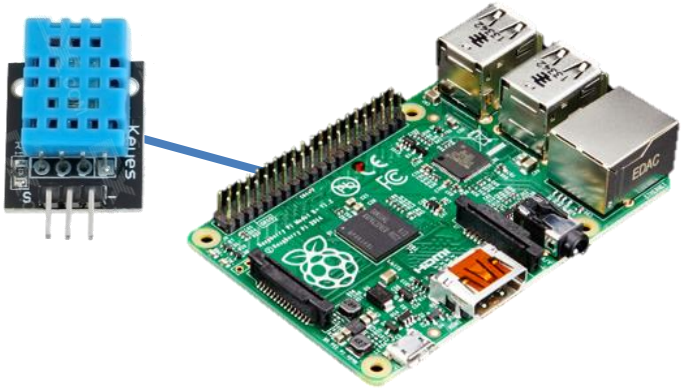
**Field 1 Chart**



# Quiz 2

- Connect DHT11 on PI, upload temperature to you channel.



# Summary

- Create a ThingSpeak account
- Practice Thingspeak (Talkback and upload data)

- Write code for Quiz 1 - 2, then demonstrate it to TAs
  - Quiz1: Use TalkBack to turn on/off the LED on Raspberry PI
  - Quiz2: Upload DHT11 to Thingspeak

# Ref: Cloud platform

- Thingspeak
  - https://thingspeak.com/
- MediaTek Cloud Sandbox
  - https://mcs.mediatek.com/zh-TW/
- Microsoft Azure IoT Suite
  - https://azure.microsoft.com/zh-tw/features/iot-accelerators/
- Amazon AWS
  - https://aws.amazon.com/tw/iot-core/features/
- restdb.io
  - https://restdb.io/
- IFTTT
  - https://ifttt.com/

# Reference

- 氣壓計 datasheet
  - http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- Filter 介紹
  - https://en.wikipedia.org/wiki/Digital_filter
  - http://stackoverflow.com/questions/4611599/smoothing-data-from-a-sensor
  - http://blog.thomnichols.org/2011/08/smoothing-sensor-data-with-a-low-pass-filter

# Reference

- Raspberry Pi IoT無線傳輸技術介紹 – Bluetooth
  - https://www.slideshare.net/raspberrypi-tw/raspberry-pi-iot-bluetooth
- Eddystone
  - https://github.com/google/eddystone
- Eddystone Protocol Specification
  - https://github.com/google/eddystone/blob/master/protocol-specification.md
- Eddystone-URL Beacon Implementations
  - https://github.com/google/eddystone/tree/master/eddystone-url/implementations/

# 補充: fritzing

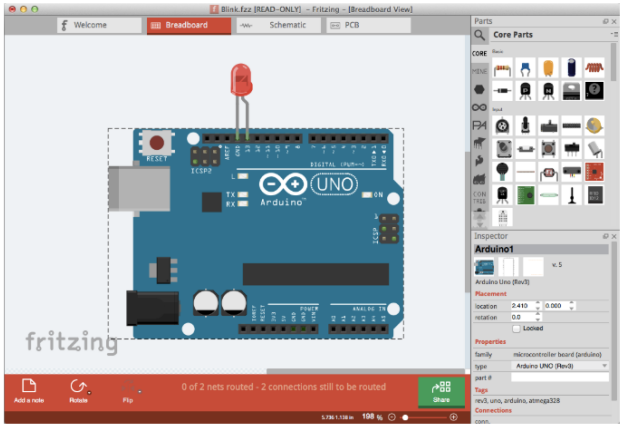
fritzing electronics made easy

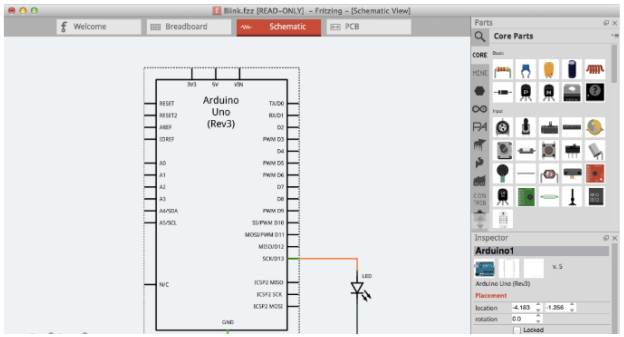
Projects Parts **Download** Learning Services Contribute FORUM FAB

Fritzing is open source, free software. Be aware that the development of it depends on the active support of the community.
Select the download for your platform below.

Version **0.9.3b** was released on **June 2, 2016**.

- Windows 32 bit
- Windows 64 bit
- Mac OS X 10.7 and up
- Linux 32 bit
- Linux 64 bit
- Source Github




http://fritzing.org/download/

# Demo video (class)