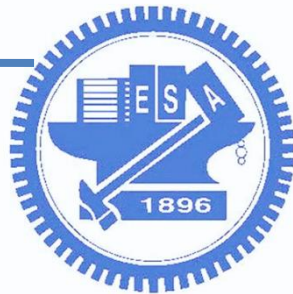


通訊網路實驗

IoT應用 Raspberry PI

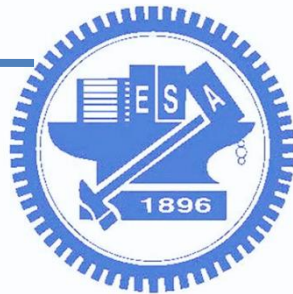
Dept. of Electrical and Computer Engineering (ECE)
National Chiao Tung University



評分標準 & 注意事項

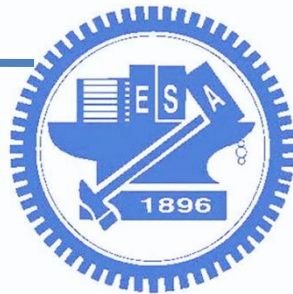
- 出席 45%
- Demo 35% (視情況微調)
- 結報 20%
 - 實驗項目
 - 實驗過程 (截圖、照片)
 - 心得

PDF檔案,期限為1週 (下次上課前一天晚上12點)



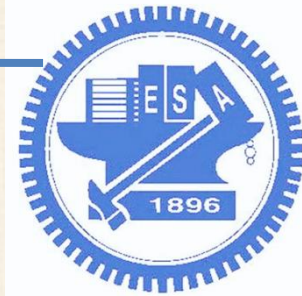
課程大綱

- 1. 認識Raspberry Pi
- 2. GPIO介紹與感測器應用
 - LED + 溫溼度感測器 + 超音波感測器



Demo項目

- Q1：用LED產生SOS的摩斯密碼
- Q2：設計一個溫度警示燈
- Q3：設計一個距離警示燈
- Q4：合併Q2,Q3 (5%)



Introduction

- 信用卡大小般的電腦



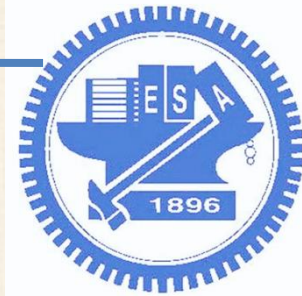
<http://www.flickr.com/photos/fotero/7697063016/>



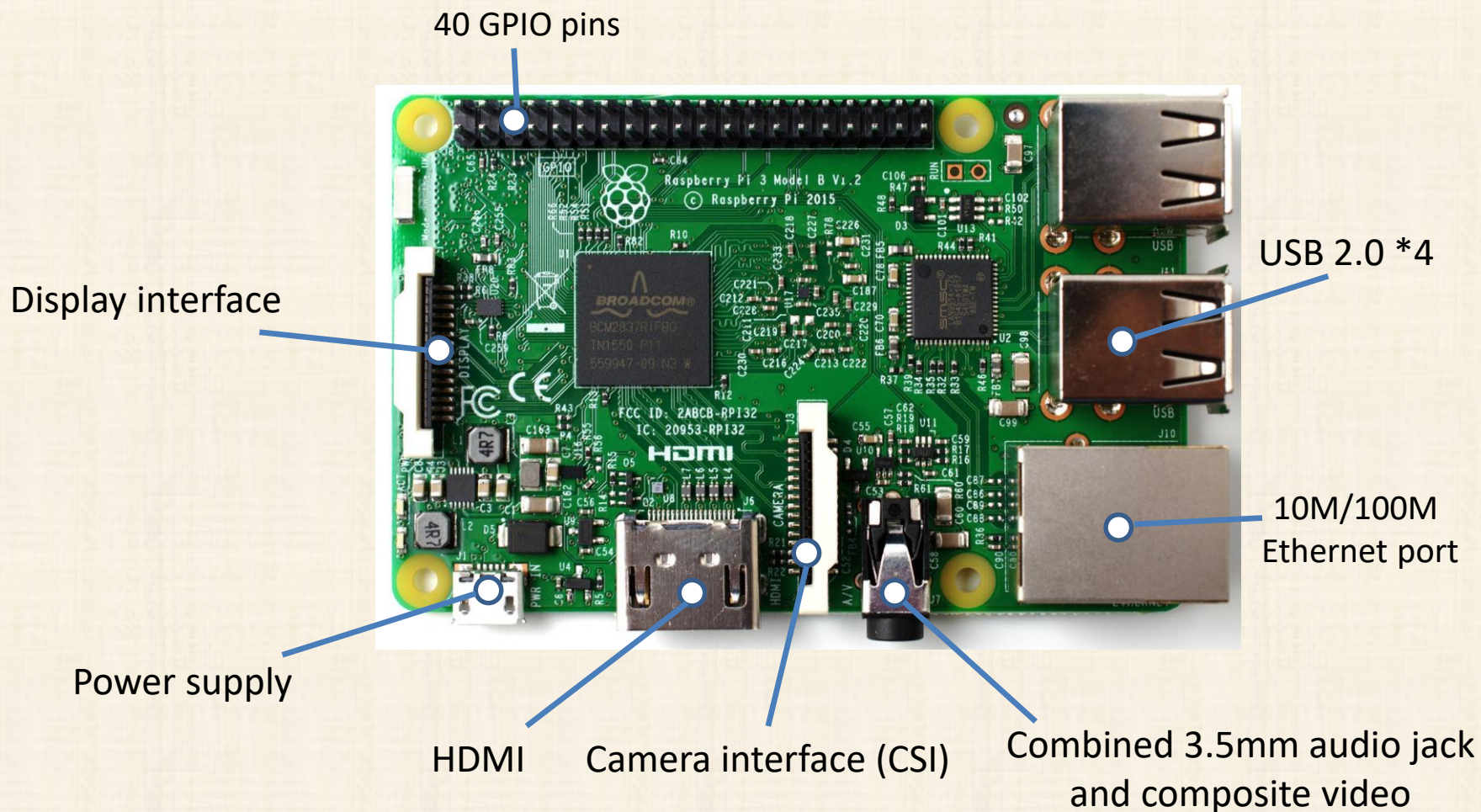
Introduction

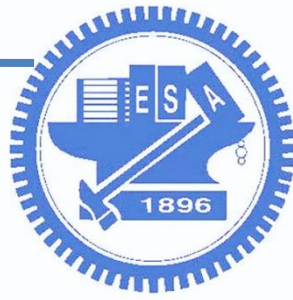
- Raspberrypi.org 基金會所設計開發
- 以低價硬體 & 自由軟體刺激電腦科學教育
- 公開的 datasheet & 線路圖 & 原始碼
- 由 Element14/RS Components/Egoman 生產製造





Raspberry Pi 硬體週邊

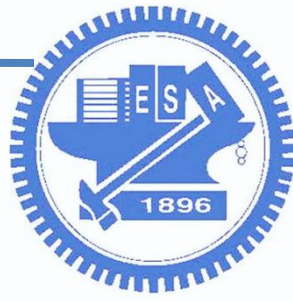




Introduction

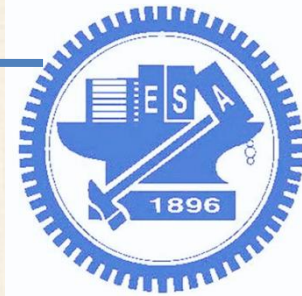
□ Raspberry Pi:

- 源自一個開放源程式碼的硬體專案平台
 - 一塊具備簡單I/O功能的電路板
 - Broadcom BCM2836 system on a chip (SoC) 的ARM晶片
 - 一大堆的Linux 軟體
- 可以用來開發交互產品
 - 它可以讀取大量的開關和感測器信號，並且可以控制電燈、電機和其他各式各樣的物理設備
 - 也可以開發出與PC一樣的周邊裝置,也可以運行在Linux PC 上的軟體進行通信
- 程式開發環境的軟體
 - 從網上免費下載與使用



Raspberry Pi 可以用在?

- 桌機 (Raspbian)
 - FTP, Web, NAS, AP, 自動化控制...等
- 多媒體影音作業系統
 - OSMC (Open-Source Media Center)
 - OpenELEC (Open Embedded Linux Entertainment Center)
- 遊戲機 (RetroPie 、 PiPlay)
- 網站滲透測試 (Kali Linux)
- Android系統 (RaspAnd)
- 超級電腦
 - 將一堆PI組裝在一起



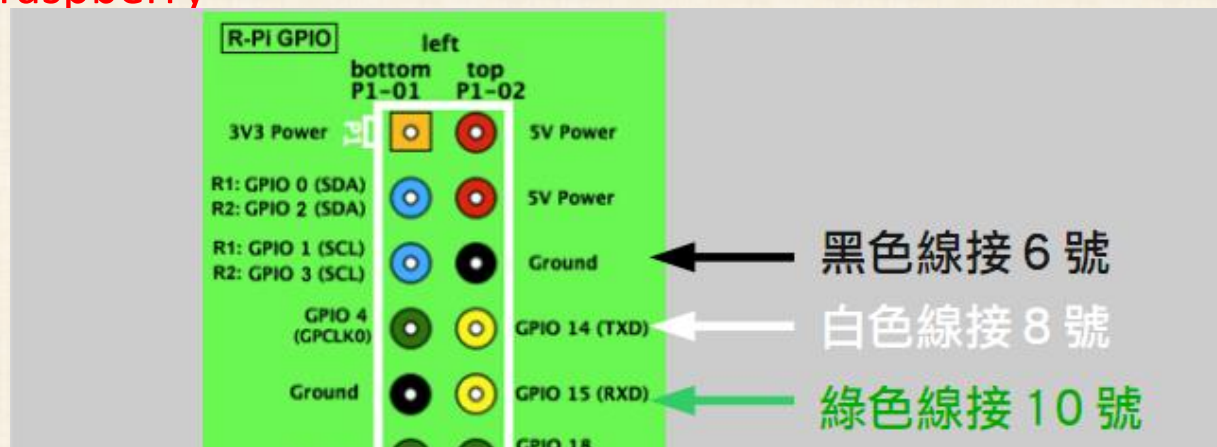
TTL序列連接

- 1. 將SD卡插到Raspberry Pi並開機
- 2. 透過 USB 轉 TTL 序列傳輸線，就可以在不需螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi

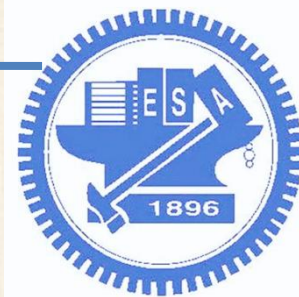
- 預設登入帳密

- ID: pi
- PW: raspberry

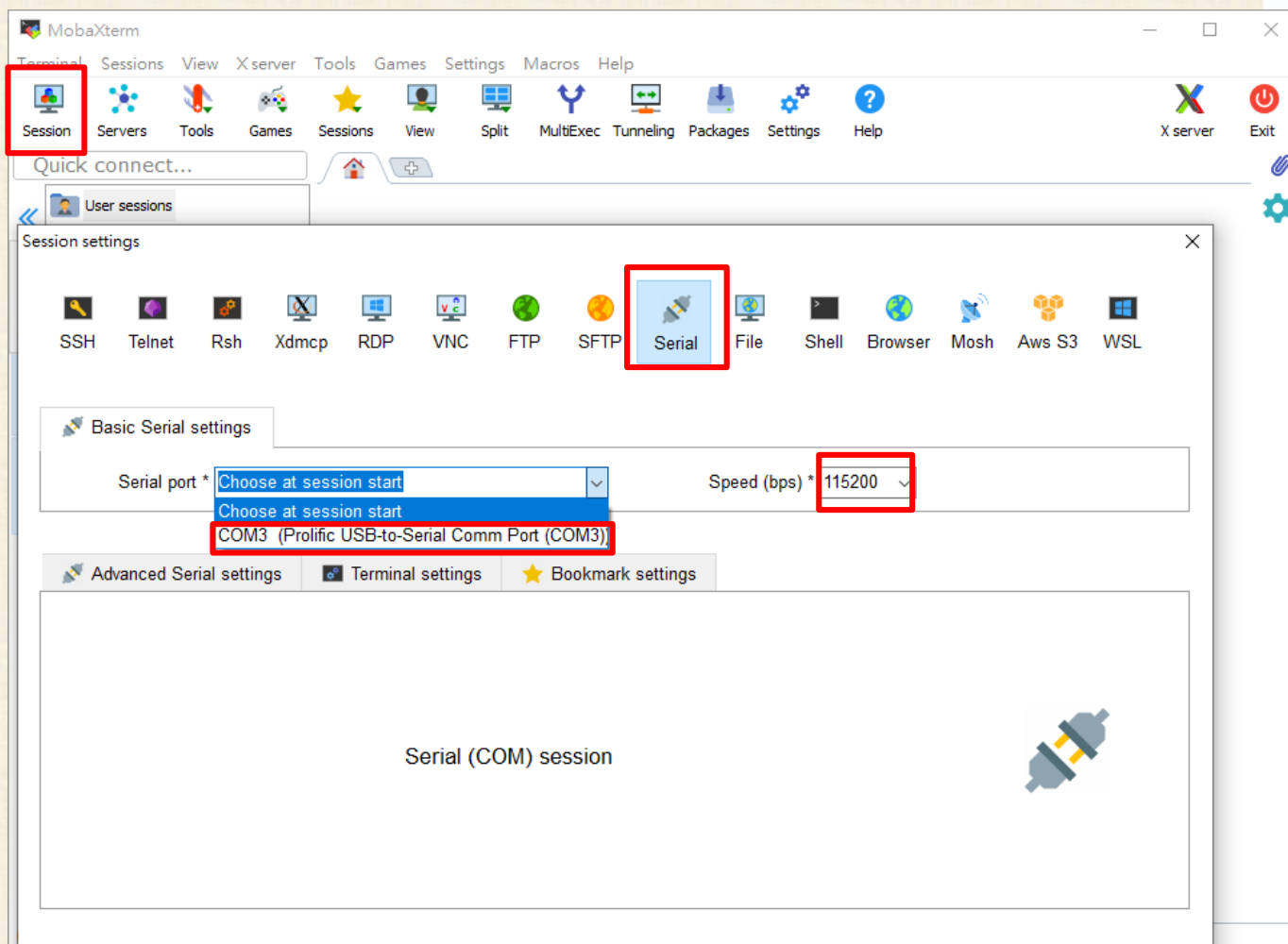
注意接腳

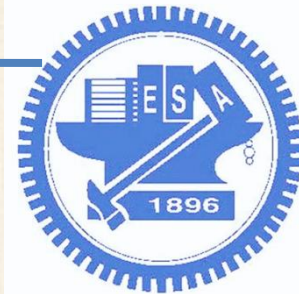


<https://www.raspberrypi.com.tw/tag/usb-to-ttl/>



MobaXterm畫面





COM3 (Prolific USB-to-Serial Comm Port (COM3))

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit

Quick connect...

User sessions

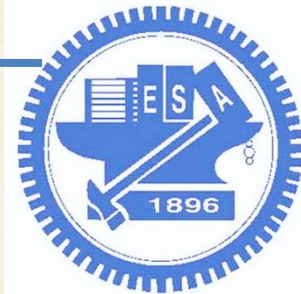
COM3 (Prolific USB-to-Serial

Raspbian GNU/Linux 10 raspberrypi ttyAMA0

raspberrypi login: pi

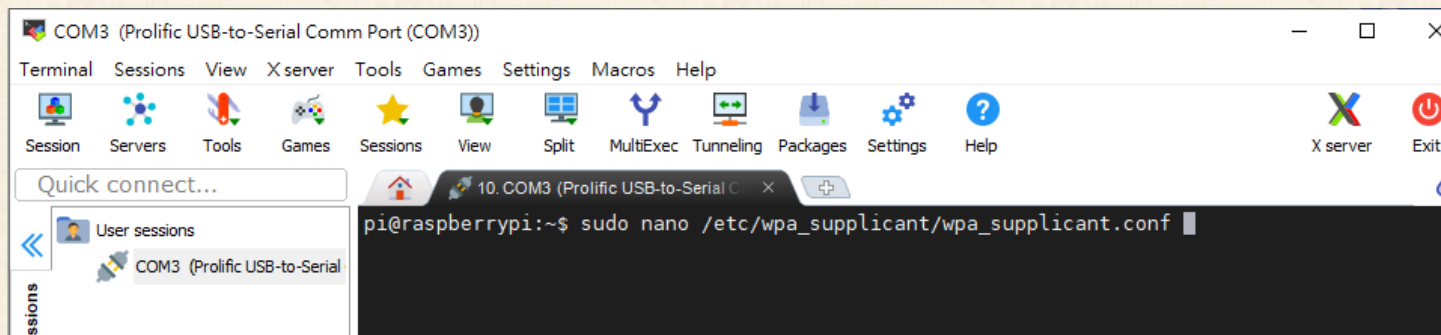
Password:

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



Wi-fi

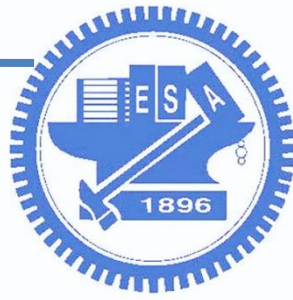
1. 輸入 `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` (編輯設定檔)



2. 填寫SSID與密碼 (新增下面欄位)

```
network={  
    ssid= "your_ap"  
    psk= "your_passwd"  
}
```

3. 重開機
輸入 `sudo reboot`
使用 `ifconfig` 與 `iwconfig` 檢查連線狀態



□ 文字編輯器 nano

□ 編輯結束按 **ctrl + x** 離開

■ 若有變動, 會問你是否存檔, 輸入 **y** 即可

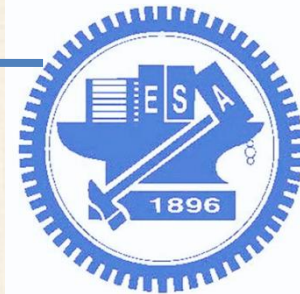
```
GNU nano 3.2 /etc/wpa_supplicant/wpa_supplicant.conf

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

network={
    ssid="BUNLAB"
    psk="7111177117"
}
```

[Read 8 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line



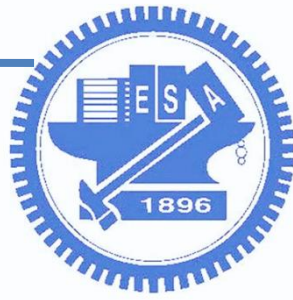
SSH連接

輸入 `sudo raspi-config`

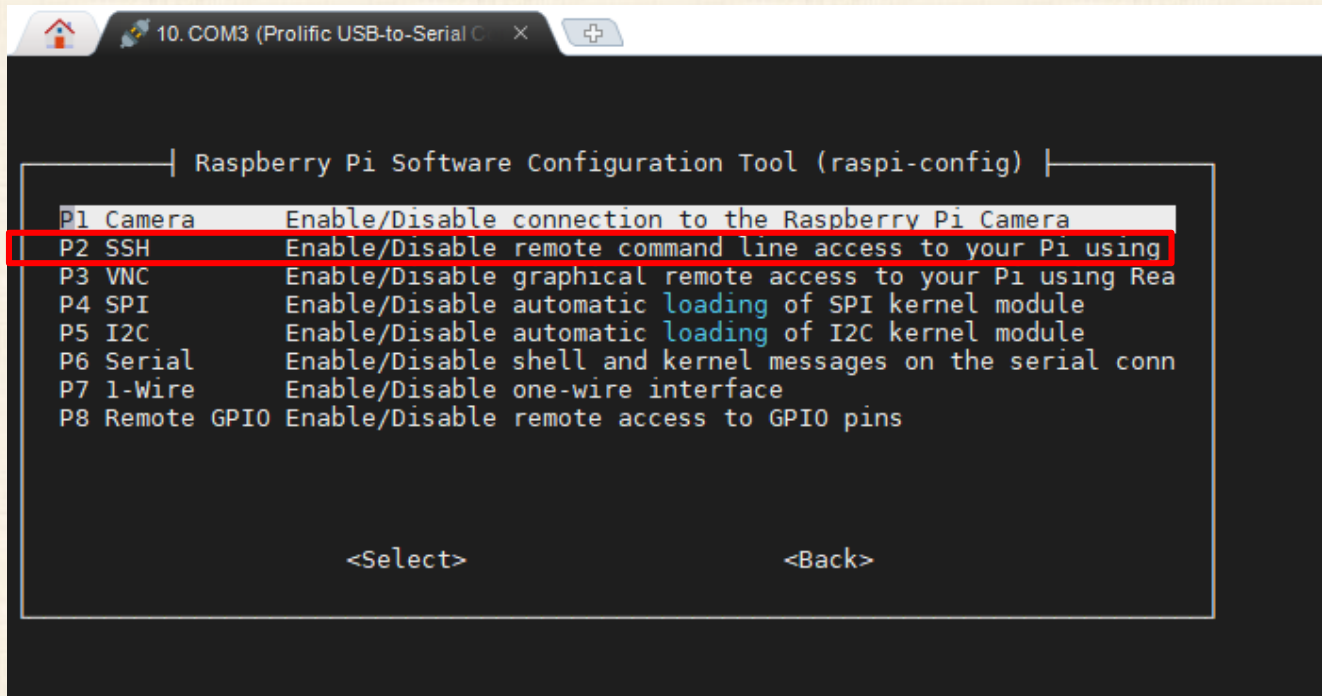
```
10. COM3 (Prolific USB-to-Serial C... x
Raspberry Pi 3 Model B Rev 1.2

| Raspberry Pi Software Configuration Tool (raspi-config) |
1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

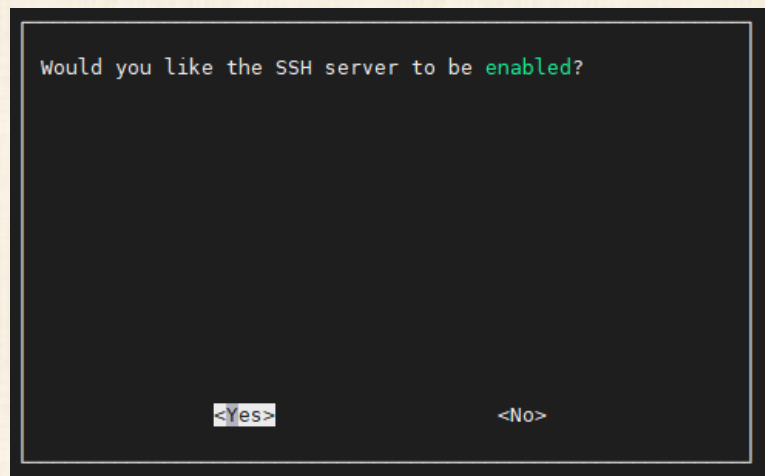
<Select>                <Finish>
```

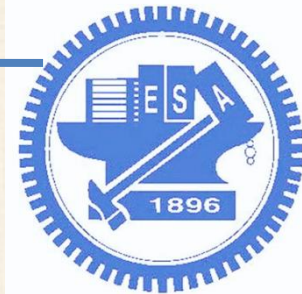


選擇 SSH



選擇 yes





Outline

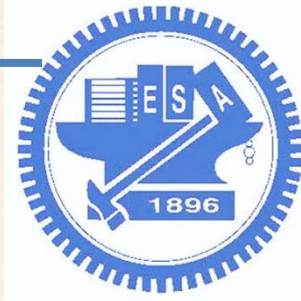
- 0. GPIO介紹, 注意事項
- 1. 安裝LED燈, 電阻的使用方式
- 2. 溫濕度sensor、超音波模組



0. GPIO on Raspberry Pi

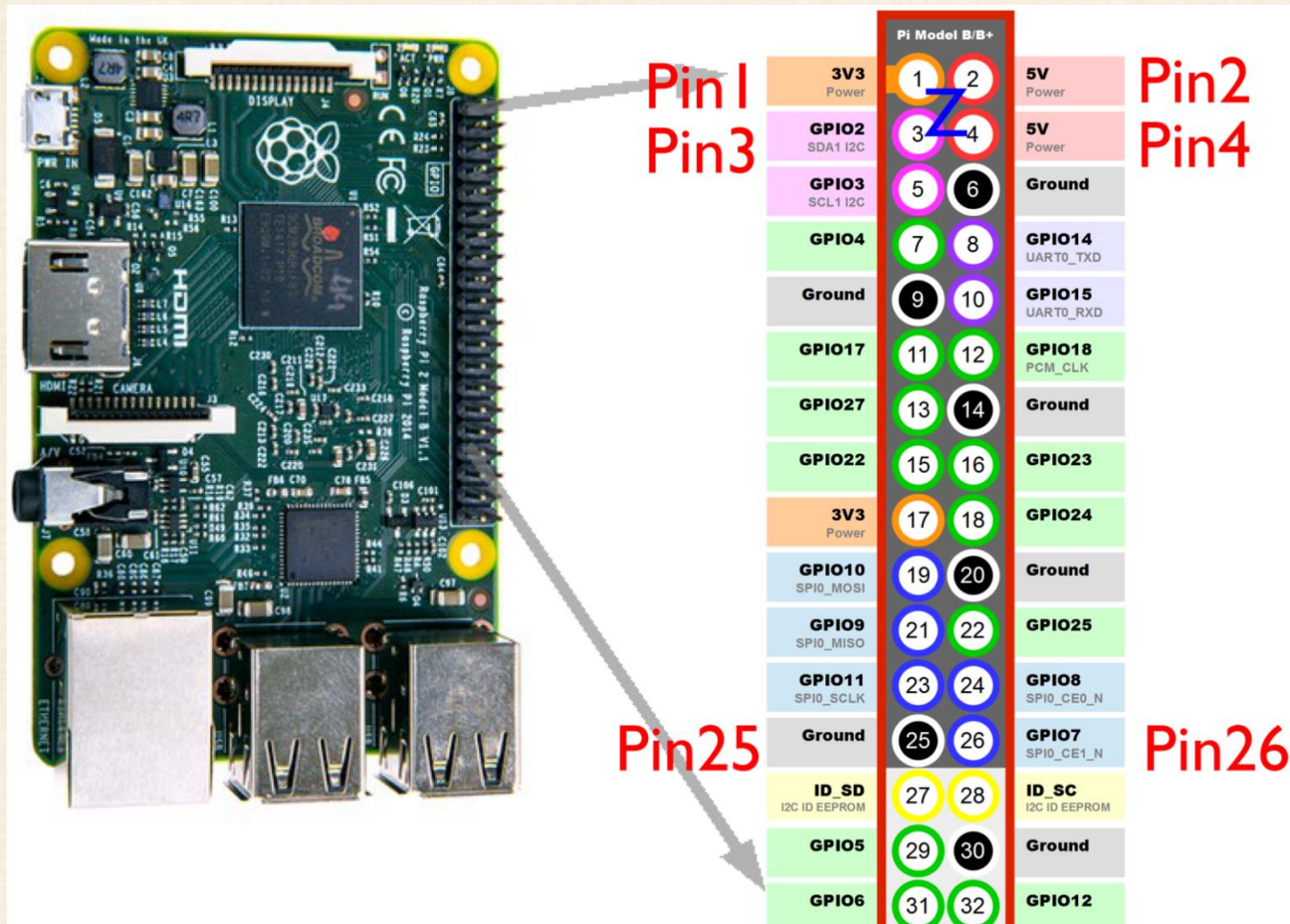
- General-purpose input/output (GPIO)
 - 通用型之輸入輸出(General Purpose I/O)
 - PIN腳可設為輸入(GPI), 輸出(GPO), 輸入與輸出(GPIO)
 - 輸出：寫值到某根腳位; 輸入：從某根腳位讀值
 - 可用軟體控制的數位訊號

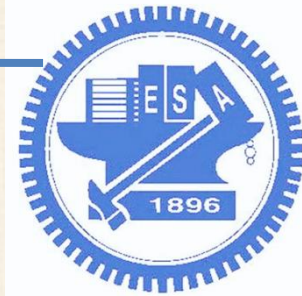




0. GPIO on Raspberry Pi

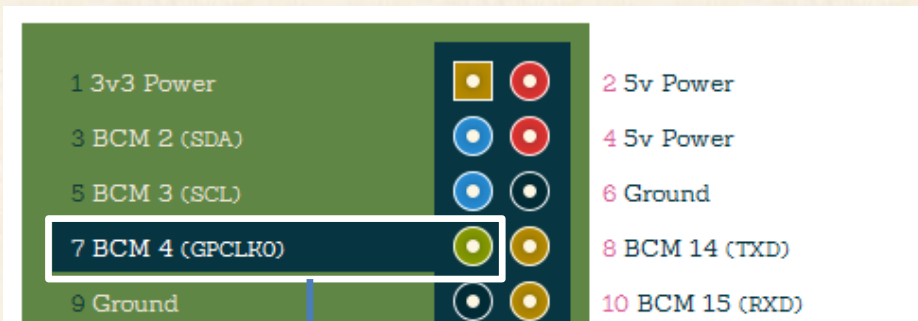
□ Z 字型的腳位編號





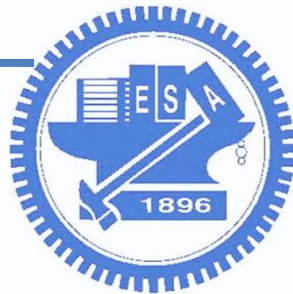
0. GPIO的腳位與應用

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



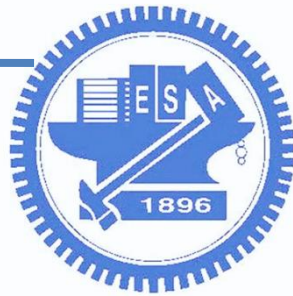
BCM 4:

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



0. GPIO的注意事項

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



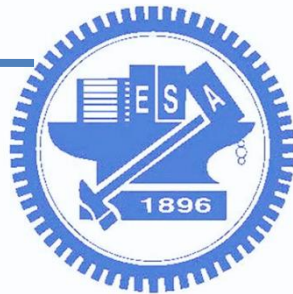
0. 控制 PI 的 GPIO

□ 寫程式

- C
- C + wiringPi
- C#
- Ruby
- Perl
- **Python**
- Scratch
- Java Pi4J Library
- Shell script



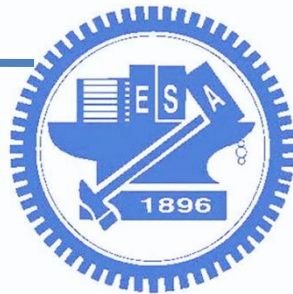
我們用這個



0. Python 簡介

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

- PS. PI預設使用Python 2



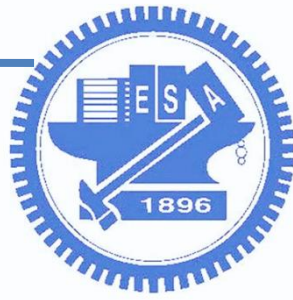
Linux文字編輯

- Nano為Linux的文字編輯套件
 - 1. EX：nano 文件檔名.py
 - 2. 使用方式跟記事本一樣
 - 3. Ctrl + X：離開 (接著會問你是否要存檔)
 - 4. Ctrl + O：調整檔名

- 執行python程式：python filename.py
 - 1. 終止執行：Ctrl + C
 - 2. 若牽涉到系統權限, 需要sudo開頭執行程式
 - Ex: sudo python netexp.py

- 簡單的Linux操作

<http://www.penguintutor.com/raspberrypi/useful-command-reference>



(1). 變數，物件，註解

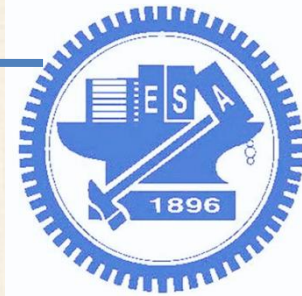
□ 動態型別 (dynamic typing)

```
# 井號是註解
```

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

```
i = [1, 2, 3, 4, 5]  # 變數 i 指到串列物件  
print i[2]          # 印出串列中第三個元素
```

```
i = "abcde"      # 變數 i 指到字串物件  
print i[0]        # 印出字串中第一個元素
```



(2). 模組

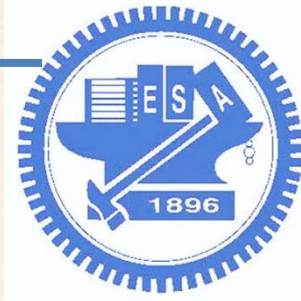
□ 模組

```
# import MODULE  
import RPi.GPIO
```

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

可以使用該模組的函式

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



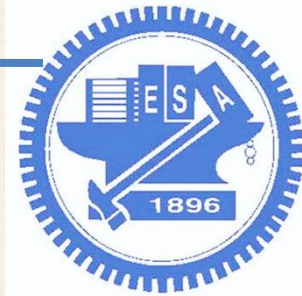
(3). 縮排

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

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



前面有 4 個空白鍵



(4). 迴圈

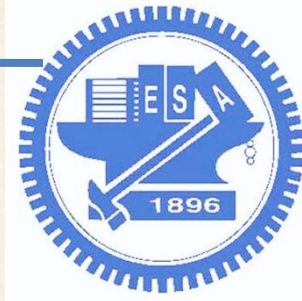
- 自動迭代 (iterator); for loop

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

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



前面有 4 個空白鍵



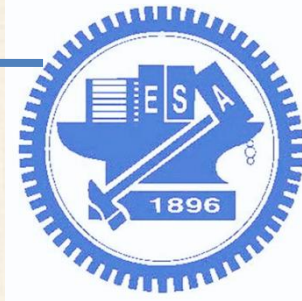
(5). 條件判斷

□ If statement

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



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



(6). 函式

□ 自訂函式 (User-Defined Functions)

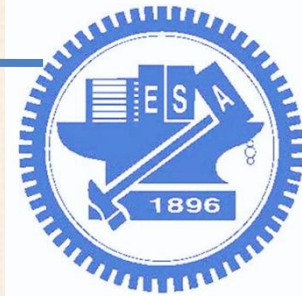
```
def function_name():  
    process  
def function_name(param_name):  
    process  
def function_name(param_name = 3):  
    process
```



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



f(4) 可以得到 17



Python 小提示

- 如果有兩個Statement 要寫在同一行使用(;)分開

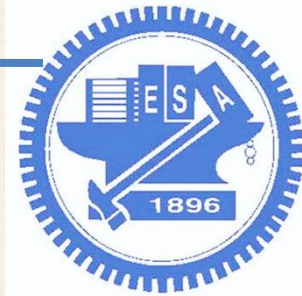
```
>>> print 'hello';print 'runoob';  
hello  
runoob
```

- 一個Statement 要分多行顯示則使用(\)

```
total = item_one + \  
        item_two + \  
        item_three
```

- Python的Statement自動默認換行，不換行需加(,)

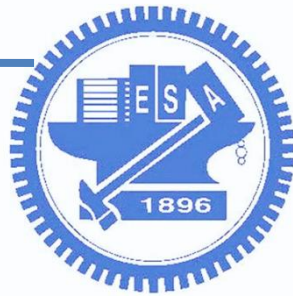
```
# 不換行輸出  
print x,  
print y,  
  
# 不換行輸出  
print x,y
```



Python 小提示

- Python 保留字元，不可拿來當變數名、常數...

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

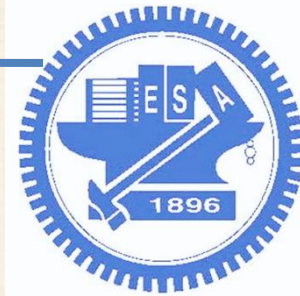


0. 開始寫程式前

□ 先安裝程式&相關函式庫

- `sudo apt-get update`
- `sudo apt-get install -y python-dev python-pip python-rpi.gpio`

1. `-dev (develop)` 開發包
2. `pip` 是一個以 Python 寫成的軟體包管理系統
3. `rpi.gpio` 為樹梅派的 GPIO 函式庫



1. 控制LED燈

□ Python code (LED閃爍)

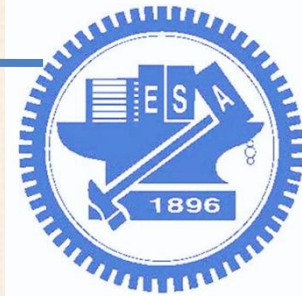
```
1 #!/usr/bin/env python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BOARD)
6 LED_PIN = 12
7 GPIO.setup(LED_PIN, GPIO.OUT)
8
9 while True:
10     print("LED is on")
11     GPIO.output(LED_PIN, GPIO.HIGH)
12     time.sleep(1)
13     print("LED is off")
14     GPIO.output(LED_PIN, GPIO.LOW)
15     time.sleep(1)
16
17 GPIO.cleanup()
```

載入函式庫

GPIO.BOARD:
按照腳位的順序編號

使用NO. 12的pin腳

判斷、控制



1. 控制LED燈

★ □ Q1: 用LED產生SOS的摩斯密碼

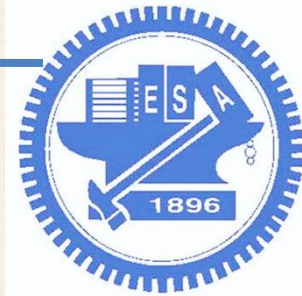
1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

PS. 可設定 one unit = 1 sec

A ● —
B — ● ● ●
C — ● — ●
D — ● ●
E ●
F ● ● — ●
G — — ●
H ● ● ● ●
I ● ●
J ● — — —
K — ● —
L ● — ● ●
M — —
N — ●
O — — —
P ● — — ●
Q — — ● —
R ● — ●
S ● ● ●
T —

U ● ● —
V ● ● ● —
W ● — —
X — ● ● —
Y — ● — —
Z — — ● ●

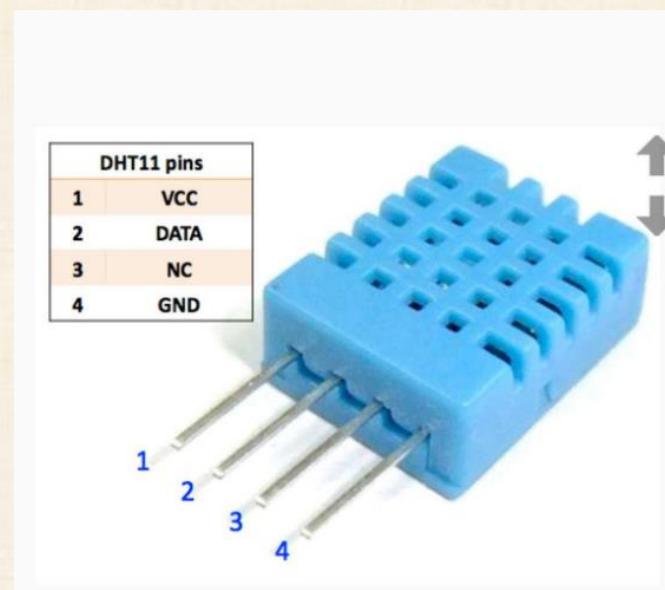
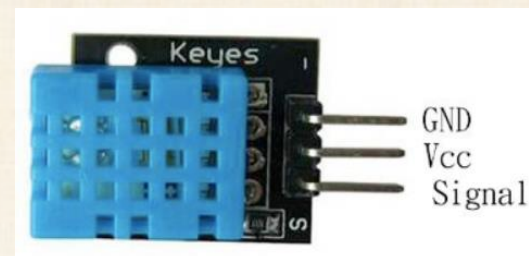
1 ● — — — —
2 ● ● — — —
3 ● ● ● — —
4 ● ● ● ● —
5 ● ● ● ● ●
6 — ● ● ● ●
7 — — ● ● ●
8 — — — ● ●
9 — — — — ●
0 — — — — —

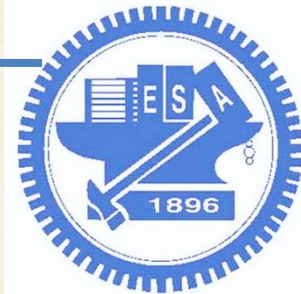


2. 溫濕度sensor

□ DHT11

- 溫濕度感應器
- 溫度: $0\sim 50^{\circ}\text{C}$, 誤差 $\pm 2^{\circ}\text{C}$
- 濕度: $20\sim 90\%$, 誤差 $\pm 5\%$
- 使用三個腳位: Data , VCC , GND (out, + , -)



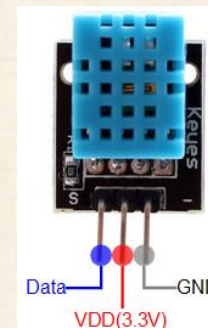


2. 溫濕度sensor

□ 安裝Adafruit的模組與範例程式

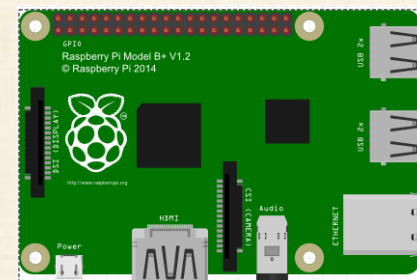
- Source code:

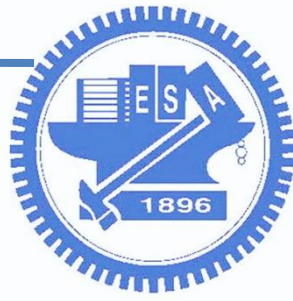
https://github.com/adafruit/Adafruit_Python_DHT



□ 先安裝編譯程式

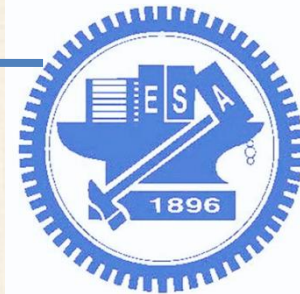
- `sudo apt-get update`
- `sudo apt-get install git-core build-essential python-dev`





2. 溫濕度sensor

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



2. 擷取溫濕度資訊

□ 執行程式

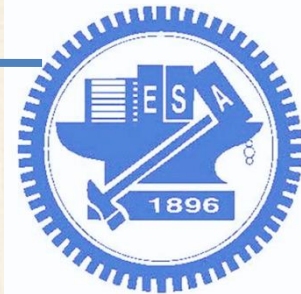
- `$ cd examples`
- `$ sudo ./AdafruitDHT.py 11 4`
 - 使用 DHT 11 感應器 (另有DHT 22, DHT 2302)
 - 連接 GPIO 4 (也就是Pin 7)
 - 得到資訊: Temp=26.0*C Humidity=37.0%

Code第一行是環境變數, 需保留:
`#!/usr/bin/python`

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

□ 調整 溫度 & 濕度 的顯示方式

- Method 1: 修改AdafruitDHT.py, 改為自己想要的格式



2. 擷取溫濕度資訊

□ Python code:

```
import sys
import Adafruit_DHT
```

載入函式庫

```
sensor_args = { '11': Adafruit_DHT.DHT11,
                 '22': Adafruit_DHT.DHT22,
                 '2302': Adafruit_DHT.AM2302 }
```

讀取.py後的參數

```
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
    sensor = sensor_args[sys.argv[1]]
    pin = sys.argv[2]
else:
```

本程式的執行方式 (後面要接2個參數)

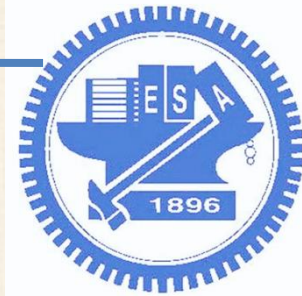
```
    print('usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#')
    print('example: sudo ./Adafruit_DHT.py 2302 4 - Read from an AM2302 connected to GPIO #4')
    sys.exit(1)
```

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

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

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

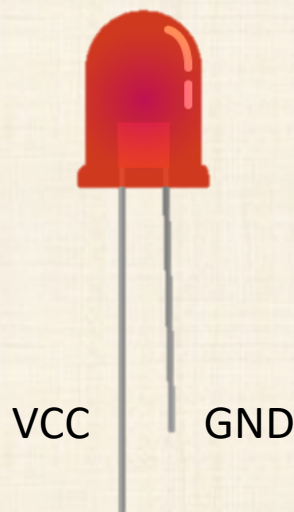
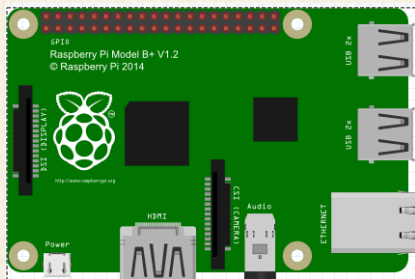
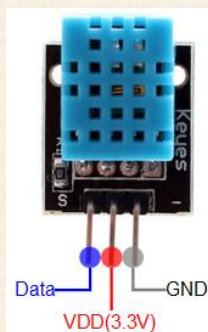
回傳數值

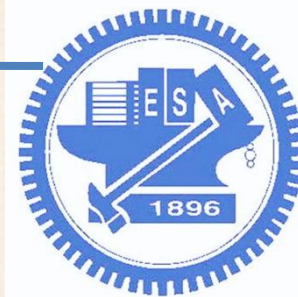


2. 擷取溫濕度資訊

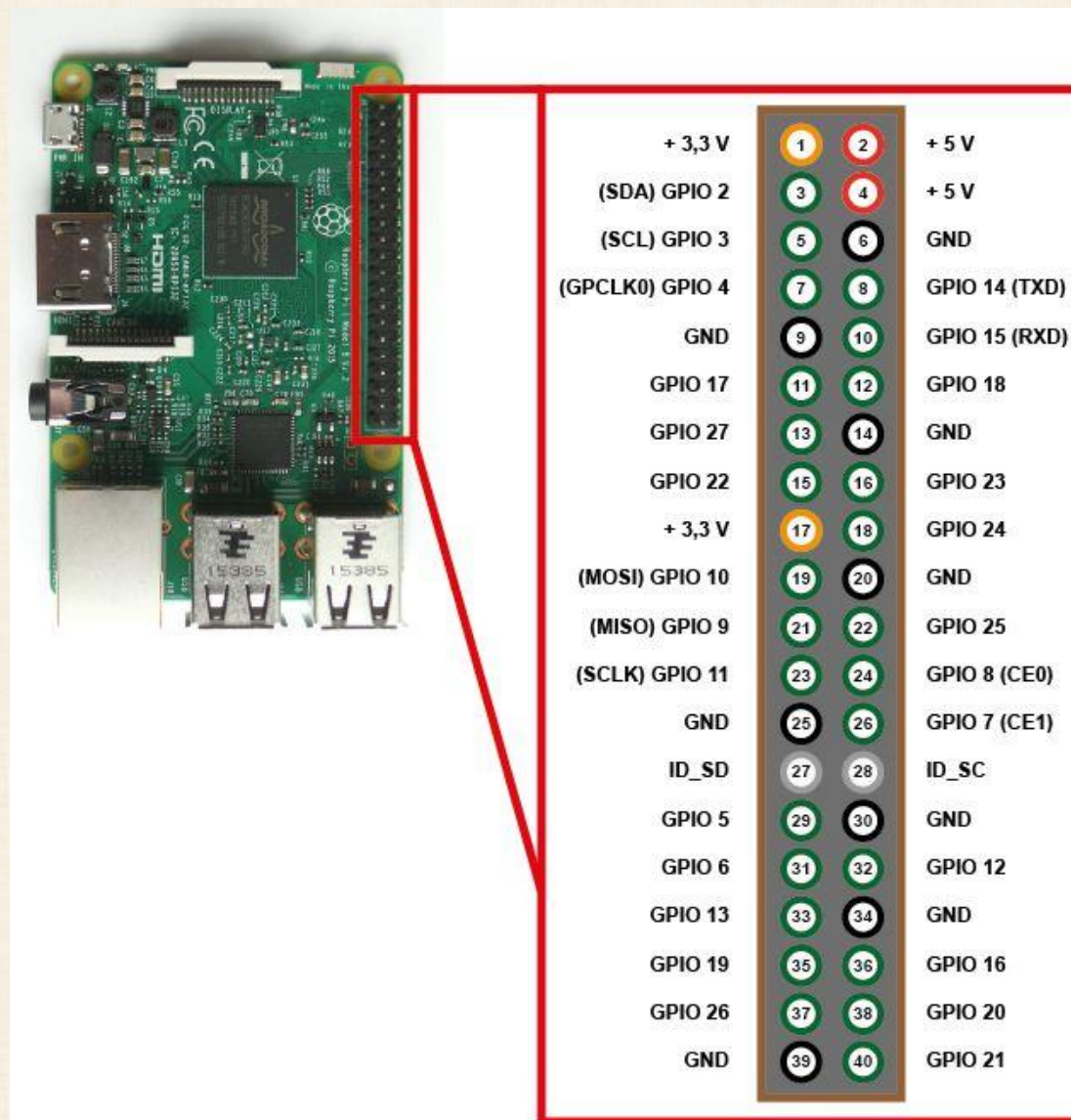
★ □ Q2: 設計一個溫度警示燈

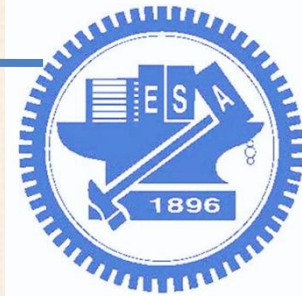
- 持續監測，當溫度大於特定值(ex: 28度), 開啟LED燈號





腳位參考圖





3. HC-SR04實驗

VCC (電源輸入)

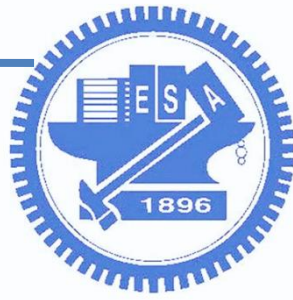
Trig (Trigger)

Echo (Echo)

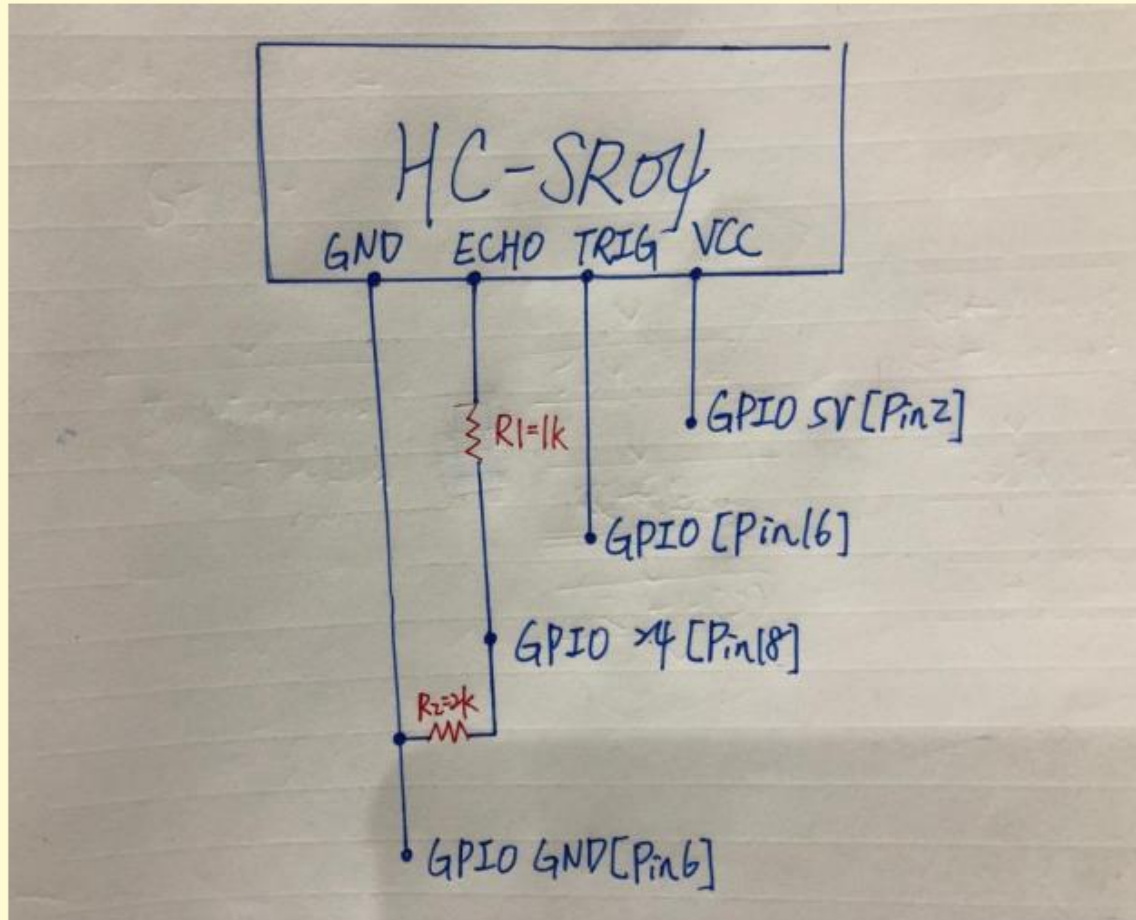
GND (接地)



- Rpi 可透過 Trigger 接腳通知 HC-SR04 進行偵測，而 Echo 接腳用來接受偵測的結果。



電路圖

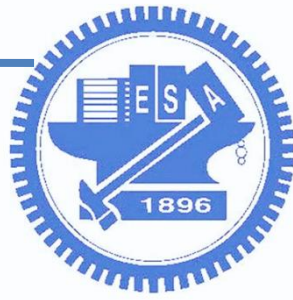


1K: 棕黑紅



2K: 紅黑紅





測量距離

```
import RPi.GPIO as GPIO
import time

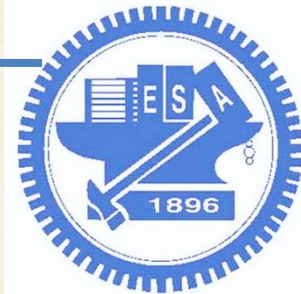
GPIO.setwarnings(False)

v=343
TRIG = 16
E = 18

print '1'
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(E,GPIO.IN)
GPIO.output(TRIG,GPIO.LOW)
def measure():
    GPIO.output(TRIG, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(TRIG, GPIO.LOW)
    pulse_start = 0
    pulse_end =0
    while GPIO.input(E) == GPIO.LOW:
        pulse_start = time.time()
    while GPIO.input(E) == GPIO.HIGH:
        pulse_end = time.time()

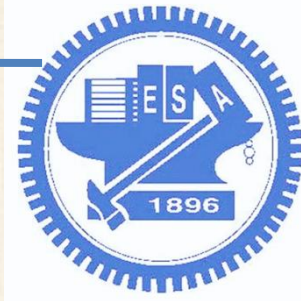
    t = pulse_end-pulse_start
    d=t*v
    d=d/2
    return d*100

while(1):
    print measure()
    time.sleep(1)
GPIO.cleanup()
pi@raspberrypi:~$ █
```

Q3

- 距離感測警示燈
 - 當距離在 $10 \sim 20$ 之間，閃爍LED燈
 - 當距離 <10 ，持續亮燈



Q4 (加分題)

- 將速度調整為 $V=331+0.6 \times T$
溫度 T : 由溫溼度感測器測量

要print出公式、結果、當前溫度

```
import RPi.GPIO as GPIO
import time

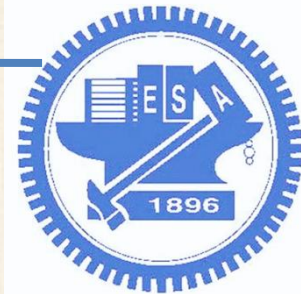
GPIO.setwarnings(False)

v=343
TRIG = 16
E = 18

print '1'
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(E,GPIO.IN)
GPIO.output(TRIG,GPIO.LOW)
def measure():
    GPIO.output(TRIG, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(TRIG, GPIO.LOW)
    pulse_start = 0
    pulse_end = 0
    while GPIO.input(E) == GPIO.LOW:
        pulse_start = time.time()
    while GPIO.input(E) == GPIO.HIGH:
        pulse_end = time.time()

    t = pulse_end-pulse_start
    d=t*v
    d=d/2
    return d*100

while(1):
    print measure()
    time.sleep(1)
GPIO.cleanup()
pi@raspberrypi:~$ █
```



Reference

- From 台灣樹莓派
 - slideshare 投影片
 - <http://www.slideshare.net/raspberrypi-tw/presentations>
 - github 程式碼
 - <https://github.com/raspberrypi-tw>