

通訊網路實驗

IoT應用 Raspberry PI

Dept. of Electrical and Computer Engineering (ECE)

National Yang Ming Chiao Tung University



評分標準&注意事項

- □ 出席 20%
- □ Demo 30% (視情況微調)
- □ 結報 50%
 - □實驗項目
 - □實驗過程 (Code + 說明)
 - □心得

PDF檔案,期限為1週 (下次上課前一天晚上12點)

學號_姓名_Labx.pdf

交錯檔案扣分哦②



課程大綱

□ 1. 認識Raspberry Pi, Python複習

- □ 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NYCU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: 結合溫度偵測和距離偵測



Introduction

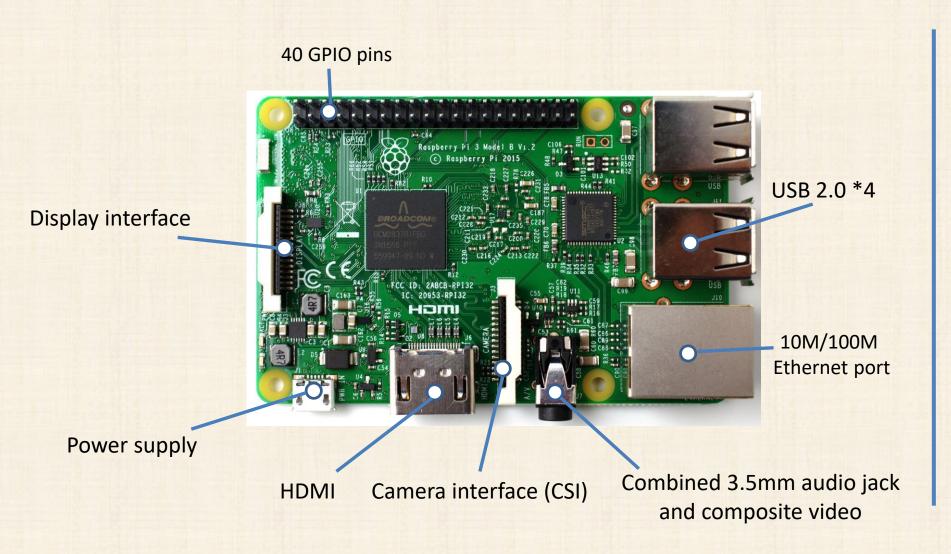
□ 信用卡大小般的電腦



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



Raspberry PI 硬體週邊





Raspberry Pi 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



注意事項

□ 插拔SD卡時,請務必切斷電源

- □關機指令
 - □ sudo shutdown -h now



TTL序列連接

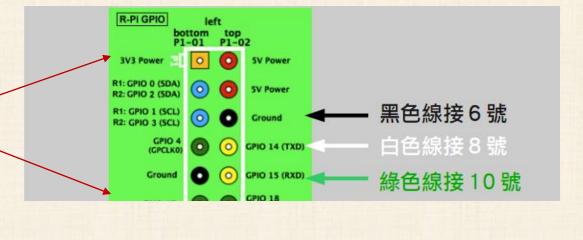
- □ 1.將SD卡插到Raspberry PI並開機
- □ 2.透過 USB 轉 TTL 序列傳輸線,就可以在不需要螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi

□ 預設登入帳密

□ ID: pi

PW: raspberry

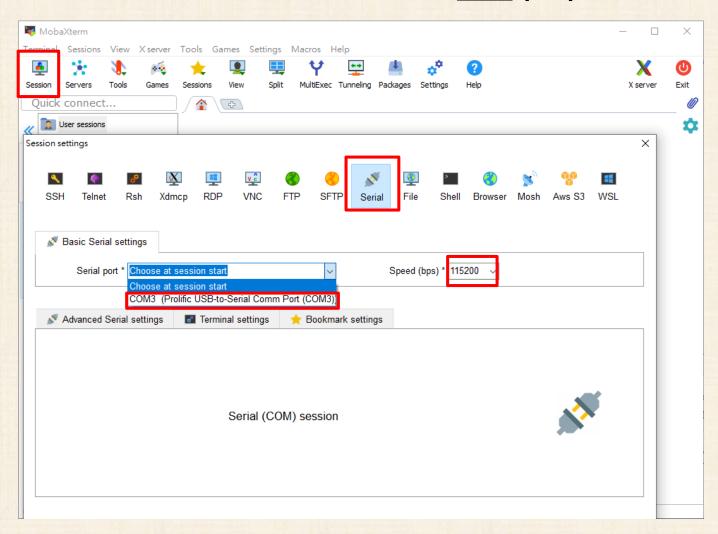
注意接角



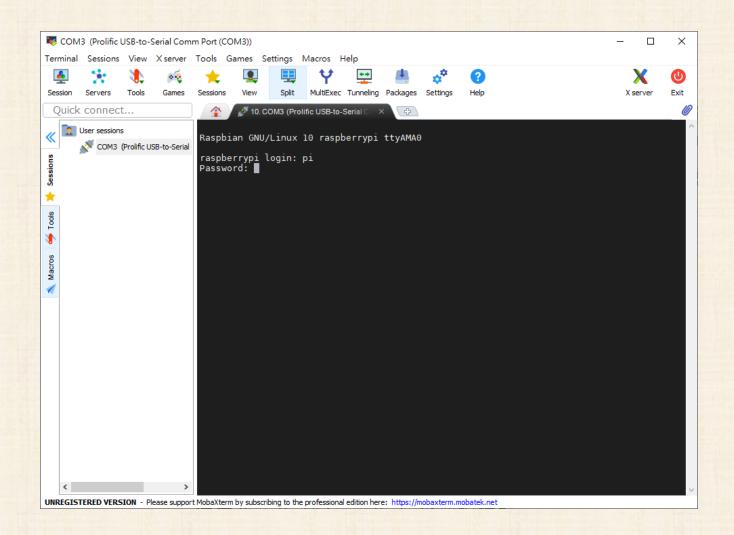
https://www.raspberrypi.com.tw/tag/usb-to-ttl/



MobaXterm畫面



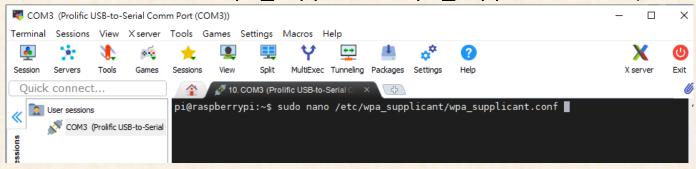






Wi-fi

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) **自行輸入,複製會有誤**

重開機輸入sudo reboot使用 ifconfig 與 iwconfig 檢查連線狀態



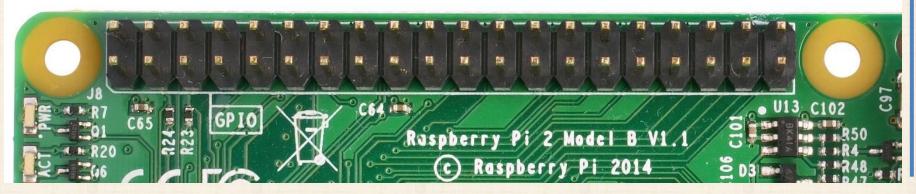
- □ 文字編輯器 nano
 - □ 編輯結束按 ctrl + x 離開
 - 若有變動, 會問你是否存檔, 輸入 Y 即可

```
Modified
 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="Bun"
     psk="7111177117"
```



GPIO on Raspberry PI

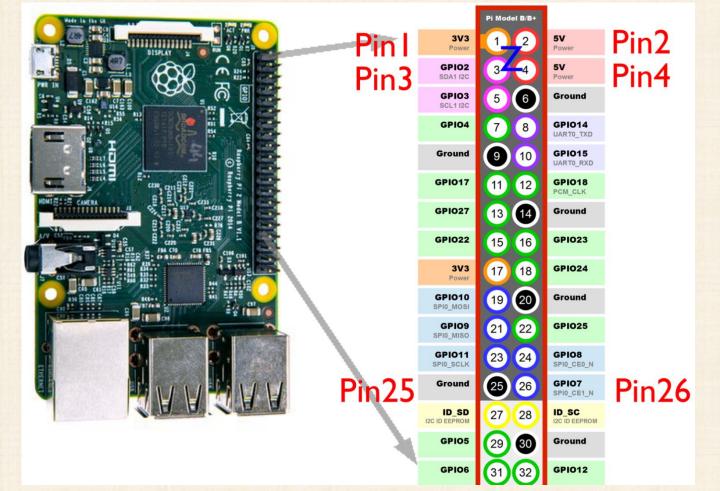
- General-purpose input/output (GPIO)
 - 通用型之輸入輸出(General Purpose I/O)
 - PIN腳可設為輸入(GPI), 輸出(GPO), 輸入與輸出 (GPIO)
 - ■輸出:寫值到某根腳位;輸入:從某根腳位讀值
 - 可用軟體控制的數位訊號





GPIO on Raspberry PI

□z 字型的腳位編號

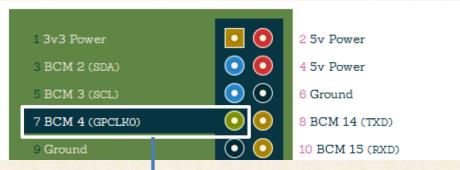




GPIO的腳位與應用

- □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



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



控制 PI 的 GPIO

- □寫程式
 - C
 - C + wiringPi
 - C#
 - Ruby
 - Perl
 - Python



我們用這個

- Scratch
- Java Pi4J Library
- Shell script



Python 簡介

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



(1). 變數,物件,註解

□動態型別 (dynamic typing)

```
# 井號是註解
              # 變數 i 指到數字物件 3
i = 3
              # 印出i
print i
          # 變數 i 指到串列物件
i = [1, 2, 3, 4, 5]
              # 印出串列中第三個元素
print i[2]
i="abcde" # 變數 i 指到字串物件
               # 印出字串中第一個元素
print i[0]
```



(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(,)

```
# 不换行输出
print x,
print y,

# 不换行输出
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

http://www.runoob.com/python/python-basic-syntax.html



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:存檔

- □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



常見指令

- - □ 查看當下目錄中的檔案
- □ cd 資料夾名稱(ex: cd Desktop)
 - □ 移動到Desktop中
- □ cd ..
 - □ 上一頁(資料夾)
- □ cd (空格)
 - □ 回主畫面 Home page (不是桌面)
- □ nano 檔案名稱 (ex: nano led.py)
 - □ 編輯led.py (如果沒有led.py,則會新建一個led.py)



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



1. 控制LED燈

□ Python code (LED閃爍)

```
import RPi.GPIO as GPIO
  import time
  GPIO.setmode(GPIO.BOARD)
  LED PIN = 12
   GPIO.setup(LED PIN, GPIO.OUT)
   while True:
10
       print("LED is on")
       GPIO.output(LED PIN, GPIO.HIGH)
       time.sleep(1)
13
       print("LED is off")
14
       GPIO.output(LED PIN, GPIO.LOW)
       time.sleep(1)
   GPI0.cleanup()
```

載入函式庫

GPIO.BOARD: 按照腳位的順序編號 使用 pin 12的接腳

判斷、控制

縮排要統一



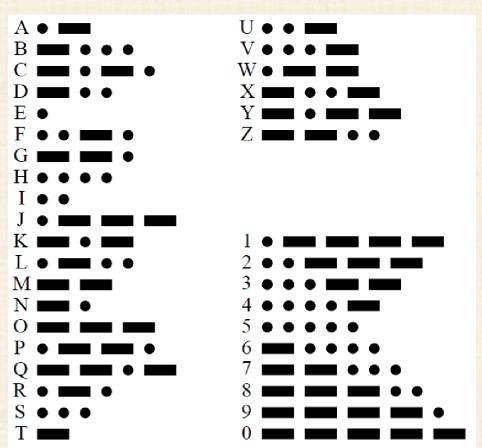
Q1



★□用LED產生NYCU的摩斯密碼

- The length of a dot is one unit.
- A dash is three units.
- The space between parts of the same letter is one unit.
- The space between letters is three units.
- The space between words is seven units.

PS. 可設定 one unit = 1 sec



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



2. 溫濕度sensor

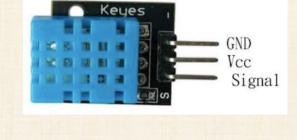
DHT11

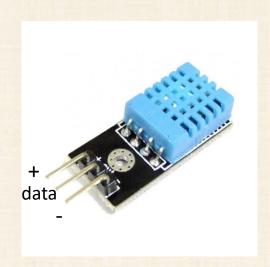
• 温濕度感應器

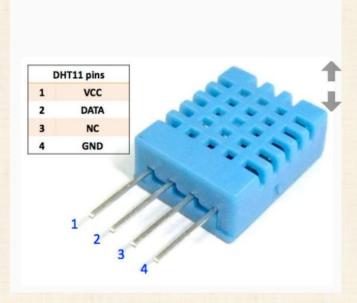
• 温度: 0~50℃, 誤差±2℃

· 濕度: 20~90%, 誤差±5%

• 使用三個腳位: Data, VCC, GND (out, +, -)



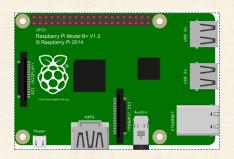






2. 溫濕度sensor

- □ 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(後面要接2個參數)
 - \$ cd examples

Code第一行是環境變數, 需保留: #!/usr/bin/python

- \$ 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%

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
 - 修改AdafruitDHT.py, 改為自己想要的格式



2. 擷取溫濕度資訊

Adafruit_Python_DHT/examples/AdafruitDHT.py:

```
import sys
                        載入函式庫
import Adafruit DHT
sensor args = { '11': Adafruit DHT.DHT11,
                                              定義.py後的參數
         '22': Adafruit DHT.DHT22,
         '2302': Adafruit DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor args:
  sensor = sensor_args[sys.argv[1]]
  pin = sys.argv[2]
else:
  print('usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#')
  print('example: sudo ./Adafruit DHT.py 2302 4 - Read from an AM2302 connected to GPIO #4')
  sys.exit(1)
                                                              導入.py後的2個輸入參數
humidity, temperature = Adafruit DHT.read retry(sensor, pin)
                                                              (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))
                                                                           回傳數值
else:
   print('Failed to get reading. Try again!')
  sys.exit(1)
```

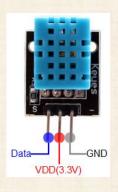


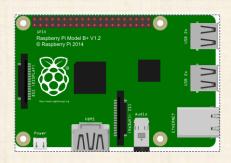
Q2

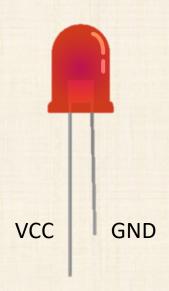


★□設計一個溫度警示燈

· 當溫度大於特定值(ex: 28度), 開啟LED燈號









腳位參考圖



+ 3,3 V + 5 V (SDA) GPIO 2 +5 V 6 5 (SCL) GPIO 3 GND 7 8 (GPCLK0) GPIO 4 GPIO 14 (TXD) 9 10 GND GPIO 15 (RXD) 1 12 **GPIO 17** GPIO 18 13 14 **GPIO 27** GND 15 16 GPIO 22 GPIO 23 17 18 + 3,3 V GPIO 24 19 20 (MOSI) GPIO 10 GND 21) 22 (MISO) GPIO 9 **GPIO 25** 23 24 (SCLK) GPIO 11 GPIO 8 (CE0) 25 26 GND **GPIO 7 (CE1)** 27 ID_SD 28 ID_SC 29 30 GPIO 5 GND 31 32 GPIO 6 GPIO 12 33 34 **GPIO 13** GND 35 36 GPIO 19 GPIO 16 37 38 GPIO 26 GPIO 20

39

40

GPIO 21

GND



3. HC-SR04實驗

VCC (電源輸入)

Trig (Trigger)

Echo (Echo)

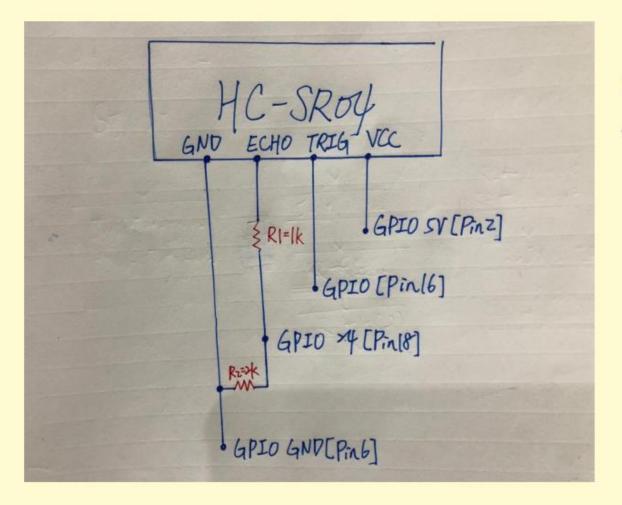
GND (接地)



□ 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

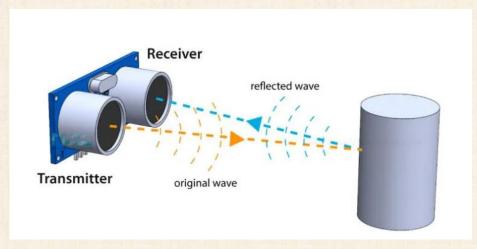




測量距離

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
v = 343
TRIG = 16
E = 18
print 'l'
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@raspberrvpi:~$
```

Using the *Trig* pin we send the ultrasound wave from the transmitter, and with the *Echo* pin we listen for the reflected signal.



https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/



Q3

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



Broadband Ubiquitous Networking Lab

Q4

- □ 將速度調整為V=331+0.6×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 'l'
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