Welcome to "IoT Platform" Course

吳昆儒

Aim of This Course

DCP3516 Raspberry PI

- Course outline
  - 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

Raspberry PI Tutorial
1 - Introduction

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

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

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 硬體週邊

Raspberry PI 硬體週邊

Raspberry PI 硬體週邊

Using Display interface

Raspberry PI 硬體週邊

Combined 3.5mm audio jack
and composite video

Camera interface (CSI)

Raspberry PI 3 Model B

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

Raspberry PI 3 Model B

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

PI2 與 PI3 的主要差異

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

Raspberry Pi 可以用在?

- OSMC

https://osmc.tv/download/

Raspberry Pi 可以用在?

- OpenELEC

http://openelec.tv/get-openelec

Raspberry Pi 可以用在?

- RetroPie

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

Raspberry Pi 可以用在?

- PiPlay

http://piplay.org/

Raspberry Pi 可以用在?

- Kali Linux

https://www.kali.org/

Raspberry Pi 可以用在?

- Android

https://www.raspberrypi.org/magpi/wp-content/uploads/2016/09/Android_Raspberry_pi.jpg

Raspberry Pi 可以用在?

- 超級電腦

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

Raspberry Pi 可以用在?

- 空間邏輯測驗(?)

Raspberry Pi 可以用在?

- 自動轉珠系統 (全自動パズドラロボ作ってみた (Niconico))
  - 影像辨識 + 自動化控制

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

Raspberry PI Tutorial
2 - Basic of RaspPI

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Basic of PI

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

準備工具

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

1. 安裝OS (Raspbian)

- 步驟1：下載映像檔
- 步驟2：將映像檔燒錄至SD卡
- 步驟3：調整開機設定檔
- 步驟4：電腦端準備 USB 轉 TTL 序列傳輸線
- 步驟5：將SD卡插到Raspberry PI並開機

步驟1：下載映像檔

- 官方下載網頁
  - http://www.raspberrypi.org/downloads
- 選擇映像檔(image)
  - NOOBS
  - **Raspbian(推薦)**
  - Pidora
  - RISC OS
  - RaspBMC
  - Arch
  - OpenELEC

步驟2：將映像檔燒錄至SD卡

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

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

選擇image位置

選擇SD卡位置

Read: 將SD卡儲存至image路徑, 可備份整個系統

Write: 將image寫入至SD卡

步驟3：調整開機設定檔

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

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

- Sol: 在SD卡根目錄的 **config.txt** 底下加上這兩行

force_turbo=1 1. 強制 CPU 的時脈維持 1.2GHz 的時脈

dtoverlay=pi3-disable-bt 2. 停用藍牙

https://www.raspberrypi.com.tw/10842/raspberry-pi-3-uart-overlay-workaround/

步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

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

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

步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

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

步驟5：將SD卡插到Raspberry PI並開機

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

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

步驟5：將SD卡插到Raspberry PI並開機

還是沒有畫面? -> 電腦重開機試試

2. PI的環境設定

- 擴充SD卡空間
  - 預設只有使用約2G的空間 -> 可是SD卡有32G
  - 在terminal下輸入: **sudo raspi-config ->** 選擇「 **Expand Filesystem** 」
  - Reboot後系統所能使用的空間就為SD卡所擁有的空間

2. PI的環境設定

- 設定Wi-Fi
  - 使用螢幕鍵盤滑鼠, 可以直接點選Wi-Fi
    -> 那TTL的方式要怎麼設定Wi-Fi?
- Sol
  1. 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
     - sudo reboot
  4. 使用 ifconfig 與 iwconfig 檢查連線狀態

2. PI的環境設定

- 設定Wi-Fi
  - 編輯設定檔之示意圖

COM3 - PuTTY

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

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

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

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

2. PI的環境設定

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

2. PI的環境設定

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

- Ex: sudo apt-get install apache2
  - 安裝網頁伺服器, Apache HTTP Server

使用apt-get安裝程式

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

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

- 彩蛋: sudo apt-get moo

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

3. 設定遠端桌面連線

- 習慣桌面環境? -> 可使用遠端軟體
- 如何得知PI的IP? -> 使用 **ifconfig** 指令查詢IP位址
- 安裝遠端軟體: vnc 或 xrdp
  - sudo apt-get install tightvncserver
  - sudo apt-get install xrdp

- vnc需要另外設定
  - 終端機執行 vncserver, 接著會詢問幾個問題, 以及登入的密碼

```
pi@raspberrypi ~ $ vncserver

You will require a password to access your desktops.

Password:
Warning: password truncated to the length of 8.
Verify:
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is raspberrypi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log
```

3. 遠端桌面控制

- 使用vnc:
  - Client端需要下載vncviewer用戶端的程式, 才可建立連線
    - PC: https://www.realvnc.com/download/viewer/
- 使用xrdp:
  - Windows系統可用內建的遠端桌面連線

3. 遠端桌面控制 (xrdp)

- 遠端畫面 (windows遠端桌面連線 + xrdp)

Raspberry PI Tutorial
3 - GPIO

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Outline

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

0. GPIO on Raspberry PI

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

0. GPIO on Raspberry PI

- Z 字型的腳位編號

0. GPIO on Raspberry PI

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

0. GPIO的腳位與應用

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

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

0. GPIO的注意事項

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

0. 控制 PI 的 GPIO

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

0. Python 簡介

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

- PS. PI預設使用Python 2

(1). 變數，物件，註解

- 動態型別 (dynamic typing)

```
# 井號是註解

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

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

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

(2). 模組

- 模組

```
# import MODULE
import RPi.GPIO

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

可以使用該模組的函式

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

(3). 縮排

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

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

前面有 4 個空白鍵

(4). 迴圈

- 自動迭代 (iterator); for loop

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


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

前面有 4 個空白鍵

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

(5). 條件判斷

- If statement

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

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

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

(6). 函式

- 自訂函式 (User-Defined Functions)

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

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

f(4) 可以得到 17

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

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. LED燈

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

https://zh.wikipedia.org/wiki/發光二極管

1. LED燈

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

Electrical / Optical Characteristics at TA=25°C

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

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

LED的順向電壓是1.85V, 電流需要20mA

1. LED燈

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

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

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

1. LED燈

- 電阻的判斷方式?

四環：右側為金或銀

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

五環：前四環很靠近

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

1. LED燈

- 麵包板的使用

1. 藍色和綠色兩塊不通
2. 藍色水平相通
3. 紅色垂直相通

麵包板內部構造

1. LED燈

- 連接LED燈
  - LED的長腳位為VCC, 短腳位為GND
  - VCC 接至 PI 的 Pin 12
  - GND 接至 PI 的 Pin 6
  - 使用1k Ω電阻
    - Q0: 1k 電阻長什麼樣子?

1. 控制LED燈

- Python code (LED閃爍)

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

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

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

GPIO.cleanup()
```

載入函式庫

GPIO.BOARD:
按照腳位的順序編號

使用NO. 12的pin腳

載入函式庫

1. 控制LED燈

★ □ 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

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

2. 溫濕度sensor

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

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

2. 溫濕度sensor

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

2. 溫濕度sensor

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

2. 擷取溫濕度資訊

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

Code第一行是環境變數, 需保留:
#!/usr/bin/python

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

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

2. 擷取溫濕度資訊

- Python code:

```
import sys
import Adafruit_DHT

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

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

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

載入函式庫

讀取.py後的參數

本程式的執行方式 (後面要接2個參數)

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

回傳數值

2. 擷取溫濕度資訊

- Q2: 設計一個溫度警示燈
  - 溫度大於特定值(ex: 25度), 開啟LED燈號

3. HC-SR04 超音波感測器

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

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

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

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

$$17150 \times Time = Distance$$

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

3. HC-SR04 超音波感測器

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

可得到時間差

VCC, Trigger, Echo, GND

3. HC-SR04 超音波感測器

- 規格與時序圖

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

4、Ultrasound timing diagram

建議測量間隔 >60ms

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

3. HC-SR04 超音波感測器

- 時序圖(Timing diagram)

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

3. HC-SR04 超音波感測器

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

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

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

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

$$660 + 0.66R2 = R2$$

$$660 = 0.34R2$$

$$1941 = R2$$

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

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

計算結果: R1=1K, R2 取 2K

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

3. HC-SR04 超音波感測器

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

3.超音波(Python)

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

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

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

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

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

載入函式庫

第一行是環境變數
很重要, 不可以刪掉!!

在Trigger上傳送10us的訊號

在Echo上偵測Pulse寬度

3. 測量距離

- Q3: DIY倒車雷達
  - 結合LED+溫濕度+超音波
  - 距離需搭配溫度校正
  - 距離30cm以上, LED沒反應
  - 小於30cm, LED開始閃爍
  - 小於15cm, LED的閃爍變快

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

Raspberry PI Tutorial
3 - GPIO

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

龍珠雷達 XD

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

Demo video

- Open Source IMU and AHRS Algorithm with x-IMU
  - https://www.youtube.com/watch?v=BXsGWoOMtmU
- 3D Box on screen rotated around its roll and pitch axes through an acceleration sensor
  - https://www.youtube.com/watch?v=85rl3KVAEGU
- 9DOF IMU Test
  - https://www.youtube.com/watch?v=C7DVH3goMUA

Outline

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

這個念square喔!!

I2C

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

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

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

I2C

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

https://zh.wikipedia.org/wiki/上拉电阻

啟用 I2C功能

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

啟用 I2C

- 接著選擇A7 I2C

啟用 I2C

- 是否啟用I2C介面？

啟用 I2C

- 啟用完畢

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

Sensor連接線路圖

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

請記得加兩個2.2k的電阻在SDA/SDL上

測試 I2C

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

有連接i2c裝置

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

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

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

沒有連接i2c裝置

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

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

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

GY-801九軸感測器

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

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

GY-801九軸感測器

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

GY-801九軸感測器

磁力計(HMC5883L)

陀螺儀(L3G4200D)

加速度計(ADXL345)

氣壓計(BMP085)

常見名詞解釋

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

1. 加速度計

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

1. 加速度計

- 一個立方體盒子放著一顆球
- 盒子運動時，可根據球的位置算出加速度的值

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

1. 加速度計

- 水平向左施力
  - 球會因為慣性貼向盒內右側

1. 加速度計

電容微加速度器的構造與原理

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

1. 加速度計(ADXL345)

另一種常見的加速度計叫 MPU6050

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

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

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

1. 加速度計(ADXL345)

1. 加速度計

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

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

1. 加速度計(ADXL345)

座標系定義

1. 加速度計(ADXL345)

- 下載安裝程式

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

cd examples/
sudo python simplectest.py
```

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

1. 加速度計(ADXL345)

- 讀取加速度資訊

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

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

1. 加速度計(ADXL345)

- Adafruit_ADXL345/ADXL345.py

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

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

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

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

1. 加速度計(ADXL345)

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

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

- 使用下面公式轉換回+/-2G的範圍：

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

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

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

1. 加速度計(ADXL345)

- Q1-1: 將程式輸出結果改為+/-2G的範圍
  - sudo python simpletest.py > log.txt
  - 上面指令可以把程式執行結果存到log.txt裡面
    終端機不會有訊息, 需按ctrl+c終止
- Q1-2: 紀錄加速度的數值,畫出靜止狀態與搖晃狀態的圖
  - 指令: cat log.txt　　可以在終端機顯示檔案內容, 再用滑鼠複製貼上

2. 陀螺儀

轉動會影響電極距離
(科式力)

微陀螺儀的構造與原理

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

2. 陀螺儀 (L3G4200D)

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

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

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

2. 陀螺儀 (L3G4200D)

- 下載安裝程式

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

- 執行結果

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

2. 陀螺儀 (L3G4200D)

```
#!/usr/bin/python

from time import sleep
import smbus
import string

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

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

載入time的sleep (載入特定成員)

轉換回正負數值

使用二的補數來表示正負號

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

設定暫存器位址

2. 陀螺儀 (L3G4200D)

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

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

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

讀第一個&第二個byte, 結合在一起

用二的補數轉換回來

顯示數字後, 往右補10個空白
(排版用)

2. 陀螺儀 (L3G4200D)

- 執行結果長這樣, 要如何轉換成角度的格式?

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

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

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

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

應用

- 姿態分析

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

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

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

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

2. 陀螺儀 (L3G4200D)

- Q2-1: 紀錄陀螺儀的數值, 畫出靜止狀態與搖晃狀態的圖
  - sudo python gyro.py > log.txt
  - 上面指令可以把程式執行結果存到log.txt裡面
    終端機不會有訊息, 需按ctrl+c終止
- Q2-2: 計算Pitch, Roll 的數值

3. 磁力計

霍爾電壓
(Hall voltage)

微地磁計的構造與原理

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

3. 磁力計 (HMC5883L)

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

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

3. 磁力計 (HMC5883L)

- 根據datasheet設定暫存器

Register Access

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

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

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

3. 磁力計 (HMC5883L)

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

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

Table2: Register List

3. 磁力計 (HMC5883L)

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

- Configuration Register B

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

Table 7: Configuration B Register

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

+/- 1.3Ga; Gain 1090

3. 磁力計 (HMC5883L)

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

- Configuration Register A

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

Table 3: Configuration Register A

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

8 samples

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

Table 5: Data Output Rates

15 Hz

3. 磁力計 (HMC5883L)

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

- Mode Register

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

Table 10: Mode Register

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

設定為Continuous-Measurement Mode

3. 磁力計 (HMC5883L)

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

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

Fig. Magnetic Field Inclination Angle

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

3. 磁力計 (HMC5883L)

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

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

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

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

def main():

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

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

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

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

讀取x/y/z的資料

顯示讀取的x/y/z的資料

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

3. 磁力計 (HMC5883L)

- Q3-1: 紀錄磁力計數值, 將磁力計水平旋轉一圈紀錄xy變化
  - python compass.py > log.csv
  - 上面指令可以把程式執行結果存到log.csv裡面
    終端機不會有訊息, 需按ctrl+c終止
- Q3-2: 嘗試算出點的中心，並利用該中心點對x,y值取反三角函數得到角度值
  - Hint: math.atan2()

3.磁力計單位

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

Performance

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

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

Reference

- Datasheet
  - 加速度
    - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf
  - 陀螺儀
    - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf
  - 電子羅盤
    - http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
  - 氣壓計
    - http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- 慣性感測器簡介 (Chih-Wei Yi)
  - https://goo.gl/BX2cor

Raspberry PI Tutorial
4 - Python on RaspPI

Dept. of Computer Science and
Information Engineering
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
- 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

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

datasheet

```
# 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)
```

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)

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

- 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, 可利用下面公式計算高度

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

氣壓計 BMP085

- **Q1: 測量目前的氣壓計資訊, 目前的高度是多少? 覺得合理嗎**

- **Q2: 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吧
  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)
- **Q3-1: 試著用low-pass filter分析sensor數值** (用excel即可)

(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

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
- **Q3-2: 觀察kalman filter與complementary filter之輸出資料, 並畫成圖表**

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
  - 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
- Paper
  - http://www.sciencedirect.com/science/article/pii/S0921889015000901

Raspberry PI Tutorial 5 - Output of embedded system

Dept. of Computer Science and Information Engineering
National Chiao Tung University

Outline

- **Cross Compilation**
- Camera Setup for Raspberry Pi
- V4L2
- Framebuffer

Cross Compilation

- It is not infeasible for a compiler to run on a resource limited embedded system
- To cross-compile is to build a binary on one platform (like PCs) that will run on another platform (like embedded systems)

Cross Compiling Environment Setup

- Clone the toolchain
  - $ git clone https://github.com/raspberrypi/tools
- Add the toolchain path to PATH
  - $ export $PATH:***YOUR_CLONE_LOCATION***/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin
- Test if you can run the cross compiler
  - $ arm-linux-gnueabihf-gcc -v

Outline

- Cross Compilation
- **Camera Setup for Raspberry Pi**
- V4L2
- Framebuffer

Find The Camera Port on Raspberry Pi

Attach The Camera Ribbon Cable to The Camera Port

Enable Connection to The Camera

- $ sudo raspi-config
  - Select **“Interfacing Options”**
  - Select **“Camera”**
  - Select **“Yes”**

Test The Camera

- There are built-in Raspberry Pi camera applications written based on MMAL API (Broadcom-specific)
  - raspistill is for capturing images
  - raspivid is for capturing video
- Run a demo mode, the camera output should be shown on the screen
  - $ raspistill -d ***or*** raspivid –d

Outline

- Cross Compilation
- Camera Setup for Raspberry Pi
- **V4L2**
- Framebuffer

Capture Camera Output via V4L2(Video for Linux 2) API

- Video4Linux is a collection of device drivers and an API for supporting realtime video capture on Linux systems
- Add V4L2 module to Raspbian
  - $ sudo modprobe bcm2835-v4l2
- Use the ***ioctl*** system call to access a V4L2 device
  - ioctl(***fd***, ***VIDIOC_XXXX***, ***arg***)

V4L2 Programming Flow

YUYV to RGB Conversion

- One YUYV pixel is converted to two RGB pixels
- RGB1
  - r1 = y1 + 1.042 * (v - 128)
  - g1 = y1 - 0.344 * (u - 128) - 0.714 * (v - 128)
  - b1 = y1 + 1.772 * (u - 128)
- RGB2
  - r2 = y2+ 1.042 * (v - 128)
  - g2 = y2 - 0.344 * (u - 128) - 0.714 * (v - 128)
  - b2 = y2 + 1.772 * (u - 128)

Lab5.1

- Finish v4l2_cap_img.c
  - Fill in fields marked with ***lab5.1_v4l2*** and ***lab5.1_YUYV2RGB***
  - Use ***arm-linux-gnueabihf-gcc*** to complie it
    - $ arm-linux-gnueabihf-gcc -I qdbmp_1.0.0 v4l2_cap_img.c qdbmp_1.0.0/qdbmp.c -o v4l2_cap_img
- Copy the executable file to raspberry pi and run it
  - You should capture an image from the camera

Outline

- Cross Compilation
- Camera Setup for Raspberry Pi
- V4L2
- **Framebuffer**

Framebuffer

- Framebuffer is a part of video memory containing a current video frame

Graphical representation of 24-bit "true color"

Frambuffer Access in Linux

- Linux framebuffer (fbdev) is a graphic hardware-independent abstraction layer to show graphics on a computer monitor
- Use the ***ioctl*** system call to access a frambuffer device
  - ioctl(***fd***, ***FBIOXXXX***, ***arg***)
- There are two structures store information about the screen
  - fb_fix_screeninfo
    - Device independent unchangeable information
  - fb_var_screeninfo
    - Device independent changeable information

fb_var_screeninfo

Lab5.2

- Finish fb_show_img.c
  - Fill in fields marked with ***lab5.2_fb_var_screeninfo***
  - Use ***arm-linux-gnueabihf-gcc*** to complie it
    - $ arm-linux-gnueabihf-gcc -I qdbmp_1.0.0 fb_show_img.c qdbmp_1.0.0/qdbmp.c -o fb_show_img
- Copy the executable file to raspberry pi and run it
  - You should see the captured image shown on the screen

Lab5.3

- By combining Lab5.1 and Lab5.2, stream the camera output on the screen
- Bonus
  - Locate the camera output at the screen center
  - Reduce the latency of the streaming
    - Hint: lower resolution, fewer buffers

V4L2 Programming Flow

Lab5 Assignment

- Lab5 setup(15%)
- Lab5.1(20%)
  - Use the camera to capture an image
- Lab5.2(20%)
  - Show the captured image on the screen
- Lab5.3(30%)
  - Stream the camera output on the screen
- Lab5.3 Bonus(10%)
- Report(15%)

Reference

- V4L2 API
  - https://www.linuxtv.org/downloads/v4l-dvb-apis-new/uapi/v4l/v4l2.html
- Frame buffer programming
  - http://betteros.org/tut/graphics1.php
  - http://raspberrycompote.blogspot.tw/2012/12/low-level-graphics-on-raspberry-pi-part_9509.html

VM Setup

Install VirtualBox and Download The Virtual Hard Disk File TA Provided

- VirtualBox
  - https://www.virtualbox.org/
- Virtual Hard Disk File
  - https://drive.google.com/open?id=0By9sLmyyi0OXeTFKVS00ZjVxZjg

Click **“New”**

Oracle VM VirtualBox Manager
New
Settings
Discard
Start
Details
Snapshots
windows10
Running
OSDI
Powered Off
ubuntu
Powered Off
General
Name: ubuntu
Operating System: Ubuntu (64-bit)
System
Base Memory: 4096 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization
Preview
ubuntu
Display
Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled
Storage
Controller: IDE
IDE Secondary Master: [Optical Drive] VBoxGuestAdditions.iso (56.28 MB)
Controller: SATA
SATA Port 0: ubuntu.vmdk (Normal, 32.00 GB)

Select The OS of The New VM and Enter A Name

Allocate Memory to The New VM

Don't allocate too much memory to the new VM. Reserve enough memory for the host machine, or it will degrade performance of the host machine.

Select The Virtual Hard Disk File TA Provided

Select The New VM and Click **“Start”**

The Password Is **esdi2017**

Check If You Can Use The Cross Compiler

2017_esdi [Running] - Oracle VM VirtualBox

Terminal

esdi2017@esdi2017: ~

```
esdi2017@esdi2017:~$ ls
a.out            Documents         hello_world.c  Pictures   tools
cscope_maps.vim  Downloads         hell_world     Public     Videos
Desktop          examples.desktop  Music          Templates
esdi2017@esdi2017:~$ tail -n 1 .bashrc
export PATH=$PATH:tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/
bin/
esdi2017@esdi2017:~$ arm-linux-gnueabihf-gcc hello_world.c -o hell_world
esdi2017@esdi2017:~$ ./hell_world
bash: ./hell_world: cannot execute binary file: Exec format error
esdi2017@esdi2017:~$ file hell_world
hell_world: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (SYSV), dynamically
 linked (uses shared libs), for GNU/Linux 2.6.26, BuildID[sha1]=028c747719d0ad63
8e7cc1a7ec25b39ac50315bb, not stripped
esdi2017@esdi2017:~$
```

Select The New VM and Click **“Settings”**

Add A Shared Folder

Check **“Auto-mount”**

Insert Guest Additions CD Image

Add Your Username to The **vboxsf** Group

```
esdi2017@esdi2017: ~
esdi2017@esdi2017:~$ sudo adduser esdi2017 vboxsf
The user `esdi2017' is already a member of `vboxsf'.
esdi2017@esdi2017:~$
```

Check If You Can Access The Shared Folder

Raspberry PI Tutorial
6 - Input of embedded system

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Python GUI in Linux frame buffer

- A module “Pygame” is used to initialized in frame buffer mode.
- When application run within X.org context, DISPLAY environment variable is passed to your application.

```
import os
disp_no = os.getenv('DISPLAY')
if disp_no:
    print "I'm running under X display = {0}".format(disp_no)
```

Python GUI in Linux frame buffer

- There are different video drivers available: X11,dga, fbcon, directfb, ggi, vgl, svgalib, aalib.
- We just use directfb, fbcon, svgalib here.
- Check SDL_VIDEODRIVER to check for X display first.

```
drivers = ['directfb', 'fbcon', 'svgalib']

found = False
for driver in drivers:
    if not os.getenv('SDL_VIDEODRIVER'):
        os.putenv('SDL_VIDEODRIVER', driver)
    try:
        pygame.display.init()
    except pygame.error:
        print 'Driver: {0} failed.'.format(driver)
        continue
    found = True
    break

if not found:
   raise Exception('No suitable video driver found!')
```

Python GUI in Linux frame buffer

- Pygame module to control display: https://www.pygame.org/docs/ref/display.html#pygame.display.Info

```
size = (pygame.display.Info().current_w, pygame.display.Info().current_h)
screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
```

http://www.karoltomala.com/blog/?p=679

Pygame initialization

- self.screen.fill((0,0,0)) to clear screen
- pygame.font.init() to initial font support
- pygame.display.update() to render the screen

Pygame display fill screen

- screen.fill((r,g,b)) to fill screen with some color
- Pygame.display.updadte() to update the display
- Example:
  - If I want to fill screen with black, black's color code is #000000.
  - It means red is 0, green is 0, white is 0.
  - Then screen.fill((0,0,0)) will fill screen with black.

Lab6-1

Audio Configuration – Command Line

- **amixer cset numid=3 2**
  - 0 – automatic
  - 1 – headphone jack
  - 2 – HDMI

Audio Configuration – Raspi-Config

- sudo raspi-config
- Advanced Options -> Audio -> HDMI

Raspberry PI Tutorial
7 - RaspPI Networking

Dept. of Computer Science and
Information Engineering
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

Web Server軟體

- Web server Software (http服務):
  - Apache
  - Nginx (發音同engine x)
  - Microsoft - Internet Information Server (IIS)
  - AWS - Amazon Web Services (AWS)

建立 Web Server 網站

- 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&測試:
  - $ ifconfig

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:01:1b:99
          inet6 addr: fe80::8ce:da36:d61c:fce/64 Scope:
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 fr
          TX packets:0 errors:0 dropped:0 overruns:0 ca
          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::b113:3451:56a1: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, Hans.com </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

HTTP

- 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

- 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

- php範例, 利用HTTP/POST傳送資料並存檔

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&Temp28.9" 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

Web Server & HTTP/POST

- Lab1: 同時有兩個sensor會傳送2筆資料, 修改php使server將資料存成不同檔案
  - Ex: sensor 1 & 2同時傳送溫濕度資訊

```
<?php
        header("Content-Type:text/html; charset=utf-8");
        /* Temp -> server */
        // ex: curl -d "sensor=1&Temp28.9" 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

- Lab2: 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

3. 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

PI3內建Wi-Fi與Bluetooth

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

原本的

```
force_turbo=1
dtoverlay=pi3-disable-bt
```

修正後

```
force_turbo=1
dtoverlay=pi3-miniuart-bt
enable_uart=1
```

使用 DTO 重新映射腳位

dtoverlay=pi3-miniuart-bt

解決方法

- 還是找不到藍牙裝置? -> 用USB的吧 XD
  - 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)
```

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
```

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

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 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

sudo hciconfig hci0 leadv 3
sudo hciconfig hci0 noscan

啟用藍牙的低耗能廣告(LE advertising)模式，並關閉掃描功能

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

- Lab 3:
  - 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
???	???
???	???
???	???
???	???
???	???
???	???
???	???

Eddystone訊息

- Lab 4: 使用 hcitool 傳送與相同的Lab 3相同的資訊, 並說明數值意義

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
- 14: the ENTIRE following data packet in bytes (16進位的14 = 20 byte)
- 02 01 1a: Eddystone Adv Flags
- 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
- 00 00 00 00 00 00 00 00 00 00 00: 共 10 byte

Bonus

- 類似Lab2, 使用兩個PI, 其中 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

Lab7 - Assignment

- Lab7-1: (20%)
  - modify PHP to support two sensors and two sensing reading.
- Lab7-2: (15%)
  - use two PI (client and server). Client sends POST to server.
- Lab7-3: (25%)
  - Generate Eddystone advertisement. (show the printscreen)
  - convert the URL to Eddystone format.
- Lab7-4: (25%)
  - Use hcitool to send the same Eddystone advertisement.
  - Explain the meaning of values
- Lab7-2.bonus: (10%)
  - Send DHT sensor data to server
- Report (15%)

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/

Raspberry PI Tutorial
8 - Linking with Arduino

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Outline

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

Arduino is Micro Controller

Digital In/Out Pins

USB Port

Atmega328p

Power Supply

Power Pins

Analog Input Pins

Arduino YUN

https://www.arduino.cc/en/Main/ArduinoBoardYun

Arduino YUN Spec

AVR Arduino microcontroller

Microcontroller	ATmega32U4
Operating Voltage	5V
Input Voltage	5
Digital I/O Pins	20
PWM Channels	7
Analog Input Pins	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

https://www.arduino.cc/en/Main/ArduinoBoardYun

Arduino YUN Spec

Linux Microprocessor (OpenWrt-Yun)

Processor	Atheros AR9331
Architecture	MIPS @400MHz
Operating Voltage	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Card Reader	Micro-SD only
RAM	64 MB DDR2
Flash Memory	16 MB
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz
PoE compatible 802.3af card support	See Power

https://www.arduino.cc/en/Main/ArduinoBoardYun

Arduino online Community

Main Site, Blog, Wiki, Forum and Store

https://www.arduino.cc/

Download the Arduino Software

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)

Examples

sketch_may09a | Arduino 1.0
File
Edit
Sketch
Tools
Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
1.Basics
2.Digital
3.Analog
4.Communication
5.Control
6.Sensors
7.Display
8.Strings
ArduinoISP
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
AnalogReadSerial
BareMinimum
Blink
DigitalReadSerial
Fade
1
Arduino Uno on COM5

Library

sketch_may09a | Arduino 1.0
File
Edit
Sketch
Tools
Help
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
sketch_
Arduino Uno on COM5

Board, Port

sketch_sep18a | Arduino 1.5.7
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Port
Programmer
Burn Bootloader
Arduino AVR Boards
Arduino Yún
Arduino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino ARM (32-bits) Boards
Arduino Due (Programming Port)
Arduino Due (Native USB Port)

sketch_sep18a | Arduino 1.5.7
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Port
Programmer
Burn Bootloader
COM1
COM7 (Arduino Yún)
Arduino Yún on COM7

Verify, Upload, Monitor

Verify

Checks your code for errors.

Upload

Compiles your code and uploads it to the Arduino I/O board.

Serial Monitor

Opens the serial monitor.

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);

Lab 8-1

- 用Arduino接收Lab 2介紹過的溫溼度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) 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
}
```

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)

C. I^2C

D. TTL

E. Software serial

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

Bonus題目

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

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

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
}
```

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);
}
```

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

Connect and Configure Yun's WiFi

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

Connect and Configure Yun's WiFi

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

Representational State Transfer(REST)

- REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements.

- Base on URL, such as http://example.com/resources/
- An internet media type for the data (JSON, XML, etc.)
- Standard HTTP methods (GET, PUT, POST, DELETE)

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);
}
```

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);
}
```

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

E. over software serial

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 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 1/3)
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 1/3)
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()
```

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

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
```

2(D) over software serial

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());
  }
}
```

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()
```

2(D) over software serial

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)

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)

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.

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

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);
}
```

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();
    }
  }
}
```

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);
}
```

Lab 8-2

- 在Raspberry Pi上面show出lab 8-1的sensor資訊
- 格式: "Temp: ___, Humi: ____, Mois: ____"

Lab 8-3

- 改用I²C的方式將moisture傳到Raspberry Pi上
- 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

Bonus

- Arduino用HTTP POST的方式，將sensor data傳到lab 7的Web server

- Arduino用HTTP GET的方式取得Raspberry Pi傳過來的資料顯示在serial monitor上

Raspberry PI Tutorial
9 - Embedded opencv

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Pi Camera

- The python-picamera library is available in the Raspbian archives
  - sudo apt-get install python-picamera

Lab9-1 Camera

- Use camera to get the image
- Draw the circle and rectangle
- Save two images
  - image.jpg = Original image
  - image2.jpg = Image with circle and rectangle

Camera

Import the necessary packages

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
```

Initialize the camera and grab a reference to the raw camera capture

```
camera = PiCamera()
rawCapture = PiRGBArray(camera)
```

Allow the camera to warm up

```
time.sleep(2)
```

Grab an image from the camera

```
camera.capture(rawCapture, format="bgr")
image = rawCapture.array

cv2.imshow("Image", image)
cv2.waitKey(0)
```

Circle and Rectangle

- Circle

  color width

  - cv2.circle(image, (a,b), r, (255,255,155),5)
- Rectangle

  color width

  - cv2.rectangle(image, (a, b), (c, d), (255,255,255), 5)

Lab9-2 Feature Detection

- Use the image.jpg in lab9-1 to do the feature detection

Cascade Classification

- A classifier is trained with a few hundred sample views of a particular object
- Cascade
  - means that the resultant classifier consists of several simpler classifiers (*stages*) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed

Face Detection

```
import cv2
import numpy

img = cv2.imread('image.jpg')
```

Load a cascade file for detecting faces

```
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

Look for faces in the image using the loaded cascade file

```
faces = face_cascade.detectMultiScale(gray, 1.1, 5)
```

Draw a rectangle around every found face

```
for (x,y,w,h) in faces:
    cv2.rectangle(image,(x,y),(x+w,y+h),(255,255,0),2)

cv2.imwrite('result.jpg',image)
```

Lab9-3 Video

- Detect the face and output to terminal dynamically

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
```

Initialize the camera and grab a reference to the raw camera capture

```
camera = PiCamera()
camera.resolution = (320, 240)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(320, 240))

time.sleep(2)
```

Capture frames from the camera

```
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array

    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
```

Clear the stream in preparation for the next frame

```
    rawCapture.truncate(0)
    if key == ord("q"):
        break
```

Lab9 bonus

- Face detect is controlled by temperature
  - Detect & Do nothing

Score

- Lab9-1 25%
- Lab9-2 25%
- Lab9-3 25%
- Lab9 bonus 15%
- Report 10%

Raspberry PI Tutorial
10 - Cloud platform

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

Abstract

- REST API Introduction
- Tutorial – restdb.io
- Coding

REST API Introduction

What is REST?

- REST (Representational State Transfer) 是軟體架構的設計風格
- 通常以4種HTTP動詞做為資料存取的方式
  1. GET – 讀取
  2. POST – 建立
  3. PUT – 更新
  4. DELETE – 刪除
- 以網址表示存取的資源
  - 想要知道所有使用者
    GET www.example.com/users
  - 想要知道一個使用者
    GET www.example.com/users/123

What is REST?

- 通常REST API都是透過HTTP的形式來呼叫
- 一個HTTP封包大致上可分為header及body
  - Header – 包含這個封包的屬性或授權
  - Body – 包含主要想傳遞的資料
- 除了GET不需要有body之外，其餘三種動作皆需要有body
- Body通常都是以JSON格式呈現 (偶爾也會有XML格式)

JSON?

```
{
    "title": "nothing",
    "content": "fkk"
}
```

- JSON (**J**ava**S**cript **O**bject **N**otation)是一種資料表示的格式
- 大概分為三種概念
  - Object： 以{開始、}結束表示一個物件
  - Array：以[開始、]結束表示一個陣列
  - Name value collection：以name: value的形一組值
- Value可以是string、number、bool三種型態

```
{
    "auth": {
        "identity": {
            "methods": ["password"],
            "password": {
                "user": {
                    "name": "admin",
                    "domain":{
                        "name": "Default"
                    },
                    "password": "123456"
                }
            }
        },
        "scope": {
            "project": {
                "name": "admin",
                "domain": {
                    "name": "Default"
                }
            }
        }
    }
}
```

Demo – REST API

Tutorial – restdb.io

What is restdb.io?

- 提供簡單REST API的資料庫
- 支援不同權限管制
- 支援在上面架設網站

Sign up

https://restdb.io

Create Database

Enter Database

Enter
Developer Mode

Get API key

restdb.io
Search everything
embedded-b0c6
development
Collections
System_log
System_jobs
Users
Email_outbound
Email_inbound
Pages/views
Media Archive
Embedded-B0c6
Database
Resources
API keys
Settings
Inspector
Name
Description
Menu
Access
system_log
Background log for this database
system_jobs
Background jobs for this database
users
email_out
email_inbound
Incoming email to this application
Description
Apikey
Public apikey (share with caution)
01333ac6194155143cc49015cf56f14498022
Add Collection
Add from Excel
Pages

Create Table (Collection)

Add Column (Field)

Happy Coding Time

But wait, before we start

Developer Tool

- REST API用的基本上都是HTTP封包 → 開始寫之前可以先用工具測試封包內容是否正確！
- 這邊我們介紹的是Chrome的應用程式 – Postman

GET Request

Runner
Import
Builder
Team Library
No Environment
GET
https://embedded-b0c6.restdb.io/rest/todo
Params
Send
Save
Authorization
Headers (3)
Pre-request Script
Tests
Code
x-apikey
01333ac6194155143cc49015cf56f14498022
Content-Type
Cache-Control
no-cache
Bulk Edit
Presets
Body
Cookies
Headers (18)
Tests
Status: 200 OK
Time: 911 ms
Pretty
Raw
Preview
JSON
[]

POST Request

記得headers也要跟GET Request一樣

Python Example

Python library

- 這邊我們要用到Python的requests library，安裝方式如下
- sudo apt-get install python-pip
  sudo pip install requests
- 這邊會需要用到的指令只有一種
  requests.request(type, url, data=..., headers=...)

Example – get data

```
import requests
url = "https://<your_database>.restdb.io/rest/<your_table>"
headers = {
          'content-type': "application/json",
          'x-apikey':  “<your_key>",
          'cache-control': "no-cache"
}
response = requests.request("GET", url, headers=headers)
print(response.text)
```

Requirements

- 在網頁上面按下拍照按鈕
(TA提供)
- Lab10-1：在Raspberry Pi上利用REST API獲取指令
(30%)
- Lab10-2：利用收到的指令控制camera拍照
(20%)
- Lab10-3：用OpenCV做臉部辨識並將結果用REST API上傳至雲端
(30%)
- Report：(20%)

Lab10-1

- 在restdb.io上面建立一個database，並建立一個叫做rpi的collection
- 在collection中加入name (text) 及value (text) 兩個欄位
- 用網頁/Postman/code/….在collection裡面加入status/ok, camera/take這兩筆資料
- 在Raspberry Pi上面用Python呼叫REST API將該資料印在螢幕上

Lab10-2

- 在rpi的collection中加入一筆camera/take的資料
- Raspberry Pi利用GET拿到這個資料
- 控制相機拍一張照片
- Raspberry Pi利用PUT將資料改為camera/done

Lab10-3

- 將Lab10-2照的相片透過OpenCV辨識有沒有人臉
- 如果有人臉→把rpi裡面的status改為yes
- 如果沒有人臉→把rpi裡面的status改為no

Raspberry PI Tutorial
10 - Cloud platform.2

Dept. of Computer Science and
Information Engineering
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 integrations
  - Worldwide community

Collect

Send sensor data to the cloud.

Analyze

Analyze and visualize your data with MATLAB.

Act

Trigger a reaction.

ThingSpeak Introduction

申請帳號

- ThingSpeak
  - https://thingspeak.com/

1. Thingspeak -> Device

- Goal: This example shows you how to **control sensor by Thingspeak** with a string. Your device will connect to that channel and get command periodic.
- Hardware Required
  - Raspberry PI (or Arduino)
  - a wired or wireless network connection
- Software required
  - ThingSpeak account, with a Channel ID and API key

1. Thingspeak -> Device

- 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 -> Arduino

Select “Channel”, create “New Channel”

1. Thingspeak -> Device

After creating new channel:

1. Thingspeak -> Device

Select “App”, choose “TalkBack”

1. Thingspeak -> Device

In TalkBack:

Set “Name”

Your own API Key

Set “Channel”

1. Thingspeak -> Device

- TalkBack ID: **12315** API Key: **O7FN66PKSX9R3JW4**

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:	yun_test

Arduino code

```
#include "Bridge.h"
#include "HttpClient.h"
```

載入相關的library

```
//ThingSpeak Settings
String thingSpeakAPI = "api.thingspeak.com";
String talkBackAPIKey = "FD8TBS8TEQN3ZNVZ";
String talkBackID = "12295";
const int checkTalkBackInterval = 15 * 1000;
// Time interval in milliseconds to check TalkBack
  (number of seconds * 1000 = interval)
```

設定ThingSpeak資訊

```
// Variable Setup
long lastConnectionTime = 0;
```

// config url: https://thingspeak.com/apps/talkbacks/12295
// GET http://api.thingspeak.com/talkbacks/12295/commands/execute?api_key=FD8TBS8TEQN3ZNVZ

Arduino code

```
void setup()
{
  // Initialize Bridge
  Bridge.begin();

  // Initialize Serial
  Serial.begin(9600);
  while(!Serial);
}
```

環境啟始化

```
void loop()
{
  // Check ThingSpeak for TalkBack Commands
  checkTalkBack();
  delay(checkTalkBackInterval);
}
```

反覆執行 checkTalkBack

Arduino code

```
void checkTalkBack()
{
  HttpClient client;
```

建立 talkBack 的網址

```
  String talkBackCommand;
  char charIn;
  String talkBackURL =  "http://" + thingSpeakAPI + "/talkbacks/" + talkBackID +
"/commands/execute?api_key=" + talkBackAPIKey;
  Serial.println(talkBackURL);

  // Make a HTTP GET request to the TalkBack API:
  client.get(talkBackURL);
```

讀取回傳值

```
  while (client.available()) {
    charIn = client.read();
    talkBackCommand += charIn;
  }
```

https://www.arduino.cc/en/Tutorial/HttpClient

Arduino code

回傳值 = TURN_ON, 開啟內建LED燈

```
  // Turn On/Off the On-board LED
  if (talkBackCommand == "TURN_ON")
  {
    Serial.println(talkBackCommand);
    digitalWrite(13, HIGH);
  }
  else if (talkBackCommand == "TURN_OFF")
  {
    Serial.println(talkBackCommand);
    digitalWrite(13, LOW);
  }
```

回傳值 = TURN_OFF, 關閉內建LED燈

```
  Serial.flush();
  delay(1000);
}
```

1. Thingspeak -> Device

In TalkBack:

設定 TURN_ON 或 TURN_OFF

Commands

Add a new command

Position	Command string	
LED	TURN_ON	Save

Commands

Position	Command ID	Command string
0	3514595	TURN_ON

Position	Command string	
LED	TURN_ON	Save

1. Thingspeak -> Device

- Use browser to check it

正常的話, 可以看到此訊息

http://api.thingspeak.com/talkbacks/12295/commands/execute?api_key=FD8TBS8TEQN3ZNVZ

1. Thingspeak -> PI

- 1. 在python裡, 抓TalkBack的command下來
  - 可以使用system call, 呼叫curl指令
  - Ex: curl http://api.thingspeak.com/talkbacks/.....
- 2. 當command為:
  - Turn_on: 開啟LED燈
  - Turn_off: 關閉LED燈

2. Device -> ThingSpeak

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

2. Device -> ThingSpeak

- In Channel, go to “Channel Settings”

2. Device -> ThingSpeak

- Set the number of filed and their name
  - Ex: Filed 1 = Temperature; Filed 2 = Humidity

2. Device -> ThingSpeak

- Use your own “Write API Key”
  - It is different from Talkback API key

Arduino code

```
// include all Libraries needed:
#include <Process.h>
```

ThingSpeak的Write API key

```
//ThingSpeak Settings
String APIKey = "Y9ETKJ5083AAZ854";
const int postingInterval = 15 * 1000;
//delay between updates to ThingSpeak
unsigned long lastRequest = 0;      // when you last made a request
String dataString = "";
```

```
void setup() {
  // start serial port:
  Bridge.begin();
  Serial.begin(9600);

  while(!Serial);    // wait for Network Serial to open
  Serial.println("ThingSpeak client");

  // Do a first update immediately
  updateData();
  sendData();
  lastRequest = millis();
}
```

環境啟始化

Arduino code

```
void loop() {
  // get a timestamp so you can calculate reading and sending intervals:
  long now = millis();

  // if the sending interval has passed since your
  // last connection, then connect again and send data:
  if (now - lastRequest >= postingInterval) {
    updateData();
    delay(2000);
    sendData();
    lastRequest = now;
  }
}
```

如果兩次Request的時間間隔超過postingInterval才會執行sendData

```
void updateData() {
  // convert the readings to a String to send it:
  dataString = random(10) + 20;
}
```

此處sensor data為隨機產生, 可替換成任何sensor資訊. Ex: 溫溼度, 照度…等

Arduino code

```
// this method makes a HTTP connection to the server:
void sendData() {
String url = "https://api.thingspeak.com/update";  // form the string for the URL parameter

  // Send the HTTP POST request
  Process p;
  Serial.print("\n\nSending data... ");

  String cmd = "curl --silent --request POST --header \"X-THINGSPEAKAPIKEY: " +APIKey;
  cmd = cmd + "\" --data \"field1=" + dataString;
  cmd = cmd + "\" http://api.thingspeak.com/update";
  p.runShellCommand(cmd);
  Console.println(cmd);
  p.close();

  Serial.println(cmd);
  Serial.println("done!");

  // If there's incoming data from the net connection, send it out the Serial:
  while (p.available()>0) {
    char c = p.read();
    Serial.write(c);
  }

}
```

利用YUN裡面的Linux系統, 執行curl送出HTTP POST

curl --silent --request POST --header "X-THINGSPEAKAPIKEY: Y9ETKJ5083AAZ854" --data "field1=1" http://api.thingspeak.com/update

2. PI -> ThingSpeak

- Use curl to send HTTP POST:

curl --silent --request POST --header "X-THINGSPEAKAPIKEY: **Y9ETKJ5083AAZ854**"
--data "field1=**value**" http://api.thingspeak.com/update

Sensor value

Your Write API key

傳送1與5到Thingspeak上

2. Device -> ThingSpeak

- ThingSpeak