



用Raspberry Pi 結合 Arduino 實現Wi-Fi 遙控車

用Python玩WiFi + S4A

講師: 楊松

主辦單位: 艾鍗學院 & 图5





Outline



Raspberry Pi 介紹

Python 語言介紹

Raspberry Pi + Python 應用案例介紹

Raspberry Pi + Python 開發無線 WiFi 遙控車介紹

- ESP8266 WiFi 模組介紹
- Motoduino 介紹
- S4A通訊協定介紹
- Raspberry Pi 透過 WiFi 模組控制 motoduino

Raspberry Pi + Python 開發機械手臂案例

如何快速學習Raspberry Pi 使用Python 控制Raspberry Pi子板

Raspberry Pi 介紹





樹莓派--Raspberry Pi

英國的樹莓派基金會所開發

基於Linux系統

信用卡大小電腦

低價硬體 - 35 美元

自由軟體

刺激學校基本電腦科學教育

為了英國的電腦科學教育...







Pi 是完整的小電腦



CPU: 900 MHz -ARM Cortex-A7

記憶體: 1GB

視訊輸出: Composite RCA

HDMI

音訊輸出: 3.5 mm jack

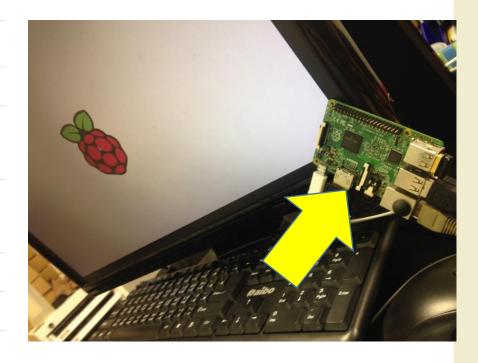
HDMI(1.3 & 1.4)

儲存: microSD

USB: USB 2.0 x4

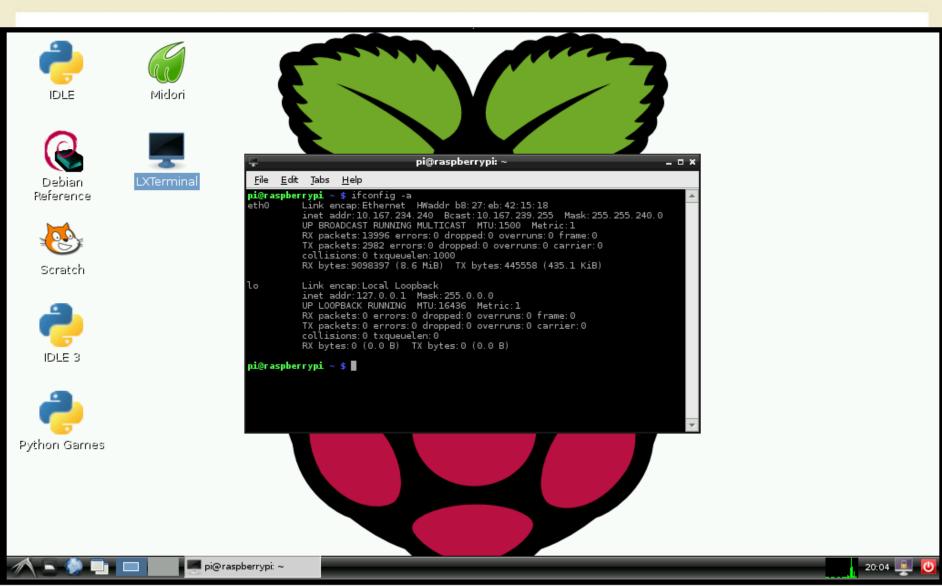
Ethernet: 10/100 RJ45

尺寸: 85mm x 56mm x 17mm



Linux 作業系統

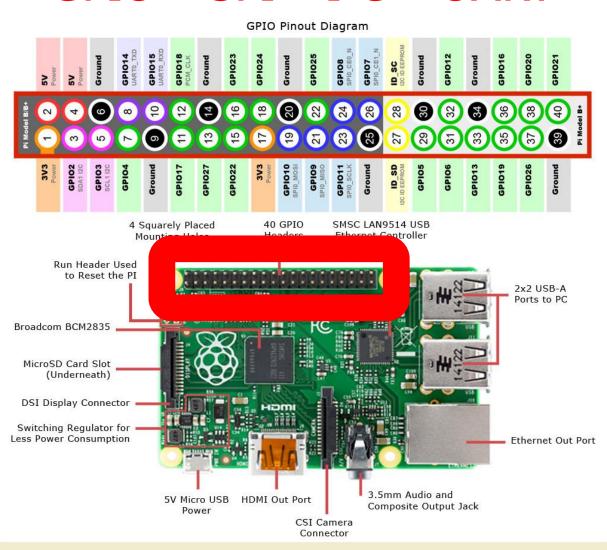




Pi 不單純只是電腦



GPIO + SPI + I²C + UART



有獎徵答



- 請問Raspberry Pi 是哪一個國家的基金會所開發?
 - •A-美國
 - •B- 英國
 - •C- 日本
 - D- 澳洲

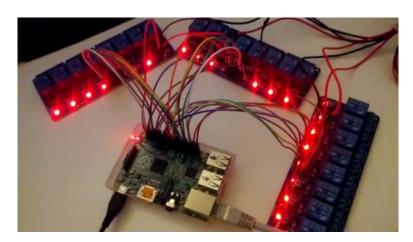
有獎徵答



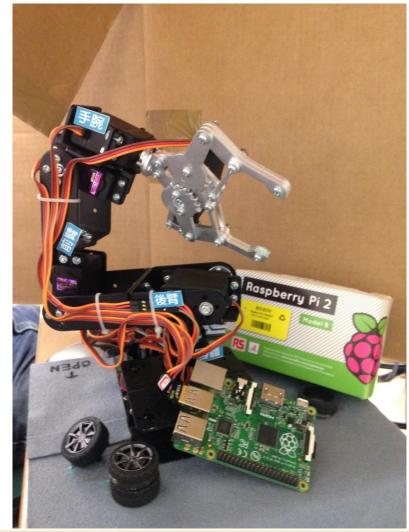
- 請問Raspberry Pi 的尺寸大小為何?
 - •A- 一個房間那麼大
 - •B-一台車子那麼大
 - •C-一張桌子那麼大
 - •D-一張名信片那麼大

Pi 控制與週邊硬體模組





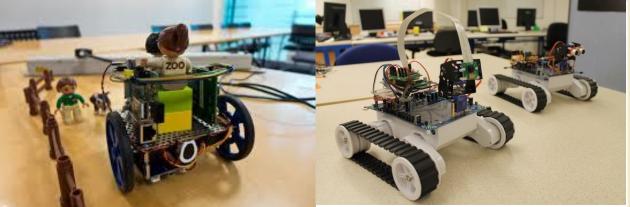




Pi 發揮創意產生應用









Coding on Linux





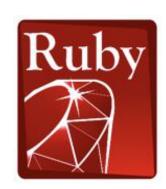






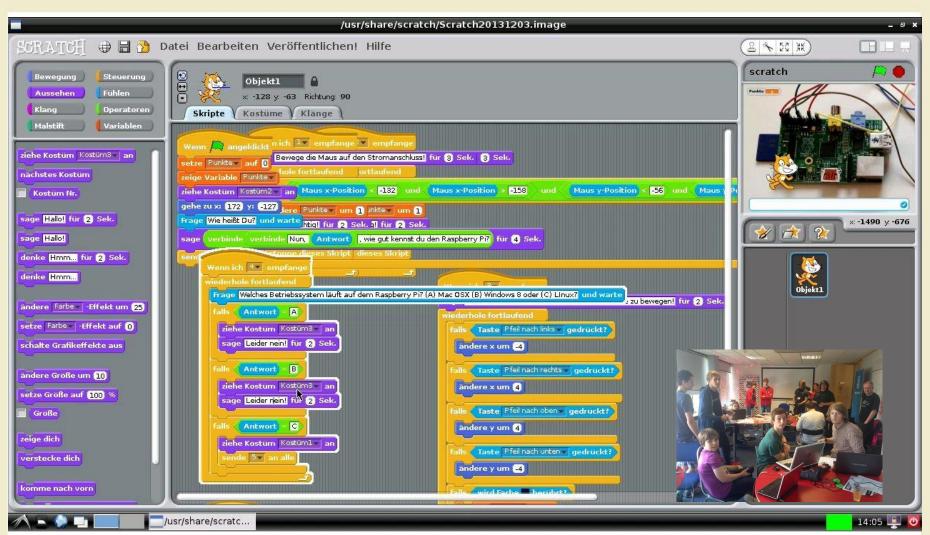






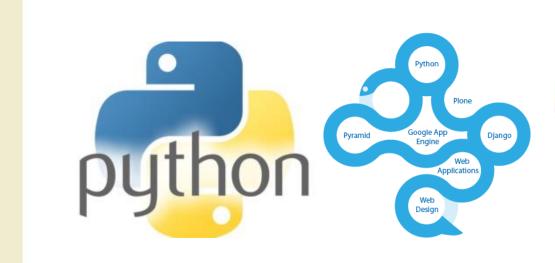
Scratch 程式語言



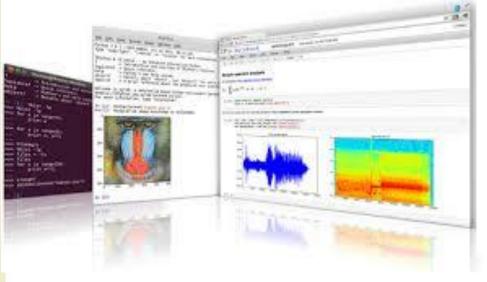


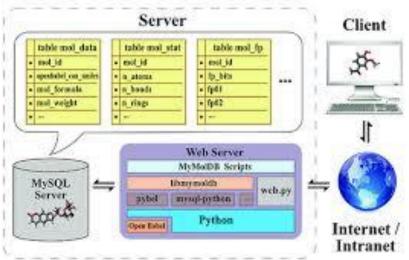
Raspberry Pi + Python











Why Python?



- 具物件導向特性的直譯式語言
- 跨平台(windows, linux, mac os x)
- 豐富的函式庫支援
 - 數值運算與科學繪圖: Numpy, Matplotlib...
 - GUI 圖形介面: Qt, Tkinter, wxPython...
 - •網路應用: Socket、Django、Flask、Turnado...
 - 資料庫: NoSQL、MySQL...
 - 週邊硬體控制: GPIO、I²C、SPI、UART、GPIB、USB

有獎徵答...



- 請問下列何者為 Raspberry Pi的作業系統?
 - A- Linux (Rasbian)
 - B- Windows CE
 - C- FreeRTOS
 - •D-Hurd

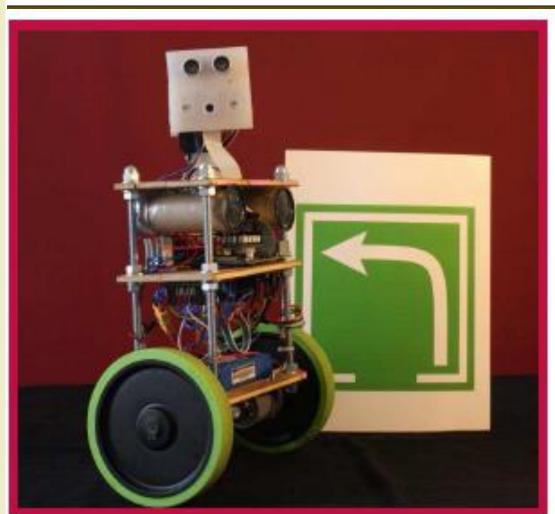
有獎徵答...



- 請問Raspberry Pi 的Linux作業系統中使用何種 圖型化程式語言做為教育使用?
 - A- LabView
 - B- AppInventor
 - C- Scratch
 - D- PureData

Robot: Raspberry Pi + Python



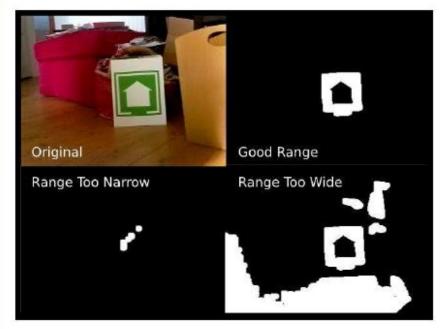






Python + OpenCV



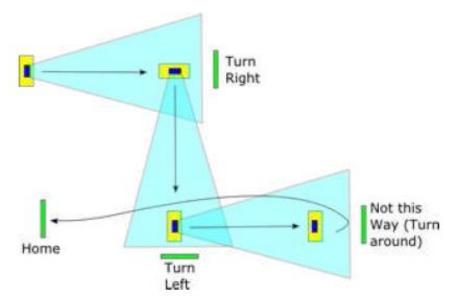




Matches between symbol and real scene



Robot tracking symbols Turn Back, Turn Left, Home, Turn Right





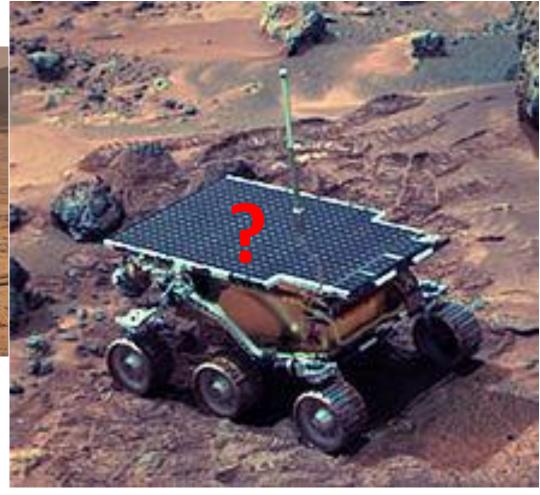


用Raspberry Pi 結合 Arduino 實現Wi-Fi 遙控車

想象圖

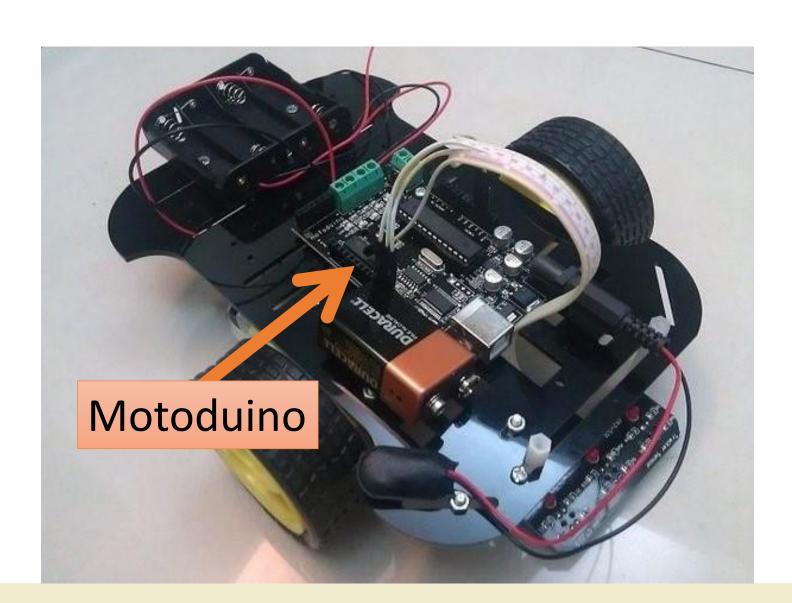






Motoduino=Arduino+Motor Driver





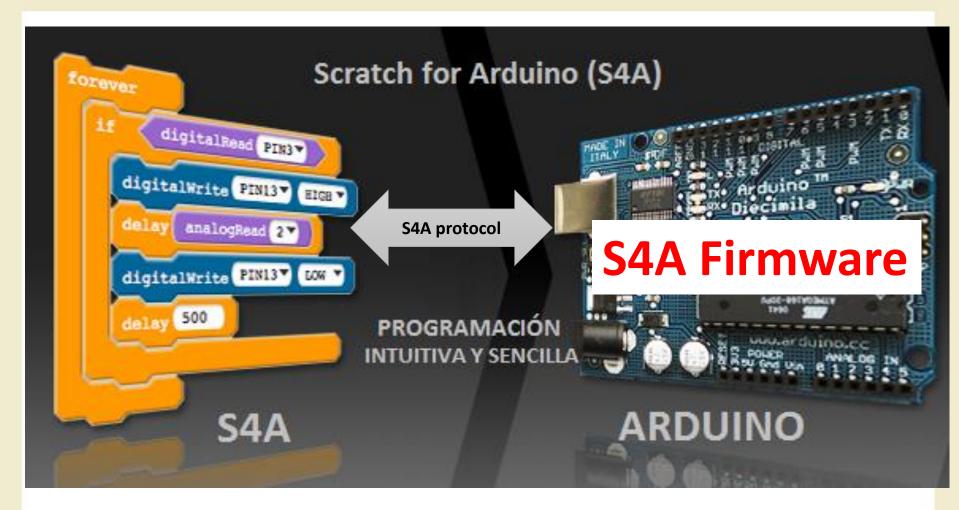
Arduino & C++



```
Blink
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press reset:
void setup () {
  // initialize the digital pin as an output.
 pinMode (led, 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
```

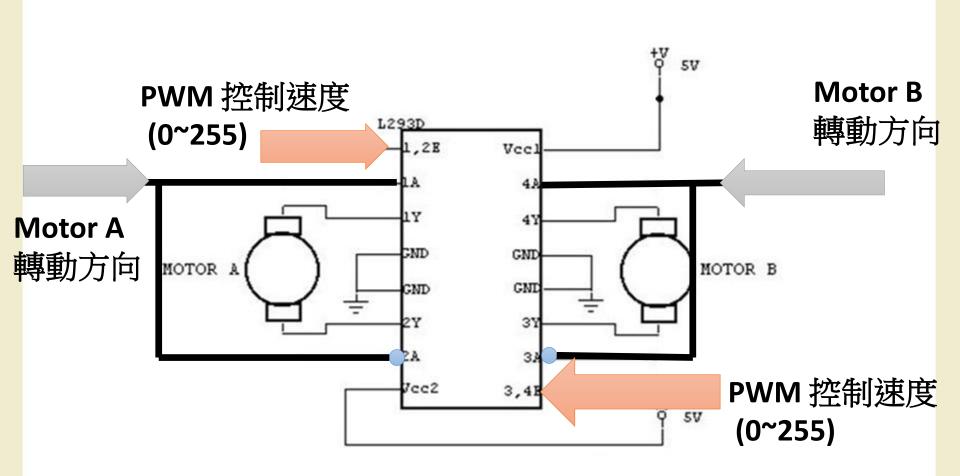
Scratch For Arduino (S4A)





Motorduino = Arduino + Motor Controlling (L293D)





又是有獎徵答...



- 請問 S4A 的 "S" 指的是?
 - A- Scratch
 - B- Super-Man
 - C- Seven-Eleven
 - D- Star-War

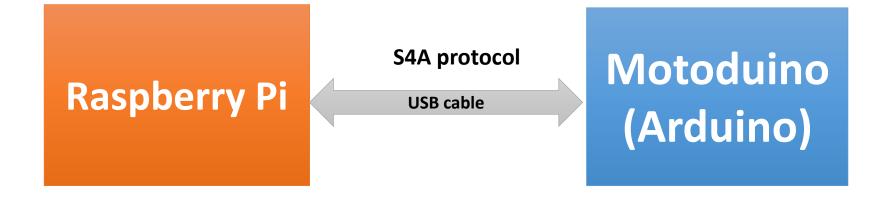
又是有獎徵答...



- 請問 Motoduino 是有哪一個國家的廠商所發展 出來的?
 - •A-沙鳥地阿拉伯
 - •B- 剛果
 - •C- 台灣
 - D- 烏克蘭

S4A 架構



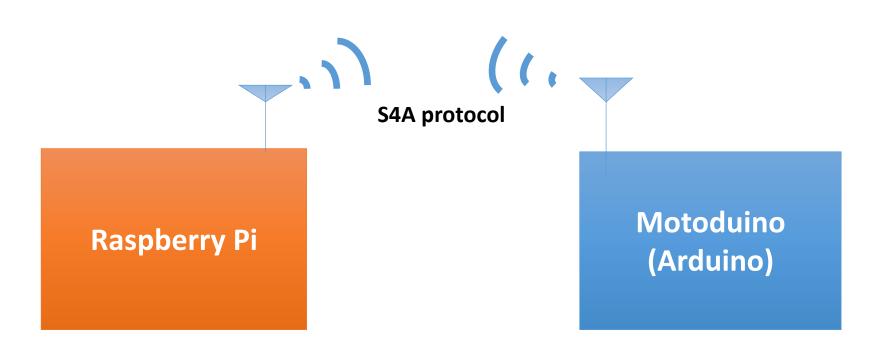


- Linux OS
- S4A using scratch

- S4A Firmware
- Motor driver (L293D)

Wireless S4A 架構





- Linux OS
- S4A protocol implemented by Python socket
- Wifi with AP mode

- S4A Firmware
- Motor driver (L293D)
- Wifi with STA mode (ESP8266)

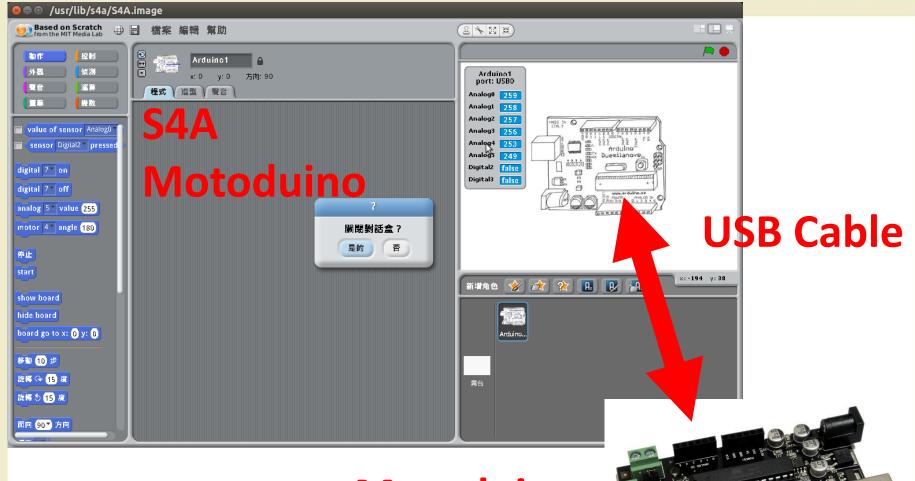
Labs



- Lab 0 : Research Motoduino
- Lab 1 : Read Packet
- Lab 2 : Resolve Packet
- Lab 3 : Send Packet
- Lab 4 : GUI (tkinter)
- Lab 5: ESP8266 UDP connection
- Lab 6 : Receive socket
- Lab 7 : Send socket
- Lab 8 : Move it !

S4A and Motoduino

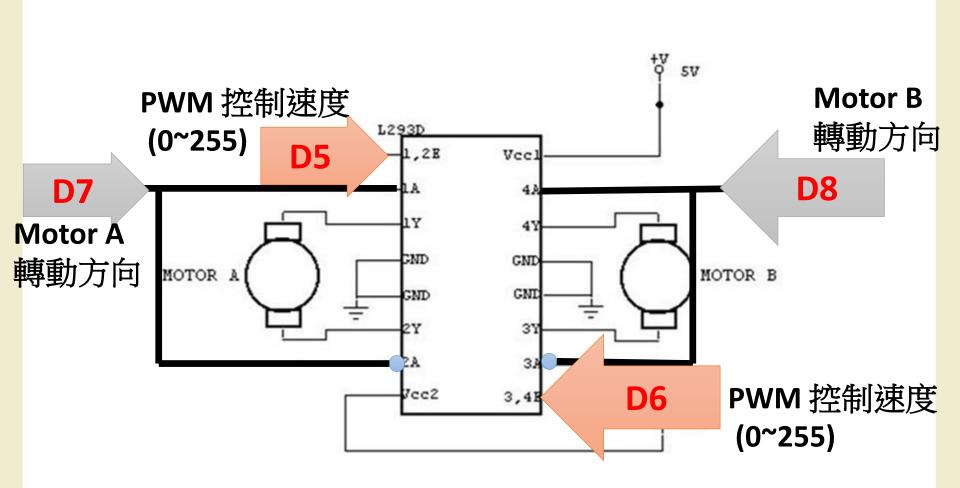




Motoduino Firmware ■

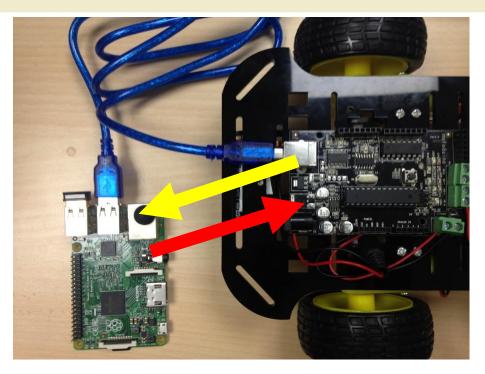
Controlling Motor

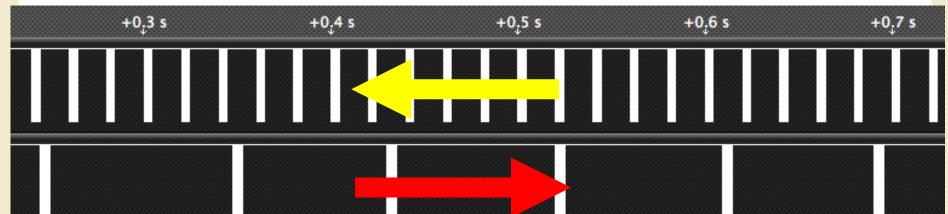




S4A Protocol – 訊號擷取







S4A Protocol – 訊號觀察



Data Stream: Motoduino -> Pi

- 每 16 ms 回報一次
- 一次回報 channel 0 ~ 7

Data Stream: Pi -> Motoduino

- 每 85 ms 下指令一次
- 一次帶有 channel 4 ~ 14 的值

S4A Protocol Spec.



Motoduino Motor Control

Actuators messages (Data

S4A Name	Motor 4	Analog 5	Analog 6	Motor 7	Motor 8	.10g 9	Digital 10	Digital 11	Motor 12	Digital 13
Туре	Continuous rotation Servomotor	Pseudo analog (PWM) Output	Pseudo analog (PWM) Output	Continuous rotation Servomotor	Servomotor (Angle control)	Pseudo analog (PWM) Output	Digital Output	Digital Output	Servomotor (Angle control)	Digital Output
Arduino I/O Pin	D4	D 5	D6	D 7	D8	D9	D10	D11	D12	D13
HEX value	A0 0d	Ap pp	Вр рр	B8 0d	Ca aa	Cp pp	D0 0v	D8 0v	Ea aa	E8 0v
BIN value	1010 0000	1010 100P	1011 000P	1011 1000 0000 000d	1100 00AA	1100 100P	1101 0000	1101 1000	1110 00AA 0AAA aaaa	
	0000 00 dd	PPP pppp	OPPP pppp	0000 000	OAAA aaaa	PPP pppp	0000 0000	0000 000 V	UAAA aaaa	0000 000

Sensors messages (Data from Arduino to Scratch)

S4A Name	Analog 0	Analog 1	Analog 2	Analog 3	Analog 4	Analog 5	Digital 2	Digital 3
Туре	Analog input	Digital Input	Digital Input					
Arduino I/O Pin	A0	A1	A2	A3	A4	A5	D2	D3
HEX value	8R Rr	8R Rr	9R.Rr	9R.Rr	AR Rr	AR Rr	BR Rr	BR Rr
BIN value	1000 ORRR ORRR mm	1000 1RRR 0RRR mm	1001 ORRR ORRR mr	1001 1RRR 0RRR mr	1010 ORRR ORRR mr	1010 1RRR 0RRR mm	1011 ORRR ORRR mr	1011 0RRR 0RRR mr

red = channel number

blue = channel data (CAPITAL means MSBs)

green = LSB of each channel

orange = protocol defined bit values

grey = padding bytes

d = direction (00=motor off, 01=clockwise, 10=anticlockwise)

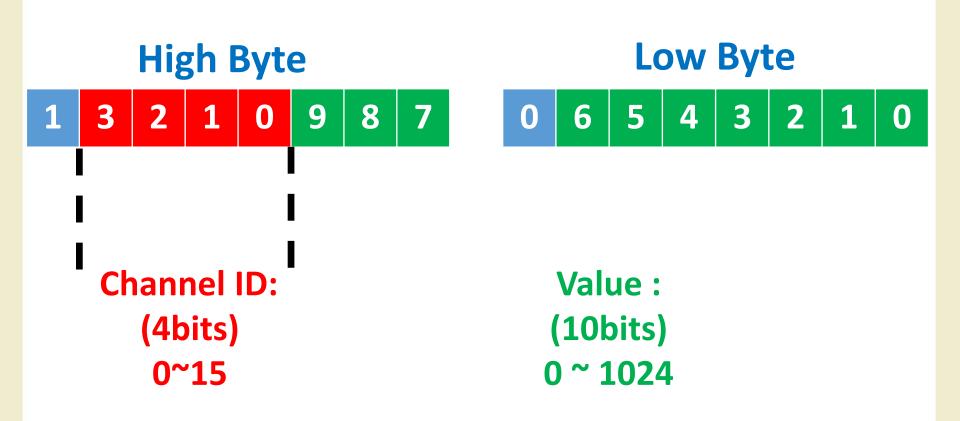
P/p = pwm value (values from 0 to 255)

A/a = angle (values from 0 to 360)

R/r = reading from sensor

解析封包格式





2 Bytes: HL-identify(2bit) +Channel ID (4 bits) + Value (10 bits)



Decoding

```
def pack_to_data(data):
    dev_id = (data[0] & 0b01111000) >> 3
    dev_val = ((data[0] & 0b00000111 ) << 7) | (data[1] & 0b01111111)
    return (dev_id, dev_val)</pre>
```

Encoding

```
def data_to_pack(dev_id,dev_val):
    data[0] = 0b100000000|( dev_id<<3)|((dev_val>>7)&0b00000111)
    data[1] = (0b00011111111&dev_val)
    return bytes([data[0],data[1]])
```

有獎徵答(快結束了) ...



- 請問 S4A 傳送封包的最小單位為何?
 - A- 2 byte
 - B- 1k byte
 - C- 1M byte
 - D- 1T byte

有獎徵答(快結束了) ...



- 請問 Motoduino 的小車每隔多久會傳一次訊息 給Raspberry Pi?
 - •A-16 ms(毫秒)
 - •B-1分鐘
 - •C-10分鐘
 - D- 30 分鐘

Labs



- Lab 0 : Research Motoduino
- Lab 1 : Read Packet
- Lab 2 : Resolve Packet
- Lab 3 : Send Packet
- Lab 4 : GUI (tkinter)
- Lab 5: ESP8266 UDP connection
- Lab 6 : Receive socket
- Lab 7 : Send socket
- Lab 8 : Move it !

Lab 1 : Read Package



```
#!/usr/bin/env python3
import serial
ser = serial.Serial("/dev/ttyUSBO",38400,8,'N',1)
while(True):
    print(ser.read(16))
```

撰寫程式讀取 Motoduino 傳過來的訊息

a8F\xb0\x00\xb8\x00\x80J\x88I\x90H\x98I\xa0H

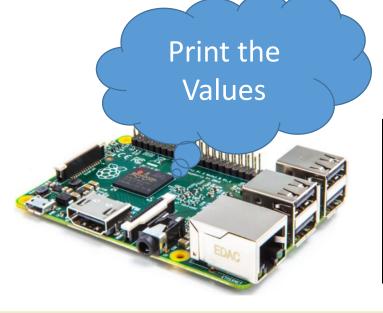


Lab 2 : Resolve packet



```
ser = serial.Serial('/dev/ttyUSB0',38400,8,'N',1)
inputs = [0,0,0,0,0,0,0,0,0,0,0,0,0,0]

while(True):
    data = ser.read(2)
    if data[0] & 0b100000000:
        dev_id, dev_val = pack_to_data(data)
        inputs[dev_id] = dev_val
        print(inputs)
    else:
        data = ser.read()
```



解析封包格式並取出各 channel 的資訊

Lab 3: Send Packet



Main loop:指令輸入

```
if __name__ == "__main__":
    s4a = s4a_slave('/dev/ttyUSB0')
    s4a.start()
    while(True):
        cmd = input().split()
        s4a.set dev(int(cmd[0]),int(cmd[1]))
```

```
pi@raspberrypi ~/s4a
6 255
6 0
6 255
7 0
7 1
7 0
```

Thread 2: 處理封包傳送



Lab 4 : GUI (Tkinter)



```
if
   name ==" main ":
    car = s4a.s4a('/dev/ttyUSB0')
    car.start()
    cars.append(car)
    root = tk.Tk()
    up btn = tk.Button(master=root, text='UP', command=up btn call
    down btn = tk.Button(master=root, text='DOWN', command=down btn
    left btn = tk.Button(master=root, text='TURN LEFT', command=turn
    right btn = tk.Button(master=root, text='TURN RIGHT', command=tur
    stop btn = tk.Button(master=root, text='STOP', command=stop btn c
    system off = tk.Button(master=root, text='disconnect'
    up btn.grid(row=0,column= 1, sticky=tk.
    down btn.grid(row=2,column=1, sticky=tk
                                                         UP
    left btn.grid(row=1,column=0, sticky=tk
    right btn.grid(row=1,column=2, sticky=t
    stop btn.grid(row=1,column=1, sticky=tk
    system off.grid(row=2,column=2, sticky=
                                             TURN LEFT
                                                        STOP
                                                               TURN RIGHT
    root.mainloop()
                                                               disconnect
                                                        DOWN
```

有獎徵答(最後一次) ...



- 請問 Motoduino 與 Raspberry Pi 的預設連接 方式為何?
 - •A- USB 連接線
 - •B-光纖通訊
 - •C- 雷射通訊
 - D- 電話線通訊

有獎徵答(最後一次) ...



- 請問接下來的的實作要將 Motoduino 和 Raspberry Pi 的通訊方式轉換為何?
 - A- WiFi 無線通訊
 - •B-光纖通訊
 - •C- 雷射傳輸通訊
 - D- 3G通訊

Labs



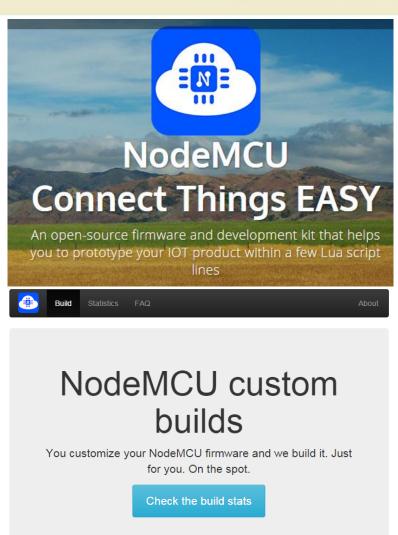
- Lab 0 : Research Motoduino
- Lab 1 : Read Packet
- Lab 2 : Resolve Packet
- Lab 3 : Send Packet
- Lab 4 : GUI (tkinter)
- Lab 5: ESP8266 UDP connection
- Lab 6: Receive socket
- Lab 7 : Send socket
- Lab 8 : Move it !

WiFi module: ESP8266



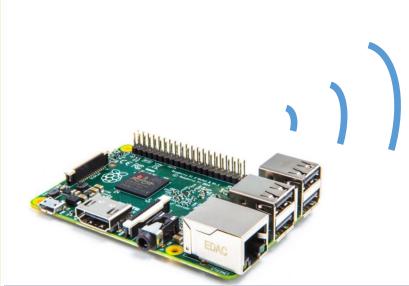


UART Interface
AT Commands
WiFi: Soft-AP, STA mode
TCP/UDP/IP Protocol



Lab 5: ESP8266 by UDP connection





```
#!/usr/bin/env python3
import socket

address = ('',12345)
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(address)

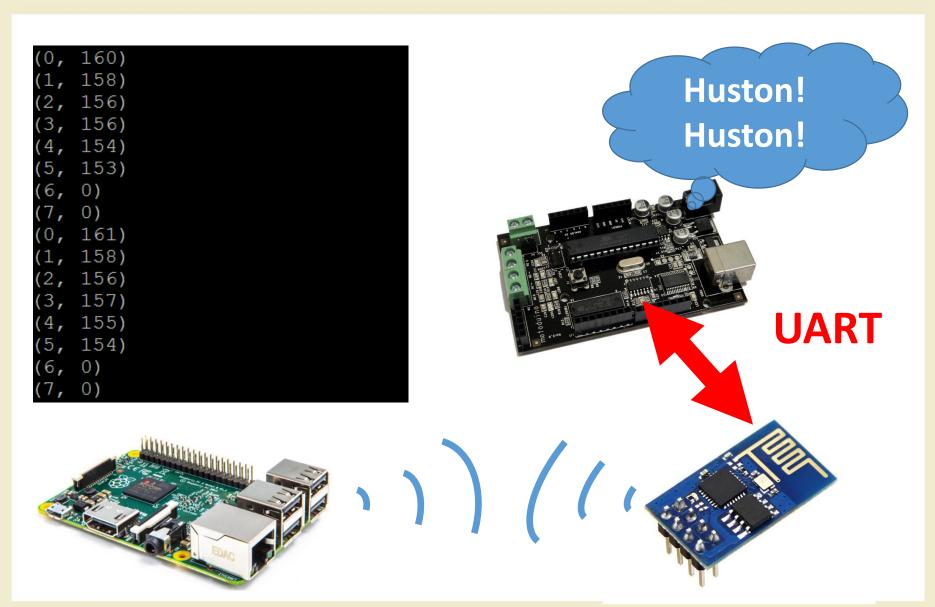
while(True):
    print('--')
    data, addr = s.recvfrom(1024)
    print(addr)
    print(data)
    s.sendto("hello\n".encode(), addr)
```



```
print("ESP8266 START")
ssid = "My_AP"
passwd = "12345678"
device id = "01"
scratch_ip = "192.168.0.1"
scratch port = 12345
                                            NodeMCU
wifi.setmode(wifi.STATION)
wifi.sta.config(ssid,passwd,1)
                                  Connect Things EASY
tmr.alarm(1,1000, 1, function()
    if wifi.sta.getip()==nil the
                                  you to prototype your IOT product within a few Lua scrip
        print(" Wait to IP addre
    else
        print("New IP address is "..wifi.sta.getip())
        tmr.stop(1)
        sck = net.createConnection(net.UDP)
        sck:on('receive', function(sck,pl) uart.write(0,pl) end)
        uart.on('data',0, function(data) sck:send(data) end,0)
        sck:connect(scratch_port,scratch_ip)
        sck:send("START UART Tunnel\n")
        uart.setup(0,38400,8,0,1,0)
    end
end)
print(wifi.sta.getip())
```

Lab 6 : Receive Package

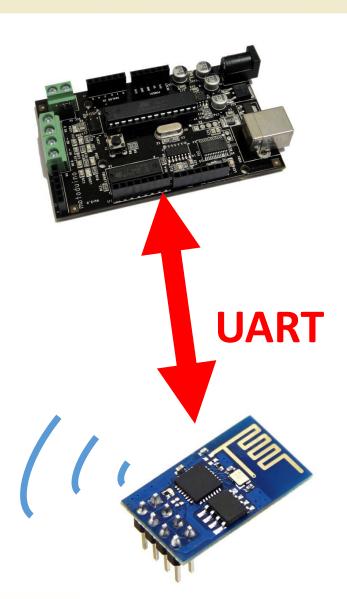




Lab 7: Web Send Package



```
def main loop(self):
   address = ('', 12345)
   s = socket.socket(socket.AF INET, socket.SOCK DGRAM)
   s.bind(address)
   #count =0
   while (True):
       data, addr = s.recvfrom(80)
       self.handle data(data)
       data = b''
       #if(count >= 1):
           count = 0
       for i in range (4,14):
          data += self.data to pack( i , self.pin outputs[i]
       s.sendto(data, addr)
       #count += 1
if name == " main ":
     ss = s4aWeb()
     ss.start()
    while (True):
         cmd = input().split()
          ss.set dev(int(cmd[0]), int(cmd[1]))
                 Do You
               Hear Me?
```



Lab 8: Move it





■ tk		
	UP	
TURN LEFT	STOP	TURN RIGHT
	DOWN	disconnect



Summery of Labs



- Lab 0 : S4A protocol for controlling Motor
- Lab 1: Read Packet
- Lab 2 : Resolve Packet
- Lab 3: Send Packet
- Lab 4 : GUI (tkinter)
- Lab 5: ESP8266 UDP connection
- Lab 6: Receive socket
- Lab 7 : Send socket

Learn...



- Scratch, Python, Lua
- Arduino (MCU)
- UART communication
- Protocol Design
- Motor Control
- WiFi communication: TCP/IP
- GUI Design



Thank For You Attention





Pi + Python 開發案例過程 六軸機械手臂

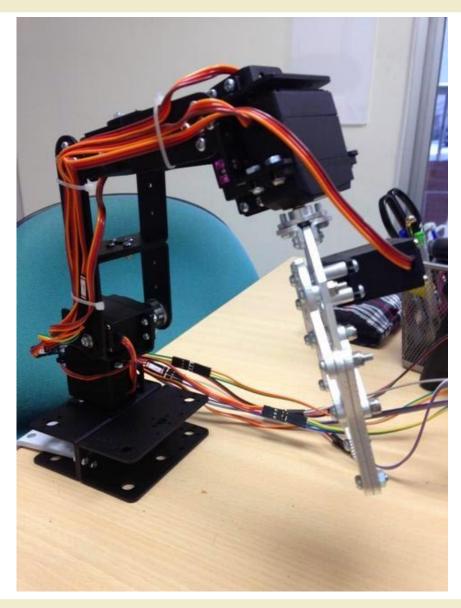
想象圖





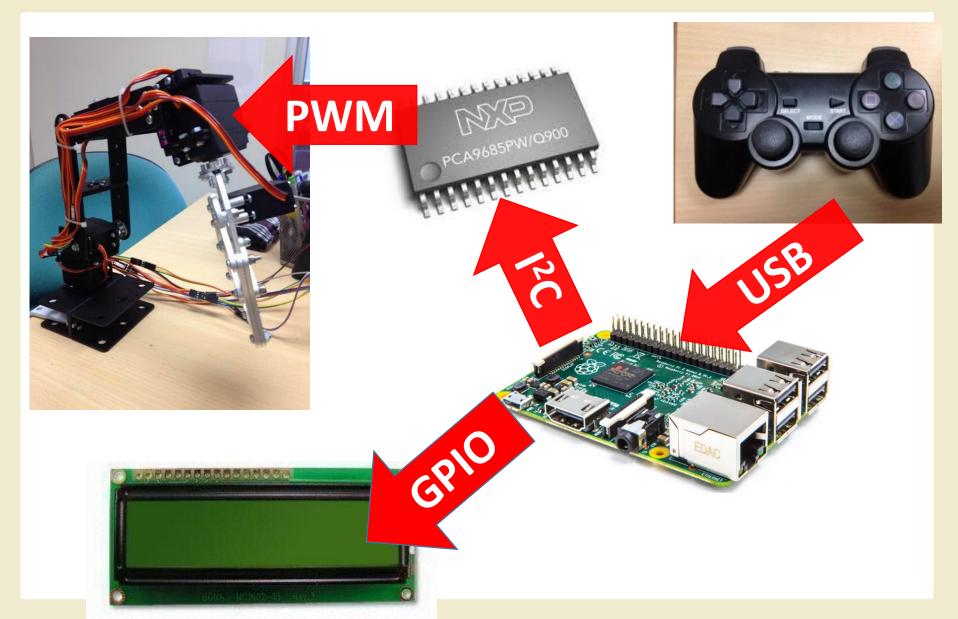
現成套件





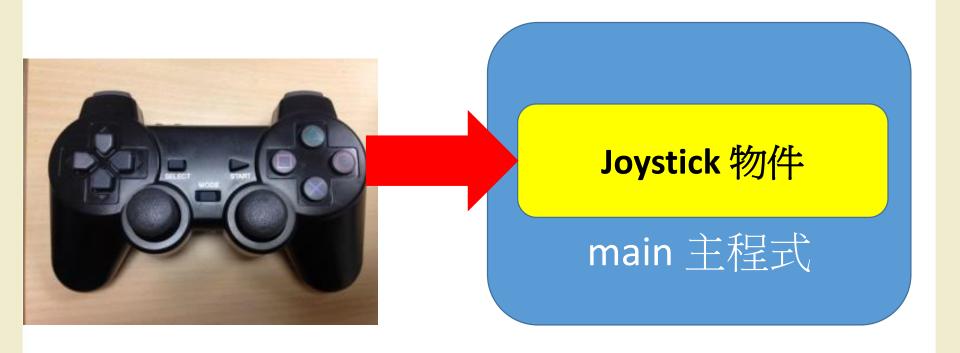
硬體架構





讀取USB搖桿指令





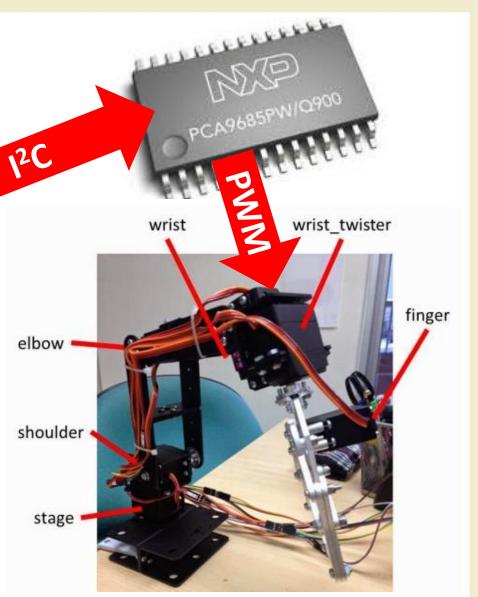
執行對應程式



main 主程式

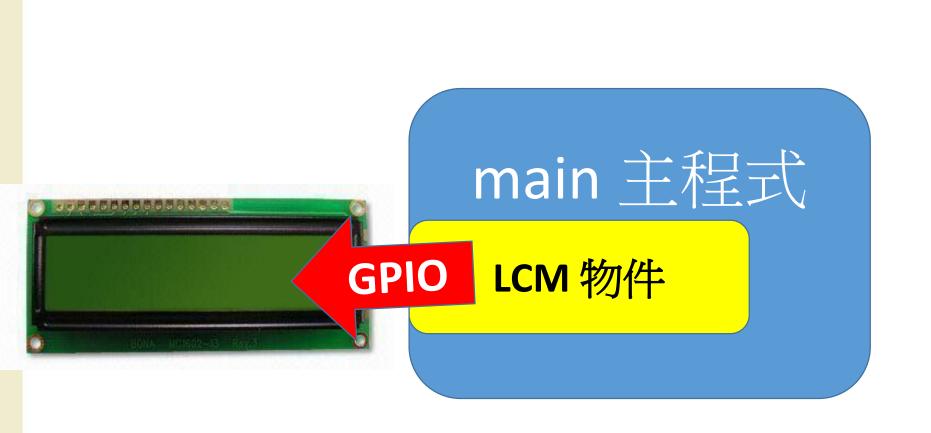
Arm 物件 arm

PCA9685 物件 pwm



字元型態LCD顯示資訊





Learn ...



- Python 開發
- I²C 通訊協定
- PWM 訊號
- 伺服馬達
- GPIO控制字元型態LCD
- USB HID Joystick 控制
- 控制系統設計