前往下載教學資源
www.geniopy.com.tw

Pomas

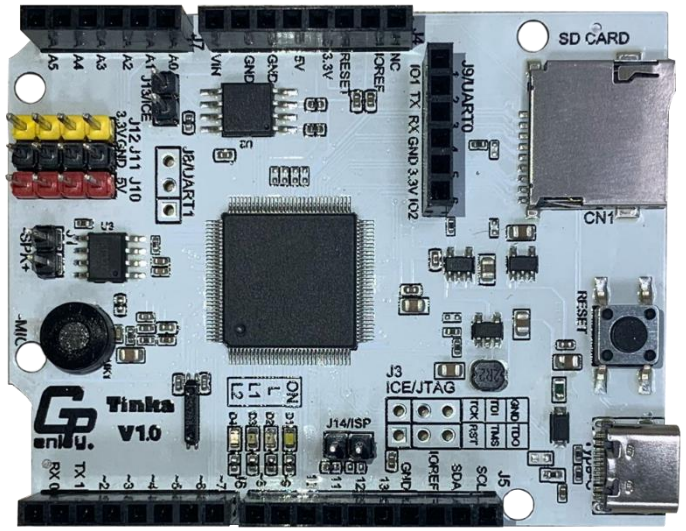
MicroPython AIOT

名威智慧通訊

專案講師 林羿君 Lynn Lin

架構

STRUCTURE



1 感測元件應用篇

❶ 用 ADC 腳位讀入光感測模組偵測亮度

❷ 用 IO 腳位讀入超音波模組感測距離

❸ 用 PWM 腳位控制伺服馬達轉動

2 通訊模組應用篇

❶ 藍芽模組與手機APP連線應用

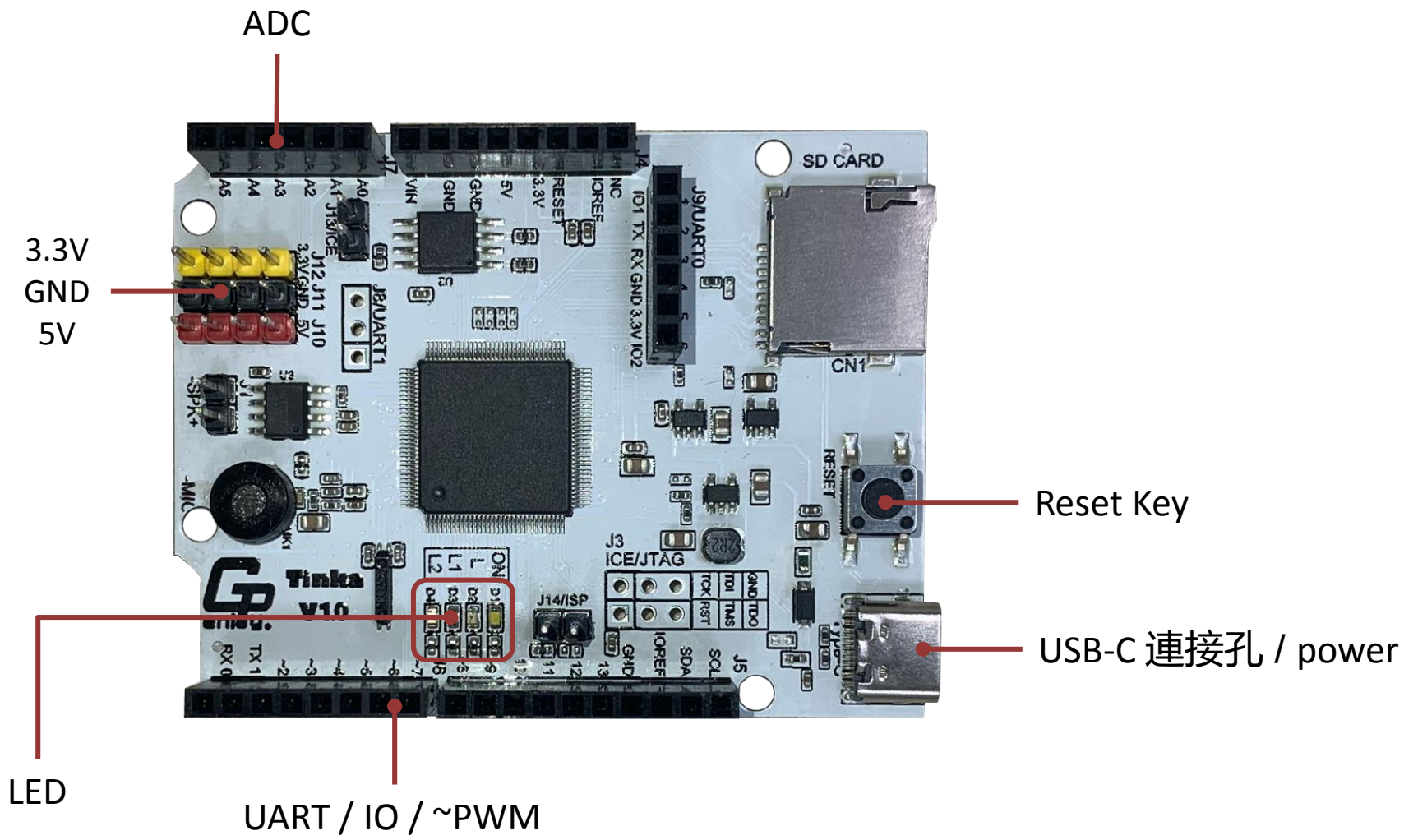
❷ 感測元件、通訊模組與 GPIO 整合應用

0

PART

Tinka

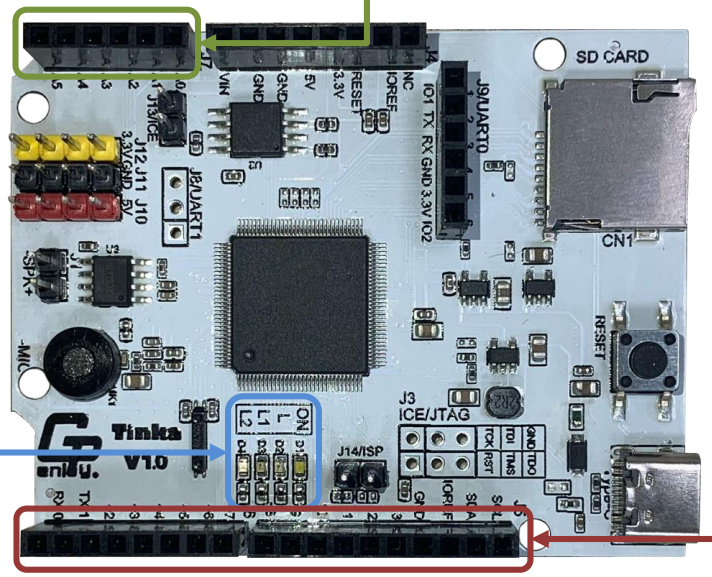
硬體介紹


ADC
3.3V
GND
5V
SD CARD
CN1
Reset Key
USB-C 連接孔 / power
LED
UART / IO / ~PWM
Tinka
V10
J3
ICE/JTAG
J14/ISP
J9/UART0
J8/UART1
J13/ICE
J12 J11 J10
RESET

輸出 / 入腳位對照表

	A5	A4	A3	A2	A1	A0
Port	A6	A7	E11	E10	E9	E8

	L2	L1	L	ON
Port	F0	F1	F2	Power led



	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO12	IO13
Port	B11	B12	A8	A9	A10	A11	A12	A13	A14	A15	D13	D12	D11	D10
ALT	UART RX	UART TX	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7	PWM8	PWM9	SPI CS	SPI CLK	SPI MOSI	SPI MISO

1

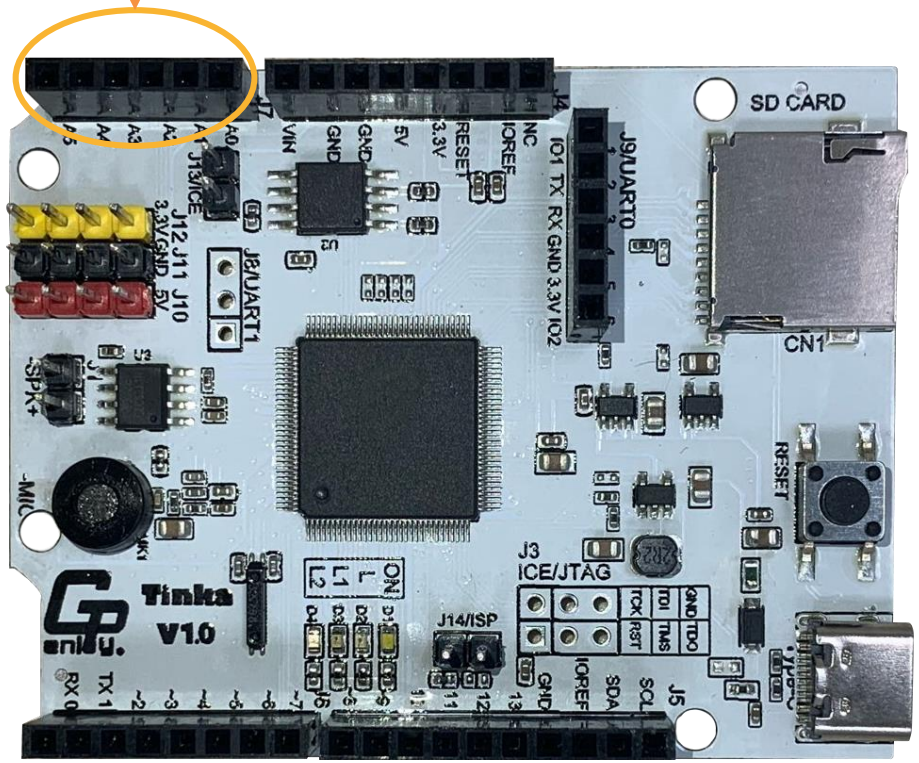
PART

光感應自動燈

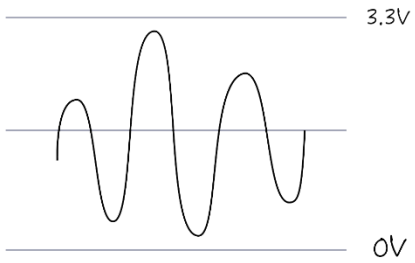
類比轉數位輸入 (Analog-to-Digital)

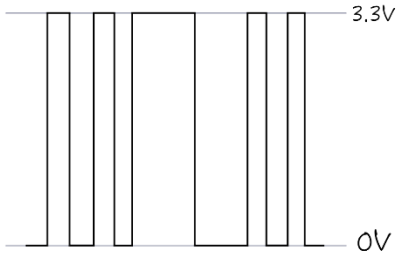
原理

ADC腳位
Analog-to-Digital Converter



	A5	A4	A3	A2	A1	A0
Port	A6	A7	E11	E10	E9	E8
ALT	ADC(5)	ADC(4)	ADC(3)	ADC(2)	ADC(1)	ADC(0)

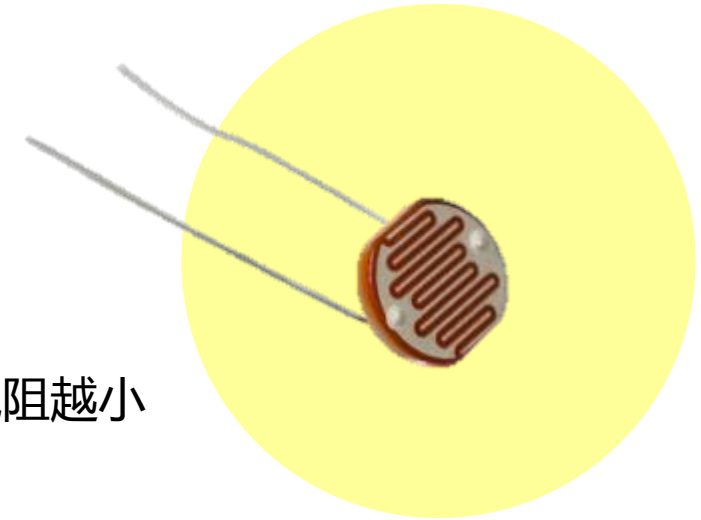




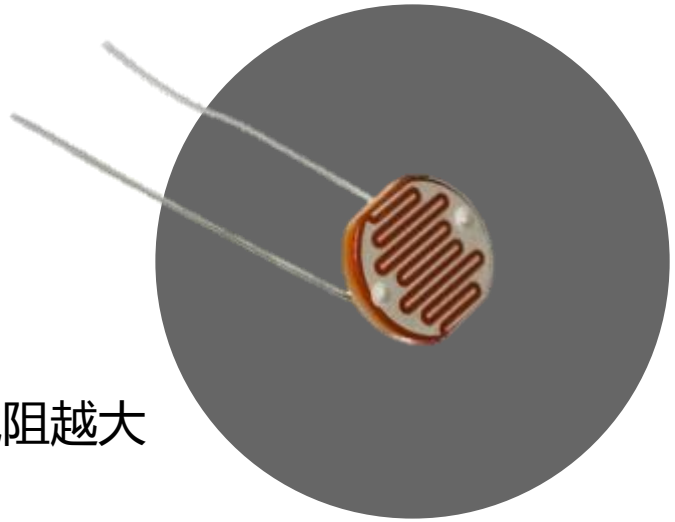
圖片來源：旗標科技 超圖解 Python 物聯網

光敏電阻

原理



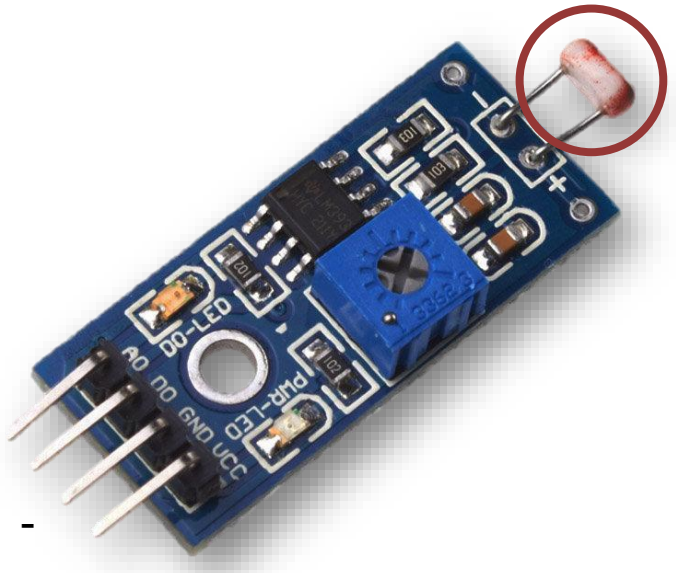
光越亮電阻越小



光越暗電阻越大

暗 ⟷ 亮

電壓高　　電壓低



AO：電壓變化

GND：-

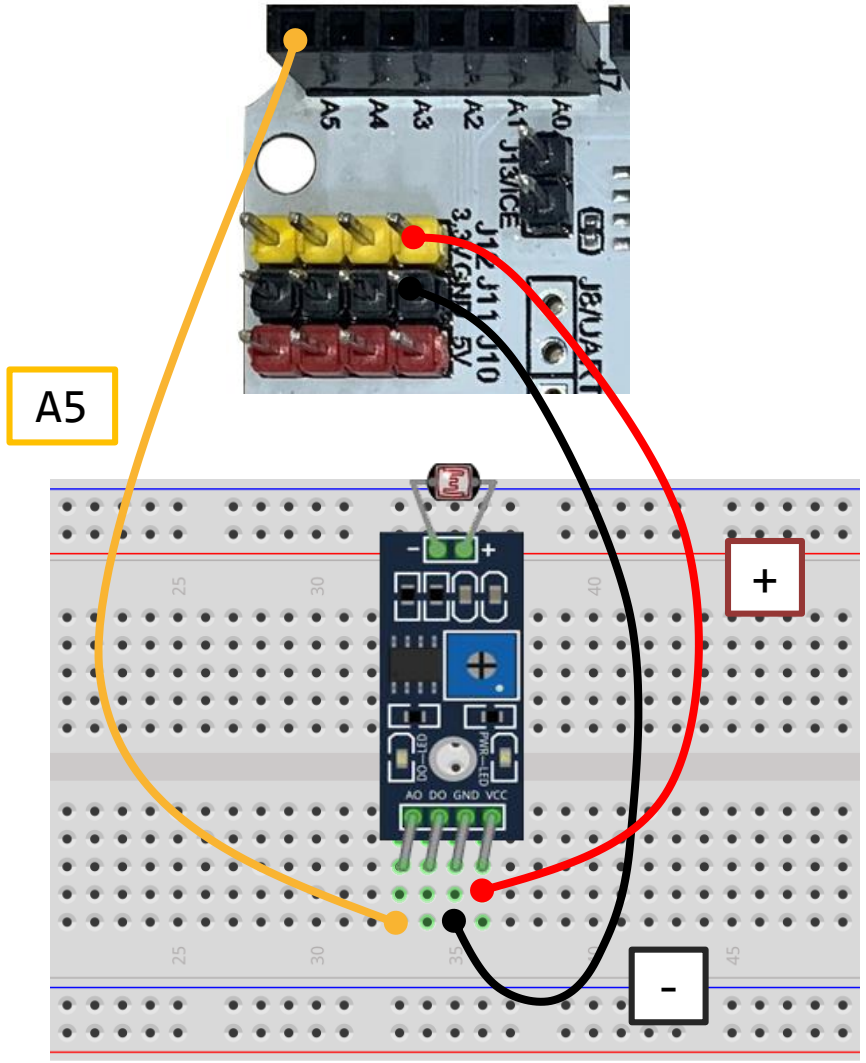
VCC：3.3V

控制板無法直接**偵測光敏電阻的變化，**
不過我們使用的是現成的**模組**
內建電路把**電阻轉成電壓**變化

圖片來源：旗標科技 超圖解 Python 物聯網

動手做

接線



在互動窗格做個小小測試吧!
利用正常環境下所讀取的亮度值來做出屬於自己的光感測燈。

本實驗會使用 A5(ADC5)
讀取光敏感測數據

```
>>> from machine import ADC
>>> light = ADC(5)
>>> light.read()
```

動手做 寫程式製作光感應燈

可參考sample code_light.py

```
1  from machine import Pin,ADC      # 匯入Pin,ADC類別
2  from gpb import LED,delay        # 匯入自訂LED,delay模式
3
4  led = LED(1)
5  light = ADC(5)                   # 設定ADC5為感測物件
6
7  while True:
8      if light.read()>1500:        # 根據實驗值調整值
9          led.on()                 # 太暗亮燈
10     else:
11         led.off()                # 夠亮熄燈
12     print(light.read())
13     delay(50)                    # 等50毫秒
```

注意：閥值要根據剛才的實驗結果做調整唷!

這裡有兩層縮排
第一層是 while
第二層是 if 結構

```
if 條件判斷式:
    成立時要做的動作
    成立時要做的動作
    ...
else:
    不成立時要做的動作
    不成立時要做的動作
    ...
```

❶ 注意 if else 尾端的**冒號**以及**縮排**

❷ > 會判斷左邊的值是否**大於**右邊
判斷式會依此得 True 或是 False

實驗結果：感測器也是開關!!

2

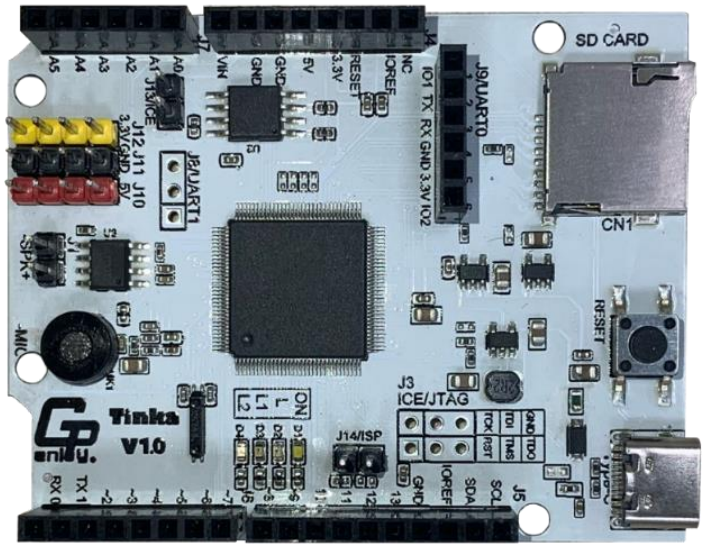
PART

溫濕度感測站

DHT11 溫濕度感測器

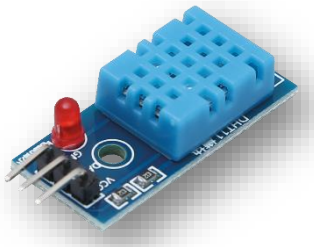
原理

數位溫濕度模組

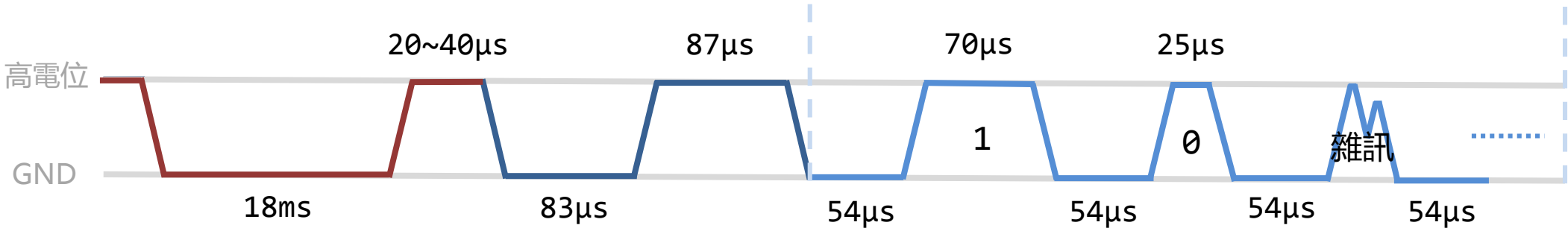


雙向通訊

使用單一傳輸線做輸出/入雙用途



Data Out 數據傳輸

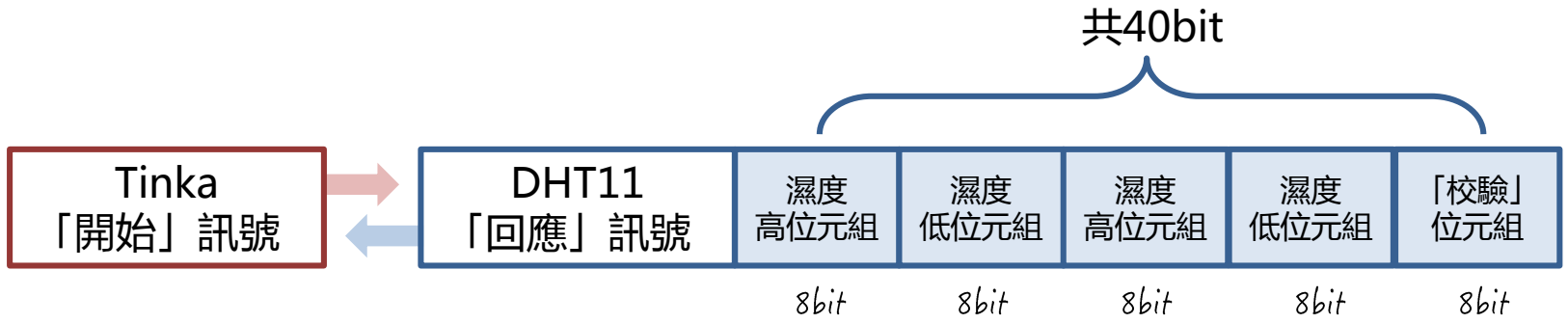


圖片來源：旗標科技 超圖解 Python 物聯網

原理

GPIO 腳位

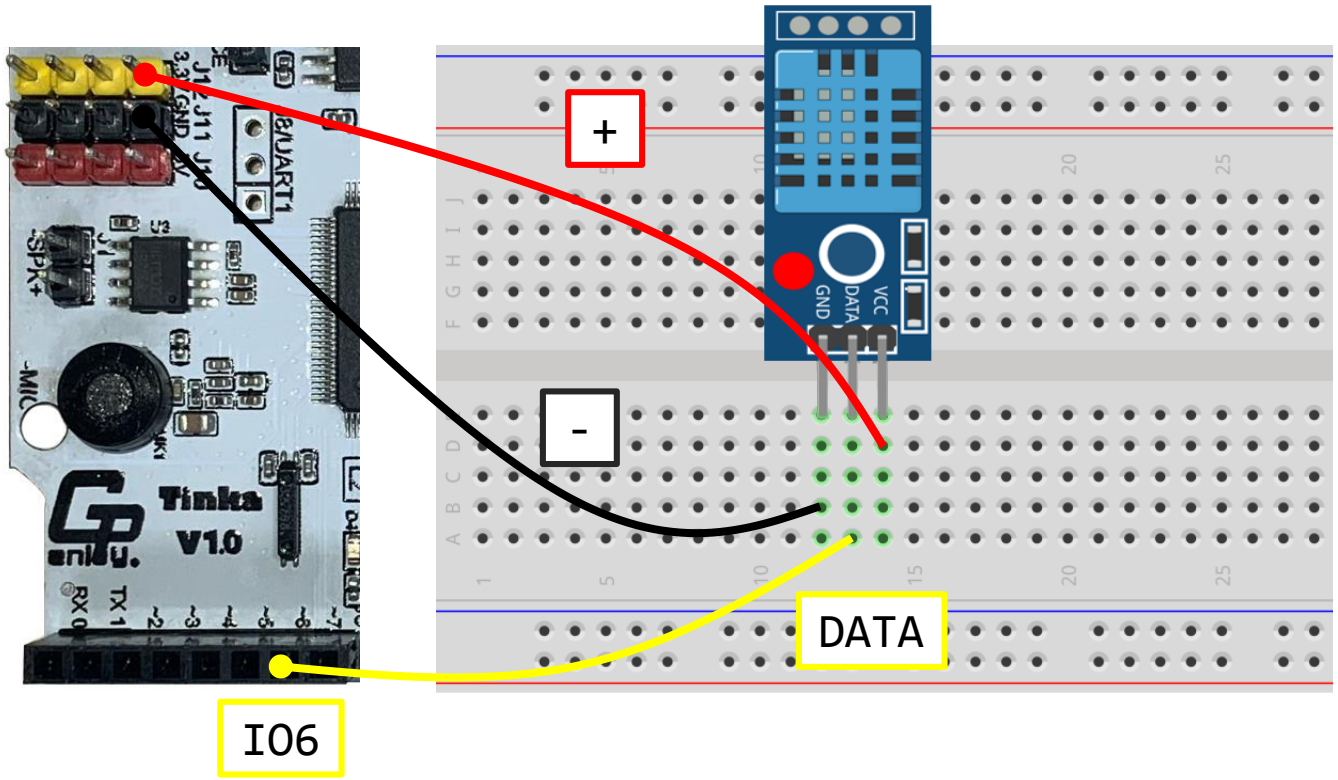
	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
Port	B11	B12	A8	A9	A10	A11	A12	A13
ALT	UART2 RX	UART2 TX	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7



圖片來源：旗標科技 超圖解 Python 物聯網

動手做

接線




```
>>> from sensor import DHT11
>>> dht = DHT11(6)
>>> dht.temperature()
26.8
>>> dht.humidity()
52.0
>>> dht.humidity()
58.0
>>> dht.humidity()
73.0
>>> dht.temperature()
27.5
```

動手做 寫程式監控溫濕度

可參考sample code_dht11.py

str(可以將其他種類的物件轉換成字串)

```
1  from sensor import DHT11                    # 匯入 DHT11 類別
2  from gpb import delay                       # 匯入 delay 函式
3
4  dht11 = DHT11(6)
5
6  while True:
7      print(str(dht11.temperature()) + "℃")  # 用數字建立字串物件再串接字串
8      print(str(dht11.humidity()) + "%")     # 用數字建立數串物件再串接字串
9      delay(2000)                            # DHT11 最快 2 秒可以回應一次
10
```

+ 在字串上是串接的意思

DHT11 因為傳輸上的基本限制
需間隔至少 2 秒才能再次詢問

```
互動環境(Shell) ×
26.3℃
56.0%
26.4℃
58.0%
26.4℃
60.0%
27.8℃
73.0%
27.7℃
74.0%
27.7℃
74.0%
27.7℃
```

2

PART

距離監測站

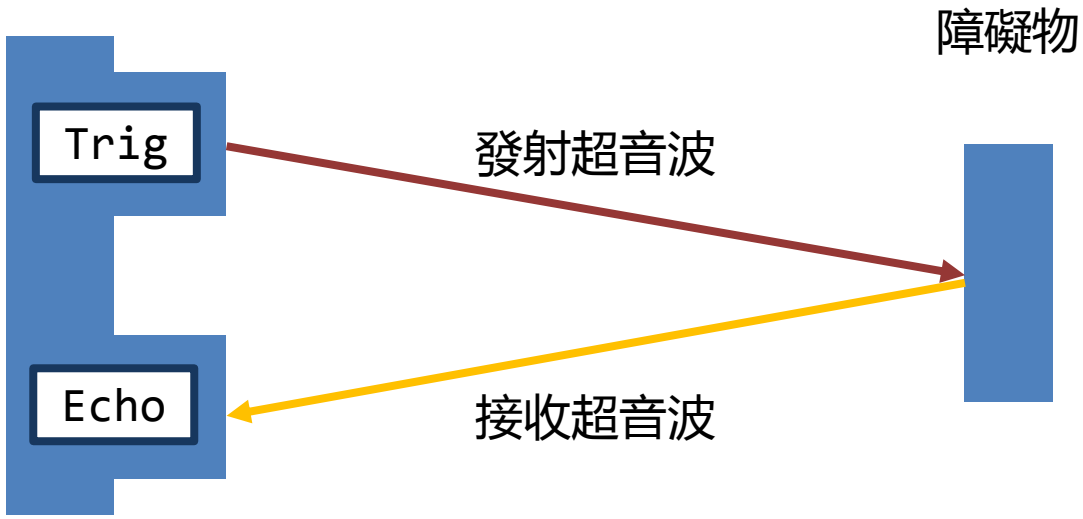
超音波模組

原理

超音波

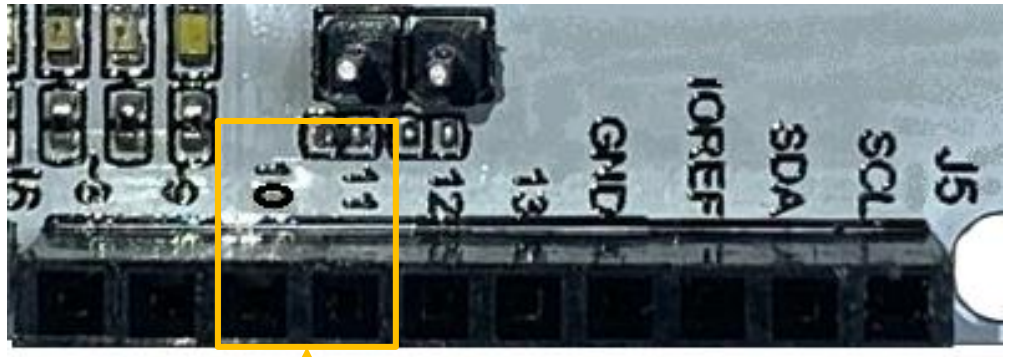
送出**高電位 10μs**
就會**發出超音波**

Trig送出超音波之後
Echo會先切到**高電位**
等**收到**超音波就
會變回**低電位**



可由超音波來回時間以及音速 340.29m/s 推算距離

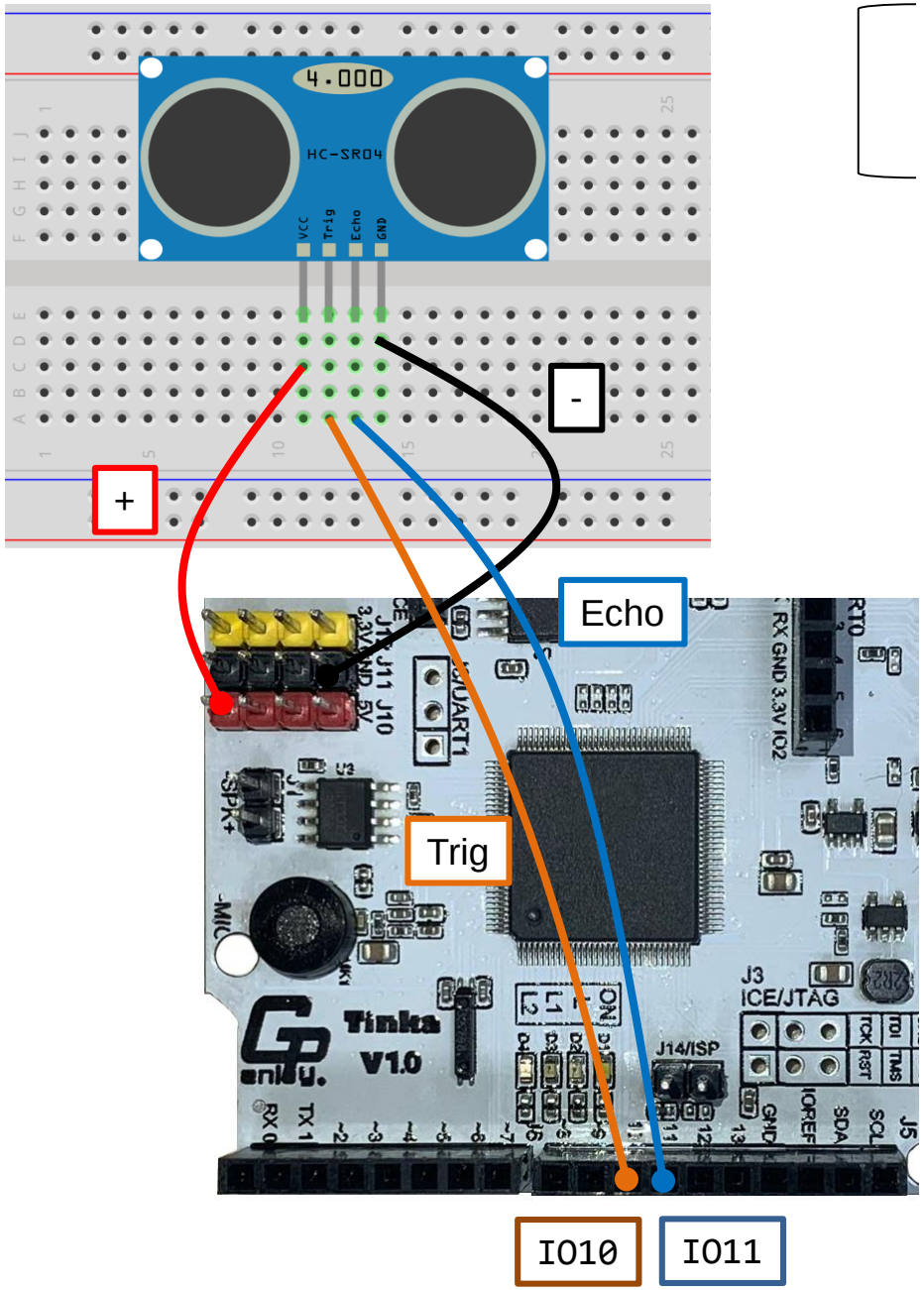
動手做 使用模組提供的函式



	IO8	IO9	IO10	IO11	IO12	IO13
Port	A14	A15	D13	D12	D11	D10
ALT	PWM8	PWM9	SPI CS	SPI CLK	SPI MOSI	SPI MISO

本實驗會使用 IO10(trig)、IO11(echo) 控制超音波測距感測器

接線



注意⚠ ： 每個模組的特性不同，
超音波模組需要用**5V**的正電才推得動唷!

```
>>> from sensor import HC_SR04
>>> sr04 = HC_SR04( 10 , 11 )
>>> sr04.Ultrasound()
16
>>> sr04.Ultrasound()
30
>>> sr04.Ultrasound()
45
```

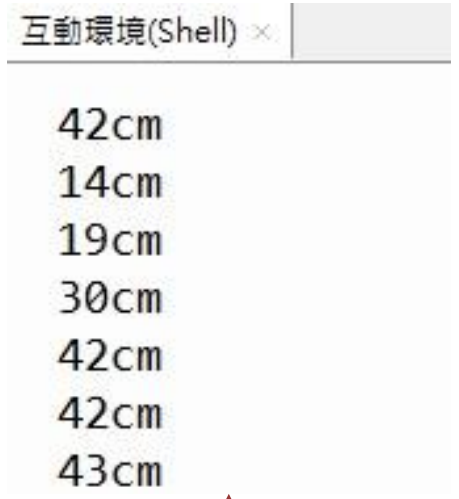
單位 : cm

動手做 寫程式監測距離

可參考sample code_sr04.py

```
1  from sensor import HC_SR04                  # 匯入HC_SR04類別
2  from gpb import delay                       # 匯入delay函式
3
4  sr04 = HC_SR04( 10 , 11 )                   # 建立腳位物件(Trig,Echo)
5  while True:
6      print(str(sr04.Ultrasound())+"cm")      # 先將數字轉為字串再串接
7      delay(1000)                             # 每秒監測一次
```

注意：先將讀取的
數字轉換成字串後
才做串接的動作唷。



注意⚠：
可測距離範圍：2cm ~ 450cm
超出範圍(太近或太遠) : 回傳 0

3

PART

伺服馬達

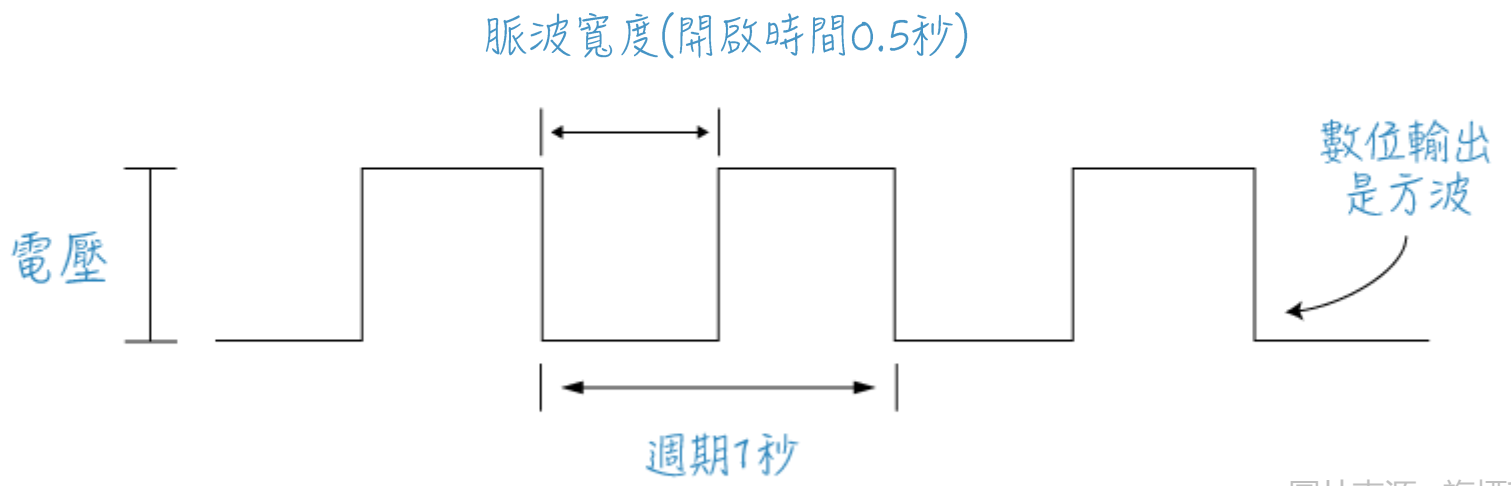
柵欄、機器手臂的根本

原理

PWM腳位

Pulse-width modulation

	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
Port	B11	B12	A8	A9	A10	A11	A12	A13
ALT	UART2 RX	UART2 TX	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7
ALT2	-	-	Servo(2)	Servo(3)	Servo(4)	Servo(5)	Servo(6)	Servo(7)

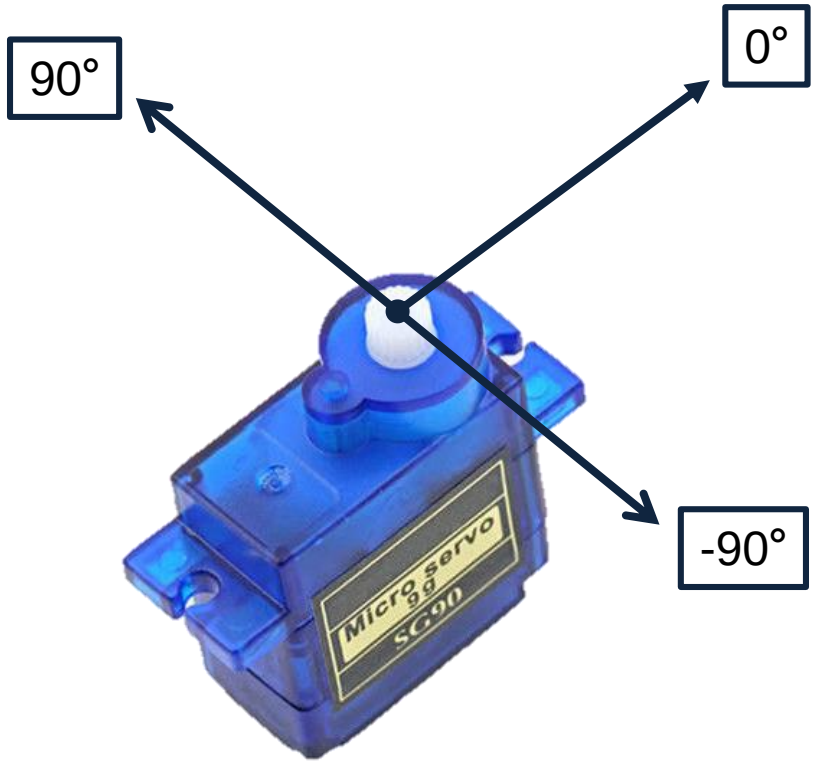


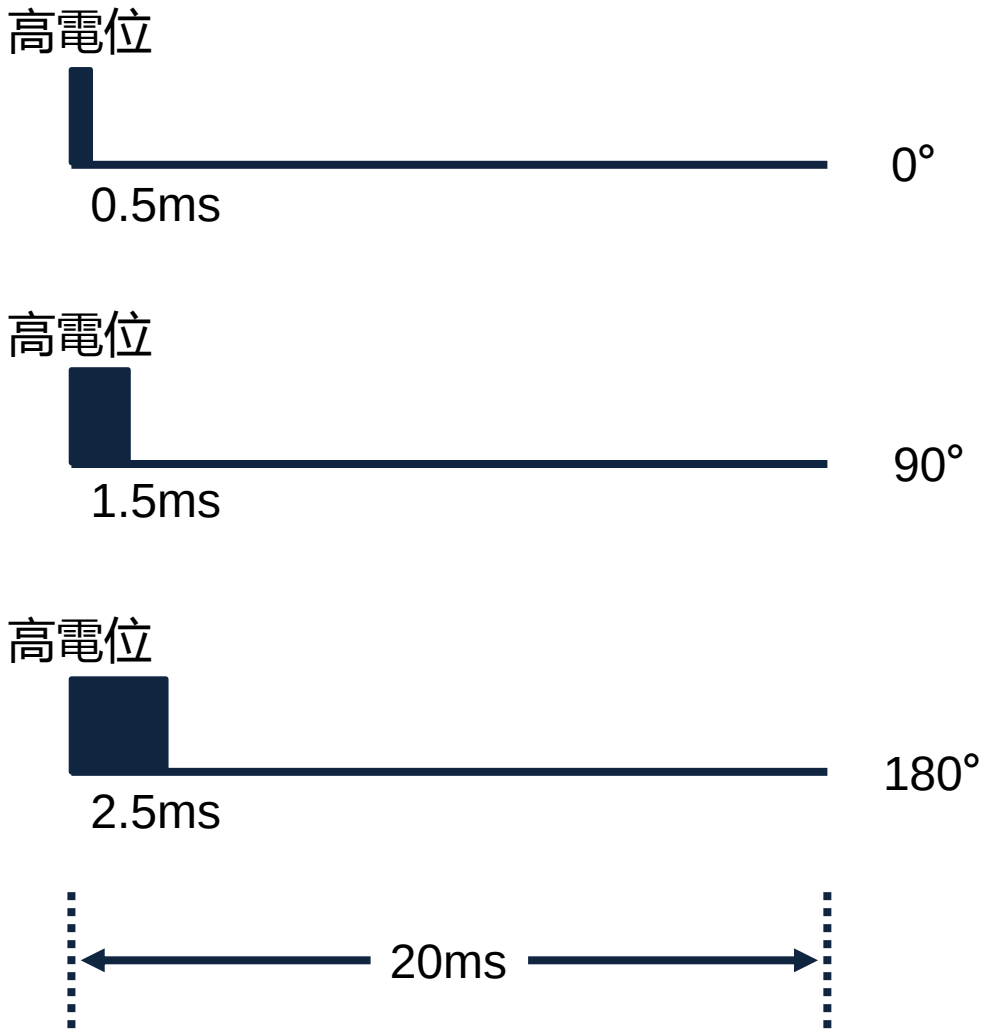
圖片來源：旗標科技 超圖解 Python 物聯網

原理

可控制角度的伺服馬達

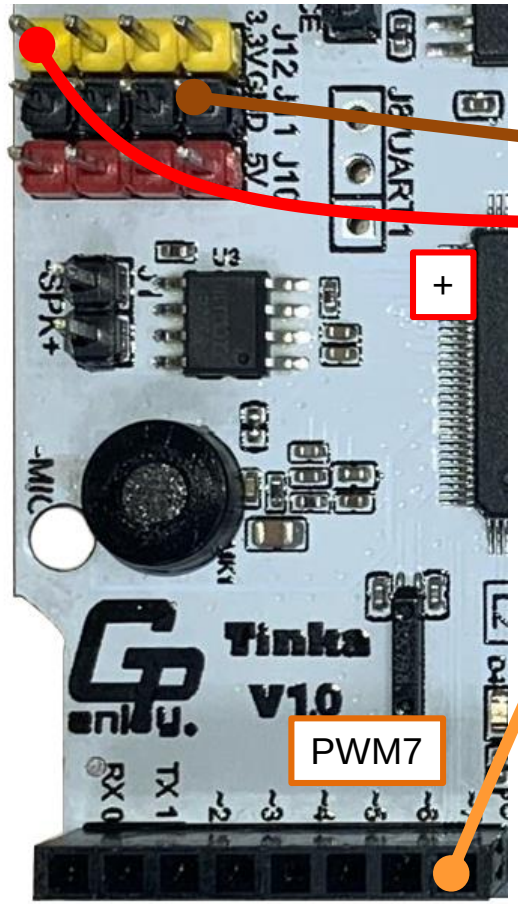
我們使用的是**90-180度伺服馬達**，
可以控制**角度**

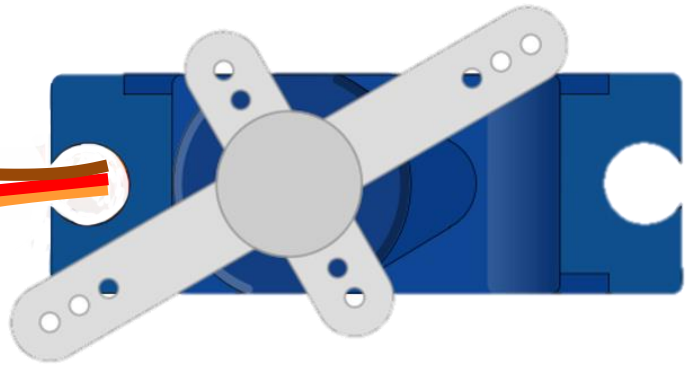




動手做

接線





	IO7
Port	A13
ALT	PWM7
ALT2	Servo(7)

```
>>> from gpb import Servo
>>> door = Servo(7)
>>> door.angle(90)
>>> door.angle(0)
>>> door.angle(-90)
>>> door.angle(  )
```

可自行填入(範圍內)**角度**

動手做

寫程式轉動柵欄

可參考sample code_servo.py

```
from gpb import Servo      # 匯入 Servo 類別
from gpb import delay      # 匯入 delay 函式

servo1 = Servo(1)          # 使用 PWM8 腳位操控 Servo1

while True:                # True 代表成立
    servo1.angle(0)        # 停止
    delay(500)             # 等 0.5 秒
    servo1.angle(-50)      # 順時針轉 50度
    delay(500)
    servo1.angle(0)
    delay(500)
    servo1.angle(50)       # 逆時針轉 50度
    delay(500)
```

動手做

借光感應模組轉動柵欄

可參考sample code_light_servo.py

```
1  from machine import Pin,ADC          # 匯入Pin,ADC類別
2  from gpb import LED, Servo, delay    # 匯入 delay 函式
3
4  door = Servo(7)                      # 使用 PWM7 腳位操控 Servo7
5  light = ADC(5)                       # 建立讀取光線變化的物件
6
7  door.angle(0)
8
9  while True:
10     if                               # 偵測亮度值
11                                      # 有人(變暗)開門
12     else:
13                                      # 沒人(變亮)關門
14     print(light.read())
15     delay(500)                       # 等05秒
```

THANKS

Q&A 時間