How to use

Ardoino to do Face Recognition

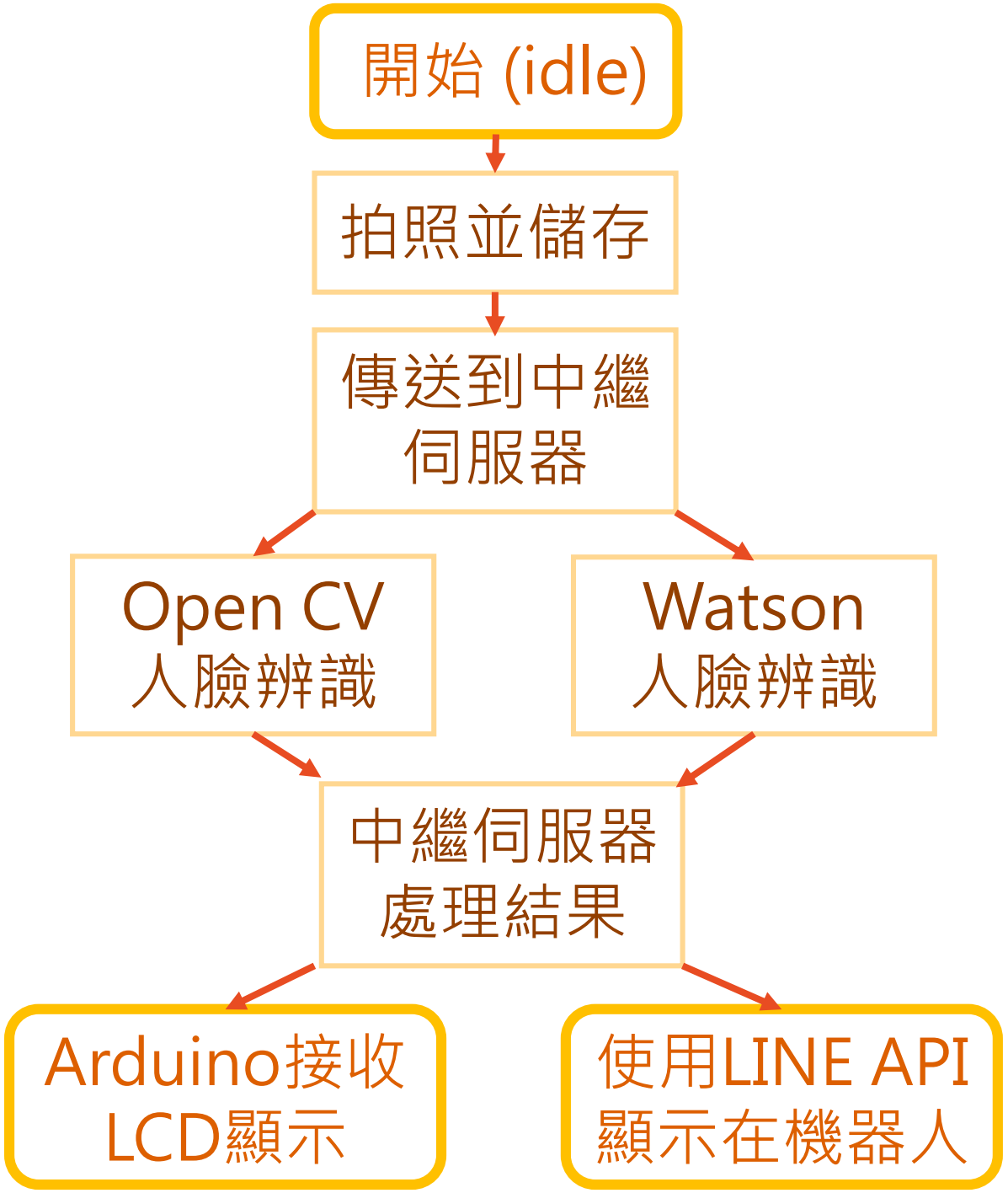
CSI 策略創新中心

北祥股份有限公司

Michael

VR人臉偵測

全部流程


開始 (idle)
拍照並儲存
傳送到中繼伺服器
Open CV
人臉辨識
Watson
人臉辨識
中繼伺服器
處理結果
Arduino接收
LCD顯示
使用LINE API
顯示在機器人

What is Arduino

- Arduino專案始於2003年，作伊夫雷亞互動設計研究所的學生專案，目的是為新手和專業人員提供一種低成本且簡單的方法，以建立使用傳感器與環境相互作用的裝置執行器。

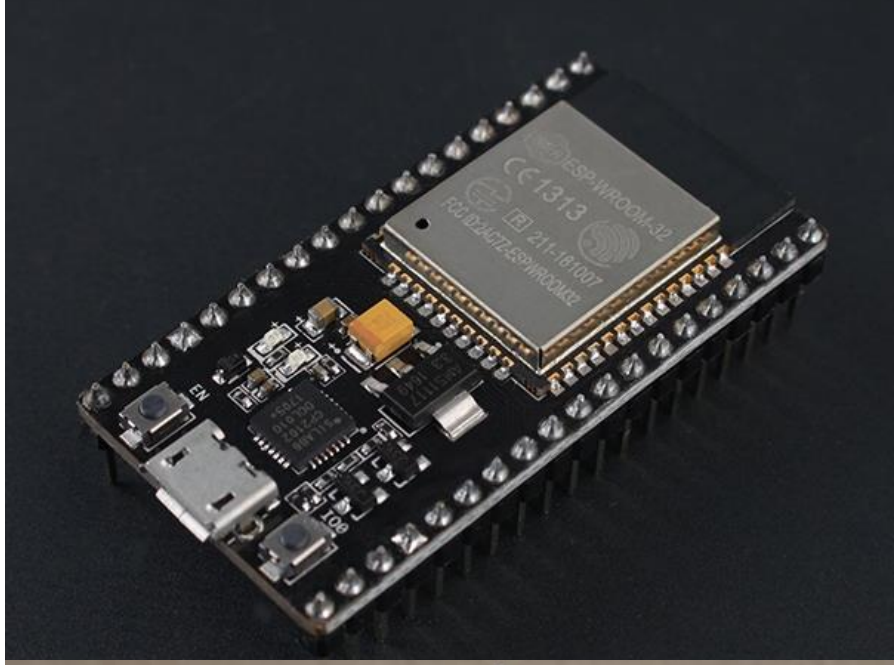
- Arduino這個名字來自義大利伊夫雷亞的一家酒吧，該專案的一些創始人過去常常會去這家酒吧。 酒吧以伊夫雷亞的Arduin命名，他是伊夫雷亞邊疆伯爵，也是1002年至1014年期間的義大利國王。

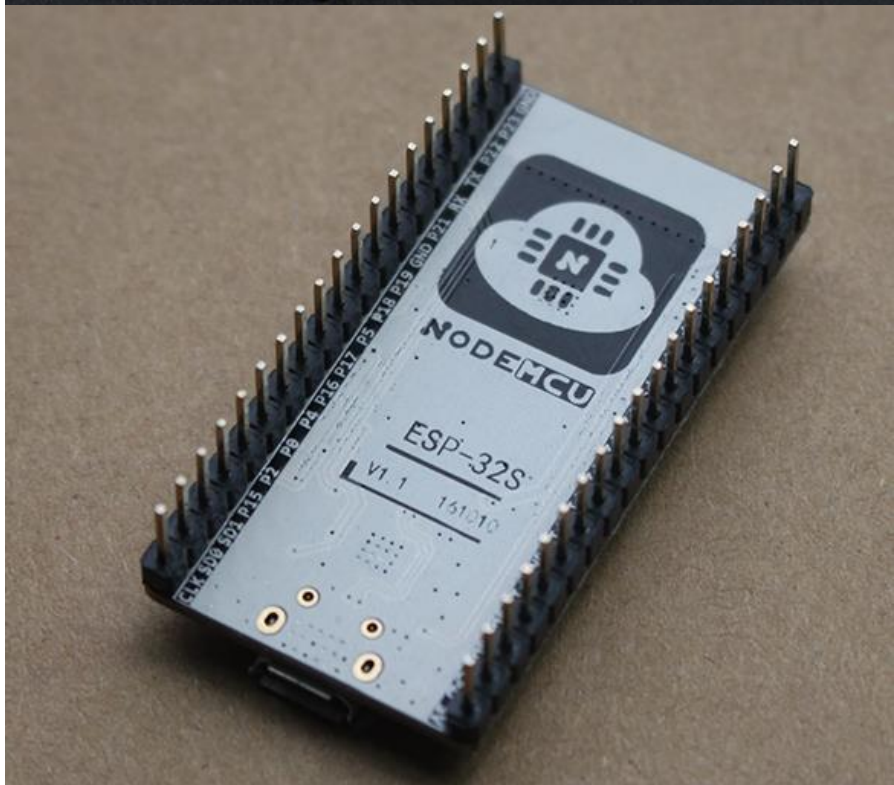
What we need

- 使用設備: (ESP32)NodeMCU-32S、麵包板、感應元件
- 開發環境: Arduino Software IDE
- 使用語言: C/C++

NodeMCU-32S

- WiFi
- 藍牙4.2
- 整合多項模組
- 接USB可用

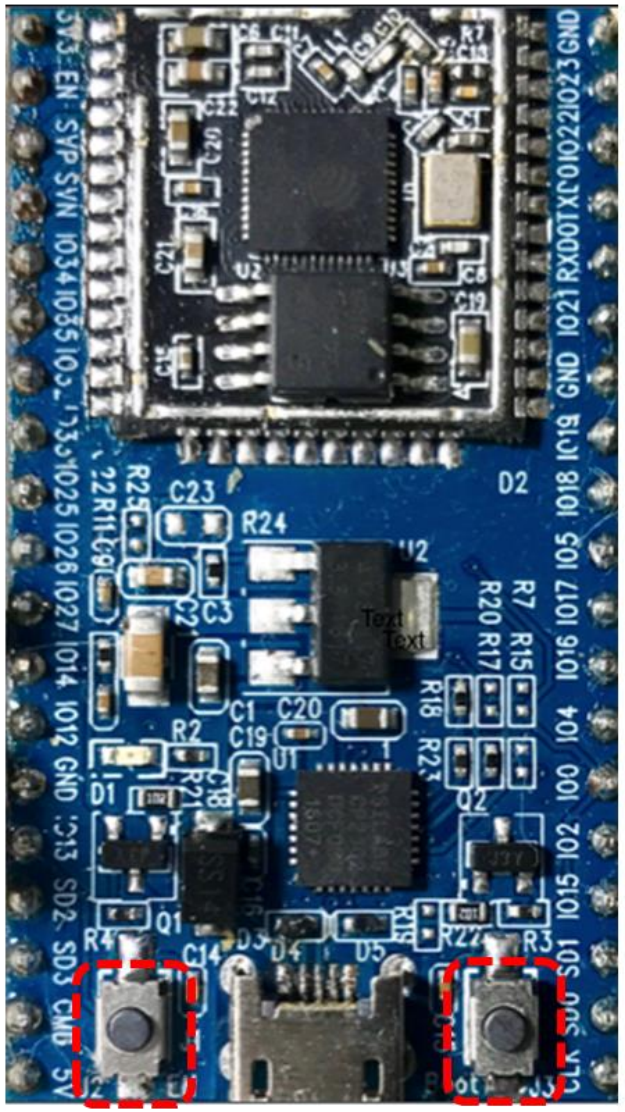




(ESP32)NodeMCU-32S

						左											
					3.3V	19	19	GND									
(pu)				RESET	EN	18	18	GPIO23	VSPI MOSI								SPI MOSI
SVP			ADC0		GPIO36	17	17	GPIO22									Wire SCL
SVN			ADC3		GPIO39	16	16	GPIO1	TX0								Serial TX
			ADC6		GPIO34	15	15	GPIO3	RX0								Serial RX
			ADC7		GPIO35	14	14	GPIO21									Wire SDA
		TOUCH9	ADC4		GPIO32	13	13	GND									
		TOUCH8	ADC5		GPIO33	12	12	GPIO19	VSPI MISO								SPI MISO
DAC1			ADC18		GPIO25	11	11	GPIO18	VSPI SCK								SPI SCK
DAC2			ADC19		GPIO26	10	10	GPIO5	VSPI SS							(pu)	SPI SS
		TOUCH7	ADC17		GPIO27	9	9	GPIO17									
	TMS	TOUCH6	ADC16	HSPI SCK	GPIO14	8	8	GPIO16									
(pd)	TDI	TOUCH5	ADC15	HSPI MISO	GPIO12	7	7	GPIO4		ADC10	TOUCH0				(pd)		
					GND	6	6	GPIO0	BOOT	ADC11	TOUCH1				(pu)		
	TCK	TOUCH4	ADC14	HSPI MOSI	GPIO13	5	5	GPIO2		ADC12	TOUCH2				(pd)		
				FLASH D2	GPIO9	4	4	GPIO15	HSPI SS	ADC13	TOUCH3	TDO			(pu)		
				FLASH D3	GPIO10	3	3	GPIO8	FLASH D1								
				FLASH CMD	GPIO11	2	2	GPIO7	FLASH D0								
					5V	1	1	GPIO6	FLASH SCK								

右



Reset Flash

麵包板上的位置

Arduino 實際IO腳位

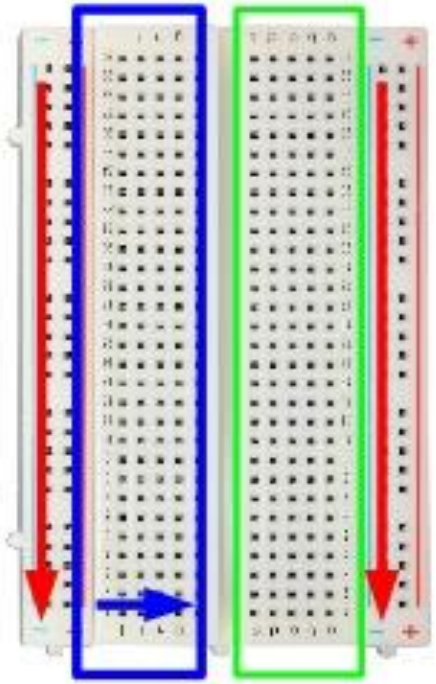
腳位對應

- 18 x Analog-to-Digital：可接類比或數位感測(粉紅色標示)
- 3 x SPI interfaces(MOSI,MISO,SCK,SS)： SD card, RFID(藍色標示)
- 3 x UART interfaces：藍芽、相機(不限腳位編號)
- 2 x I2C interfaces(SDA,SCL) and I2S interfaces：LCD、氣壓、陀螺儀(GPIO 21,22)
- 16 x PWM output channels：數位輸出也可類比輸出(不限定腳位編號)
- 2 x Digital-to-Analog Converters (DAC)：不太需要用
- 10 x Capacitive sensing GPIOs：提供觸控電容(棕色標示)
- GPIO 34,35,36,39：Input only
- GPIO 1,2,3,6,7,8,9,10,11：系統用，勿使用(淺灰色標示)
- GPIO 18：重開機

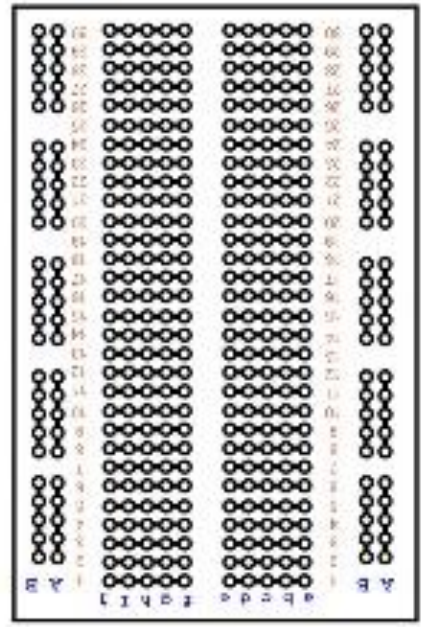
麵包板

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



麵包板的內部結構



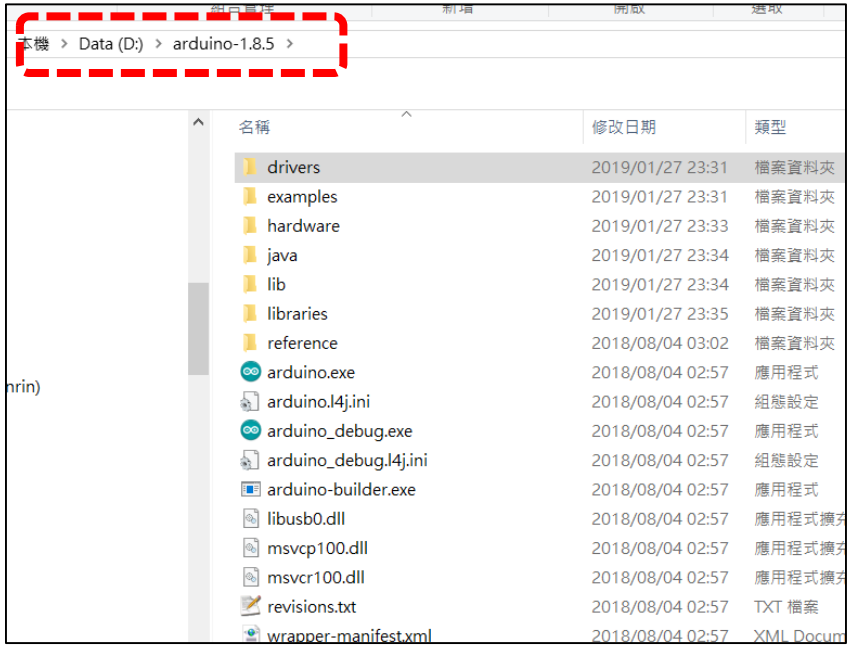
http://bugworkshop.blogspot.tw/2012/12/diy-breadboard.html

一、環境設定

1. 下載arduino-1.8.5.zip
   並設定Arduino IDE for NodeMcu-32s
2. 簡單測試程式：HelloWorld
3. ESP32函式庫安裝

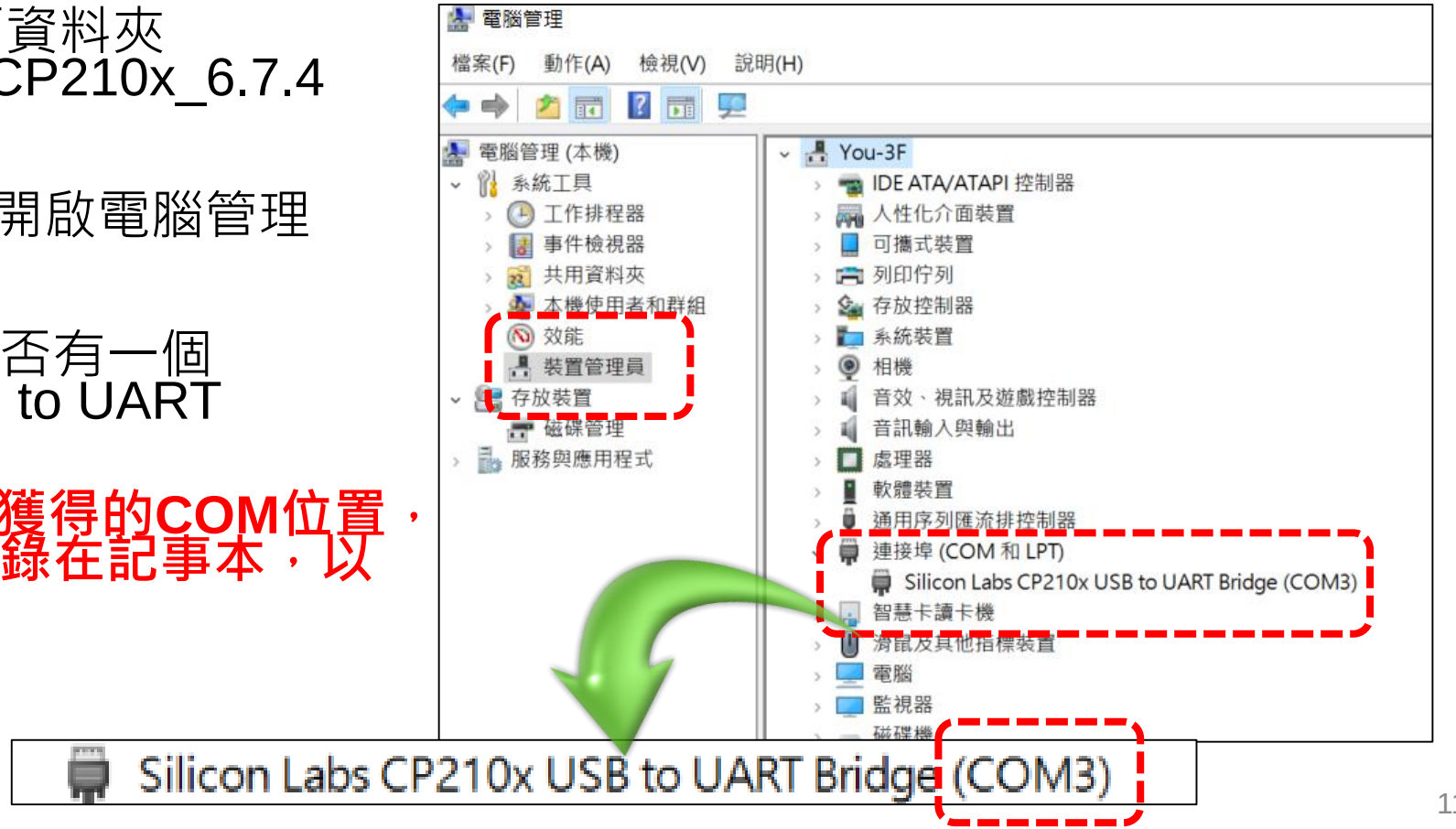
解壓縮到指定的資料夾

- arduino-1.8.5.zip
- 本版本為免安裝版，解壓縮後即可執行
- 建議解壓縮到D硬碟中，也可放到USB中
- 例如本例，解壓縮到d:\arduino1.8.5\



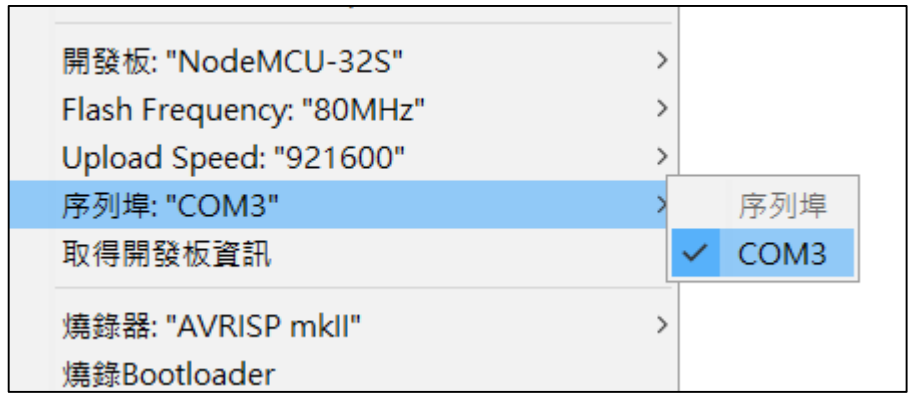
安裝驅動

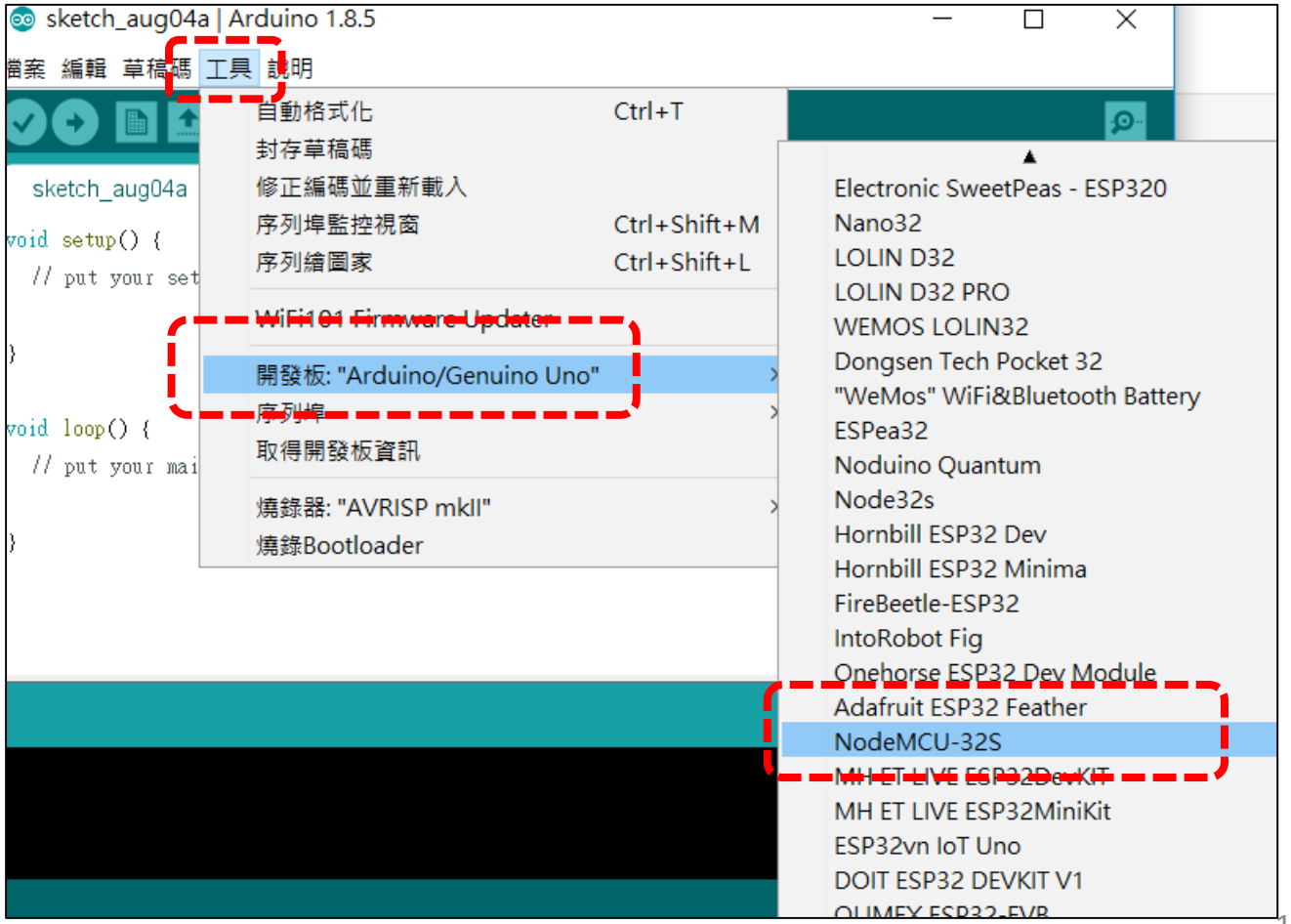
1. 解壓縮後後，將NodeMcu-32s插入USB
2. 此時會出現安裝畫面，但由於沒有驅動程式，因此自行指定驅動程式
3. ESP32驅動程式位於以下資料夾 D:\arduino-1.8.5\drivers\CP210x_6.7.4
4. 安裝驅動完畢後
5. 在我的電腦/右鍵/管理/，開啟電腦管理
6. 點選左側裝置管理員
7. 點選右側連接埠，檢查是否有一個 Silicon Lab CP210x USB to UART (COMX)
8. **請注意最後的ComX代表獲得的COM位置，每次可能都會不同，請紀錄在記事本，以後會用到。**



設定Arduino開發界面

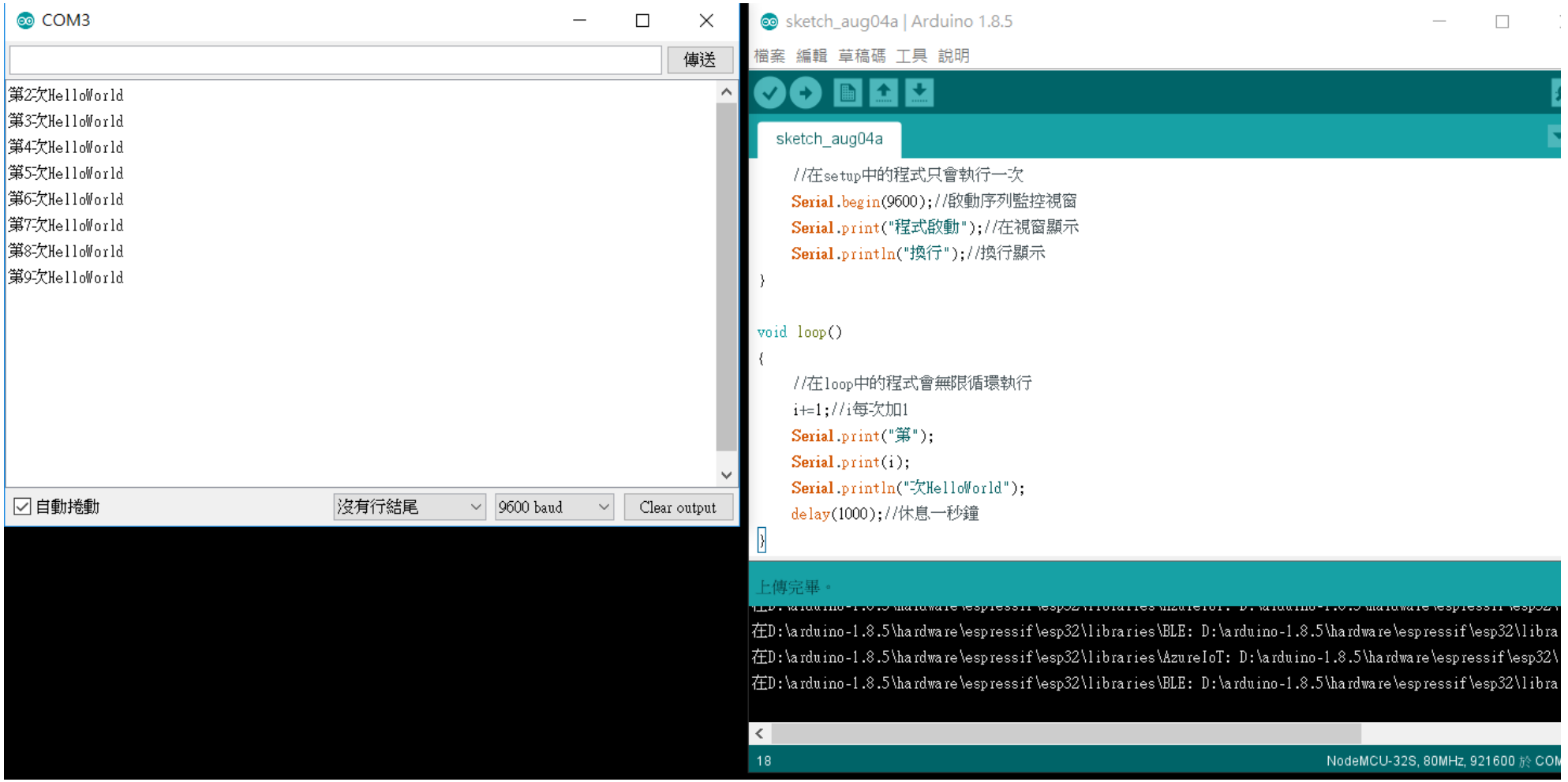
- 開啟arduino-1.8.5下的Arduino主程式：arduino.exe
- 點選功能表/工具/開發版並往下拉，直到選擇NodeMCU-32s
- 更改序列埠為COMX，依照**投影片頁碼6的第6點選擇**
- UploadSpeed選擇：921600





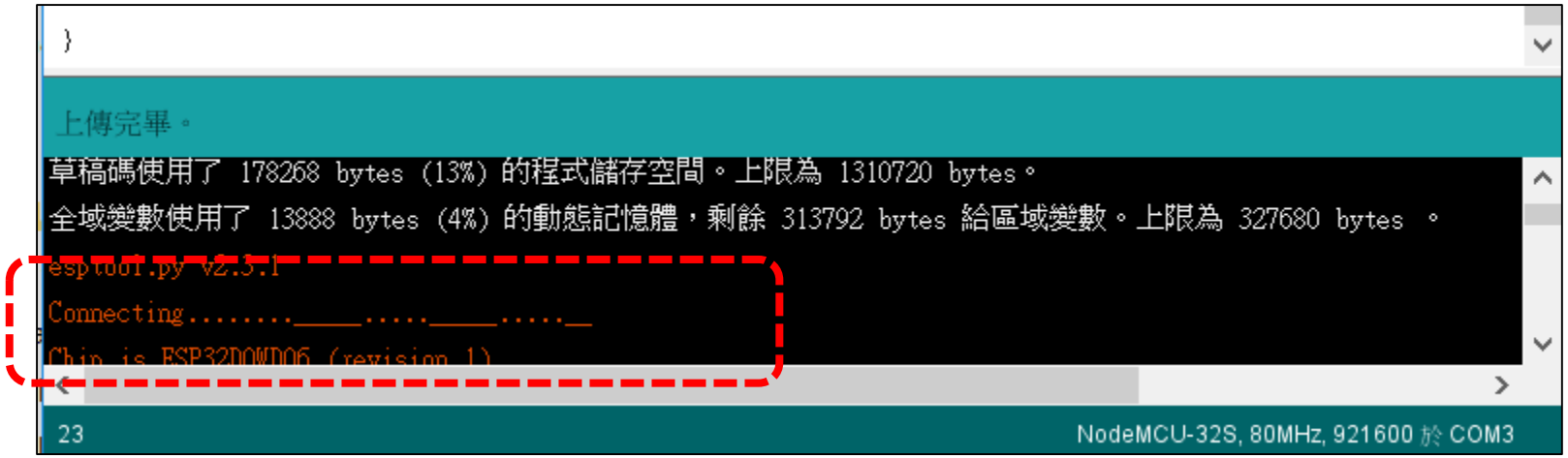
簡單測試程式：HelloWorld

- 1.HelloWorld.txt



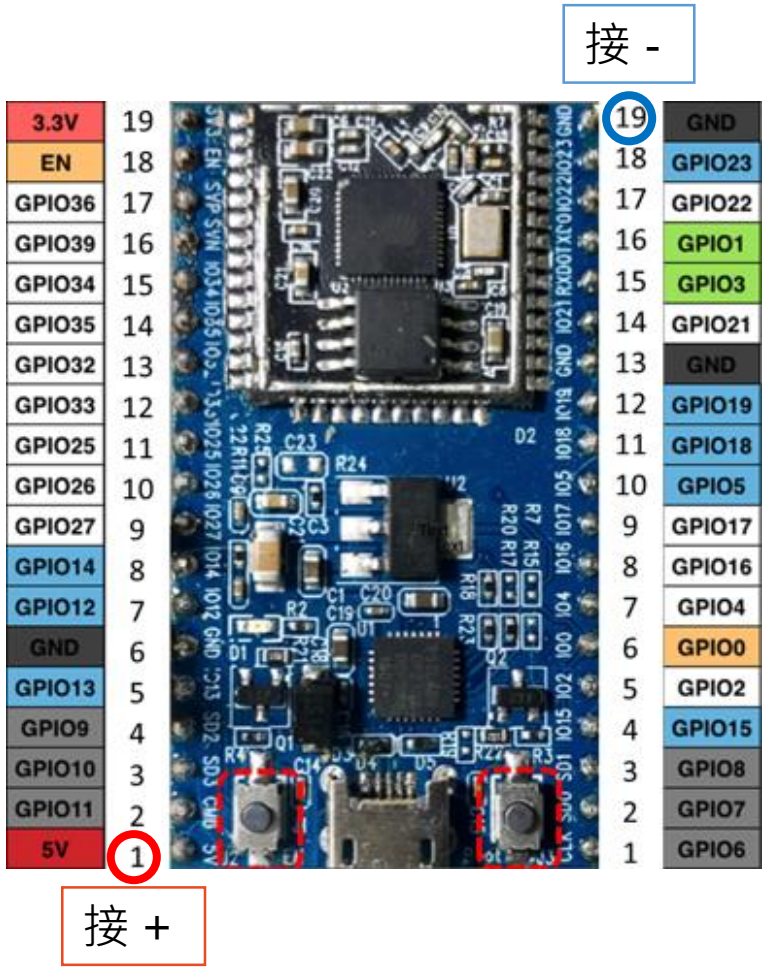
注意事項

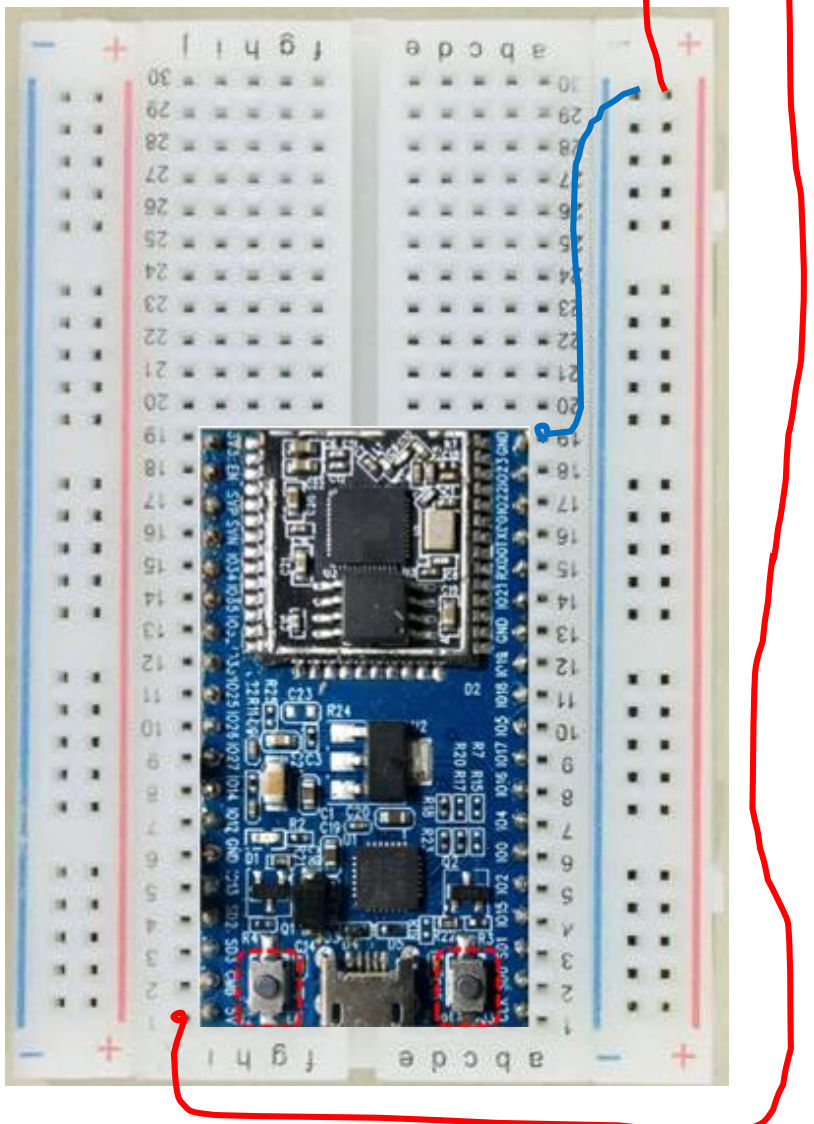
- Arduino IDE 上傳程式碼，出現……_____.....____時，須按住右側Flash鍵
- 後續程式撰寫與一般Arudino幾乎無異
- 若需要函式庫，請Google
  關鍵字：ESP32+所需功能



Get Start

- 一般我們會將正電5V及GND接至兩側
- 所有元件要通電才會有訊號，任何元件都要接上高低電位





Arduino
程式結構

Setup & loop

Void setup() {

設定的部份

僅一開始執行一次

}

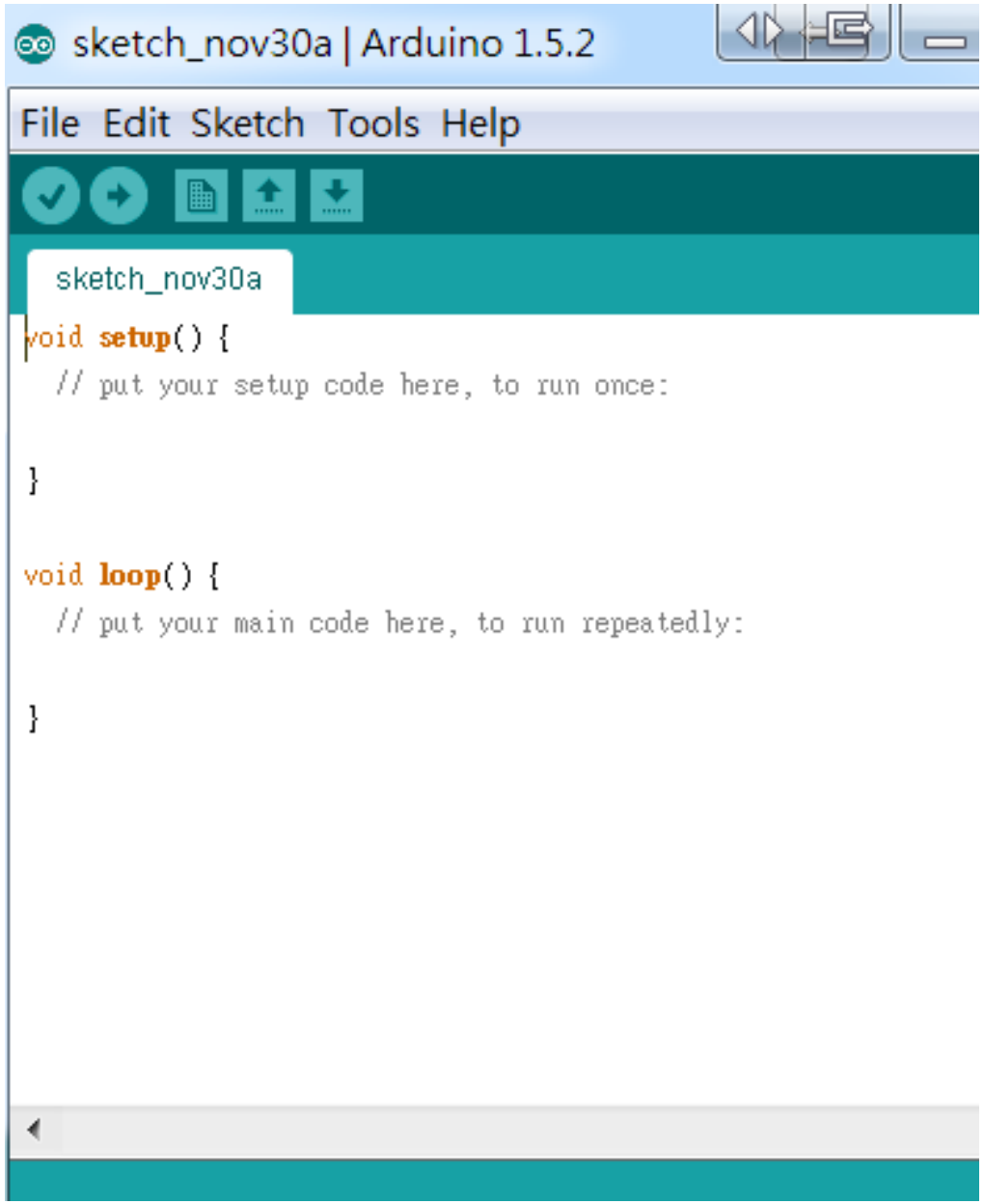
Void loop() {

工作內容

Setup內容完成之後

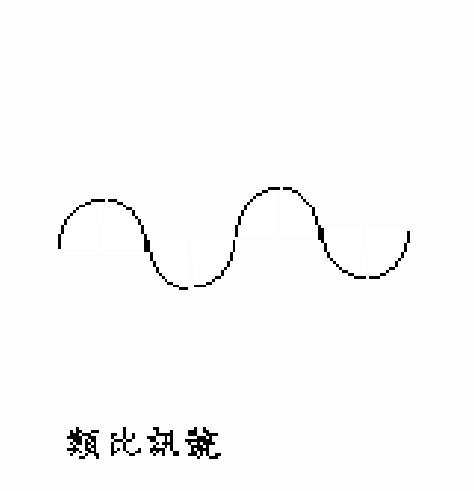
不斷重覆運作

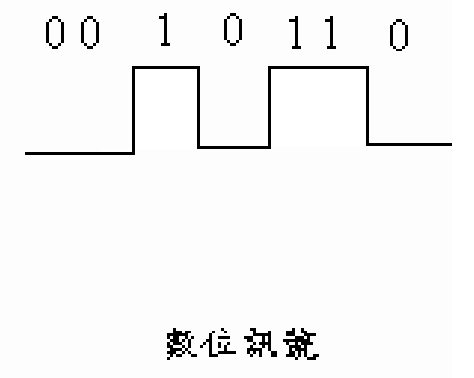
}



常用語法

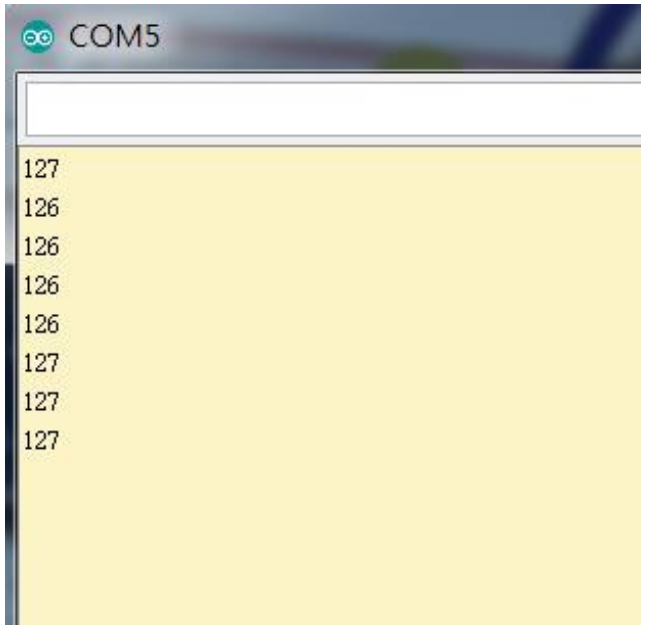
- 數位感測輸入（人體、觸摸）
  - pinMode(腳位編號,INPUT);　　//宣告為輸入
  - Statue=digitalRead(腳位編號);　　//讀取的值（1或0）
- 數位輸出（LED、繼電器）
  - pinMode(腳位編號, OUTPUT);　　//宣告為輸出
  - digitalWrite(腳位編號,HIGH);　　//輸出5V
  - digitalWrite(腳位編號,LOW);　　//輸出0V
- 數位模擬類比PWM輸出（RGB LED、DC馬達）
  - 須宣告PWM channels=0~15, FREQ=5000, resolution = 8
  - ledcAttachPin(GPIO, channel)
  - ledcWrite(channel, dutycycle)





常用語法

- 類比輸入(火焰、水位、光敏、振動)
  sensorValue = analogRead(腳位編號);　//讀取的數值
- 序列監視視窗可以觀察類比輸入的數值
  - 在setup()加入此行
    - Serial.begin(9600);　//宣告讀取數值及速率
  - 在loop()加入以下
    - delay(500);
    - int sensorValue = analogRead(腳位); //讀取數值
    - Serial.println(sensorValue);　//輸出數值



First Program

- 讓LED隔一秒閃亮

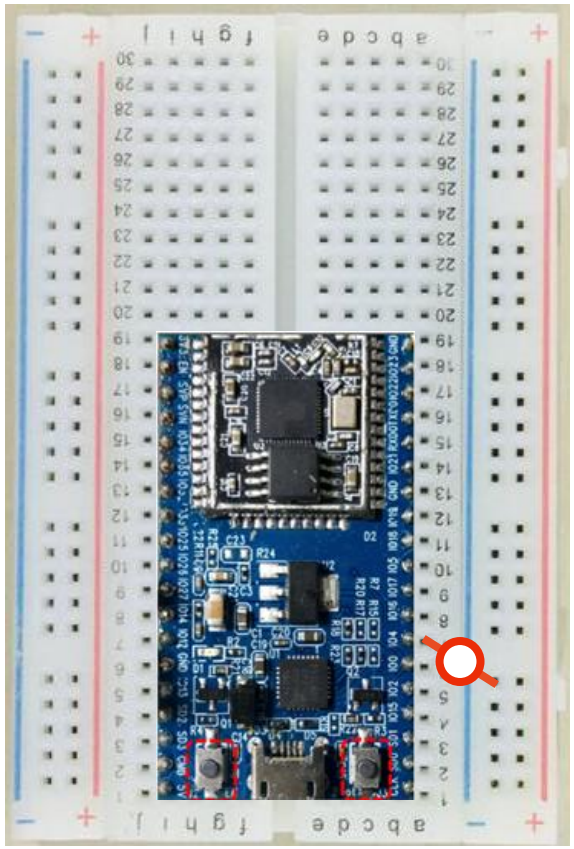
```
void setup()
{
    pinMode(4, OUTPUT);     // 宣告4為輸出
}
```

```
void loop()
{
    digitalWrite(4,HIGH);       // turn the LED on
    delay(1000);                // wait for a second
    digitalWrite(4,LOW);        // turn the LED off
    delay(1000);                // wait for a second
}
```

開→停一秒→
關→停一秒→ …

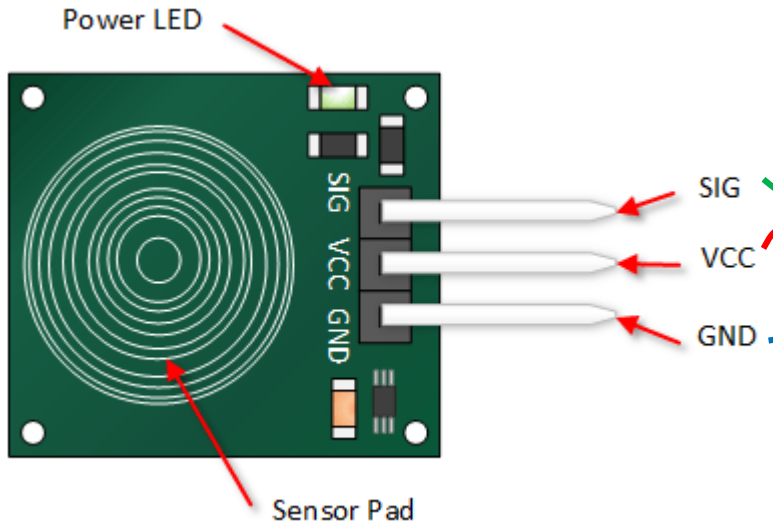
參考P5配置圖
4為輸出 → 代表使用GPIO4

LED燈兩腳長度不同，長接正短接負，接錯可能會燒掉



Adruino
元件接法

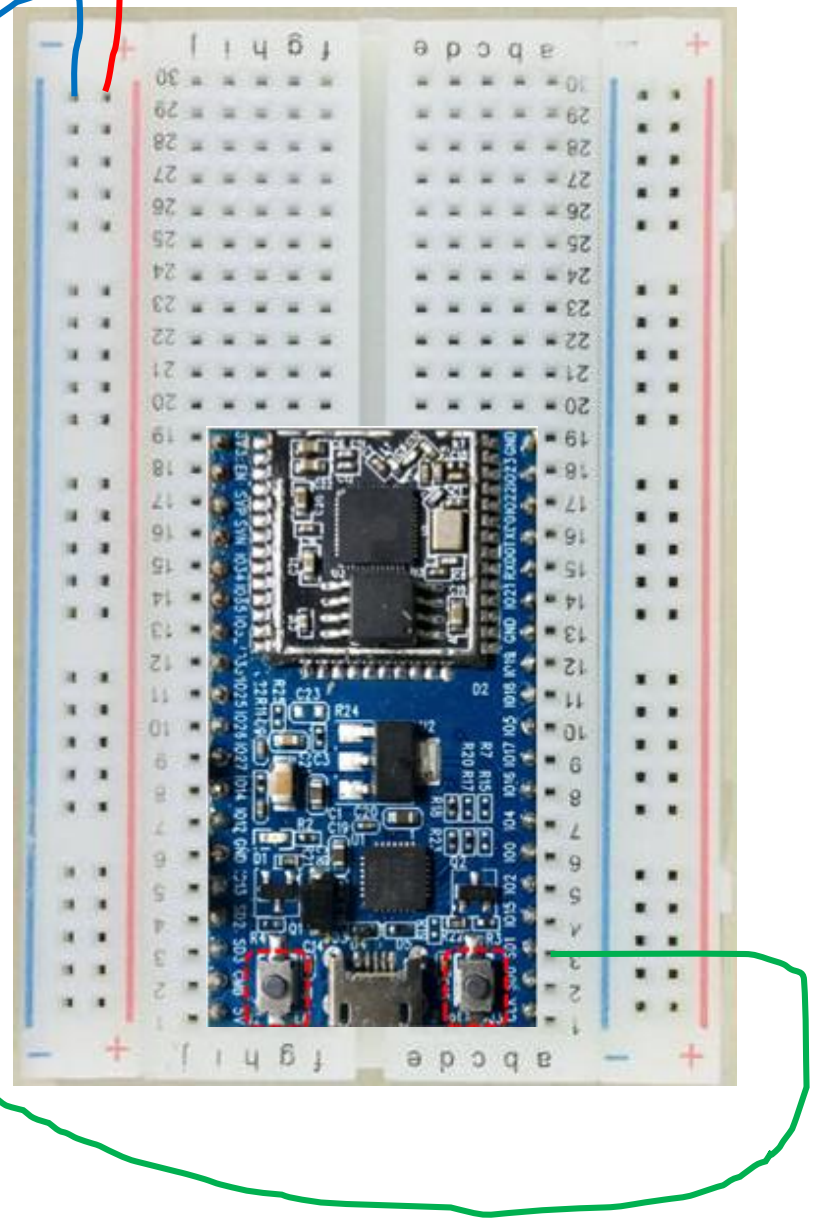
觸控元件



VCC 接高電位

GND 接低電位

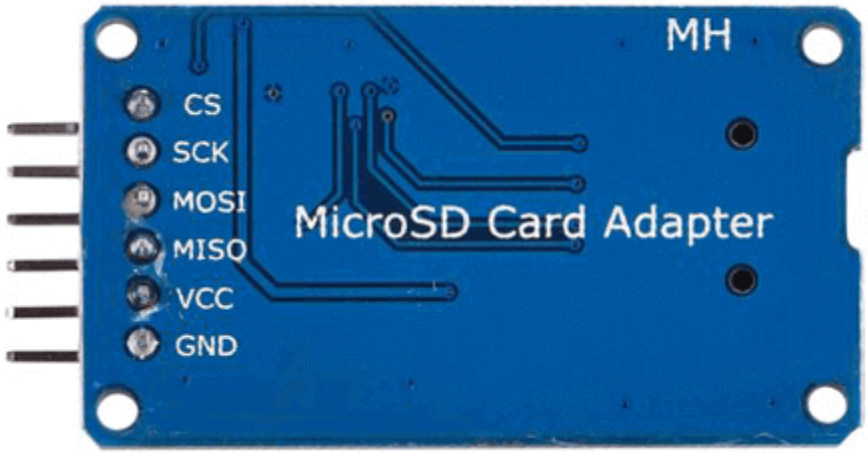
SIG 接 GPIO36



Adruino
元件接法

SD卡儲存空間

CS GPIO5
SCK GPIO18
MOSI GPIO23
MISO GPIO19
VCC 接高電位
GND 接低電位



Adruino
元件接法

LCD 顯示器

GND 接低電位

VCC 接高電位

SDA GPIO21

SCL GPIO22

Adruino
元件接法

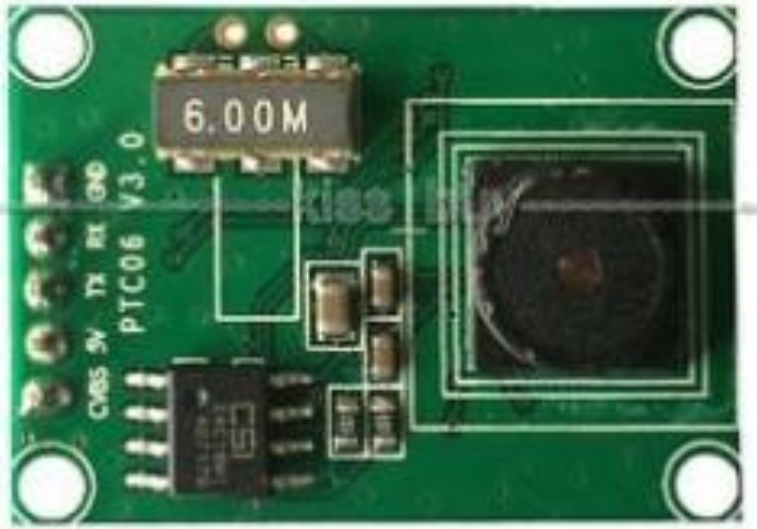
相機

GND 接低電位

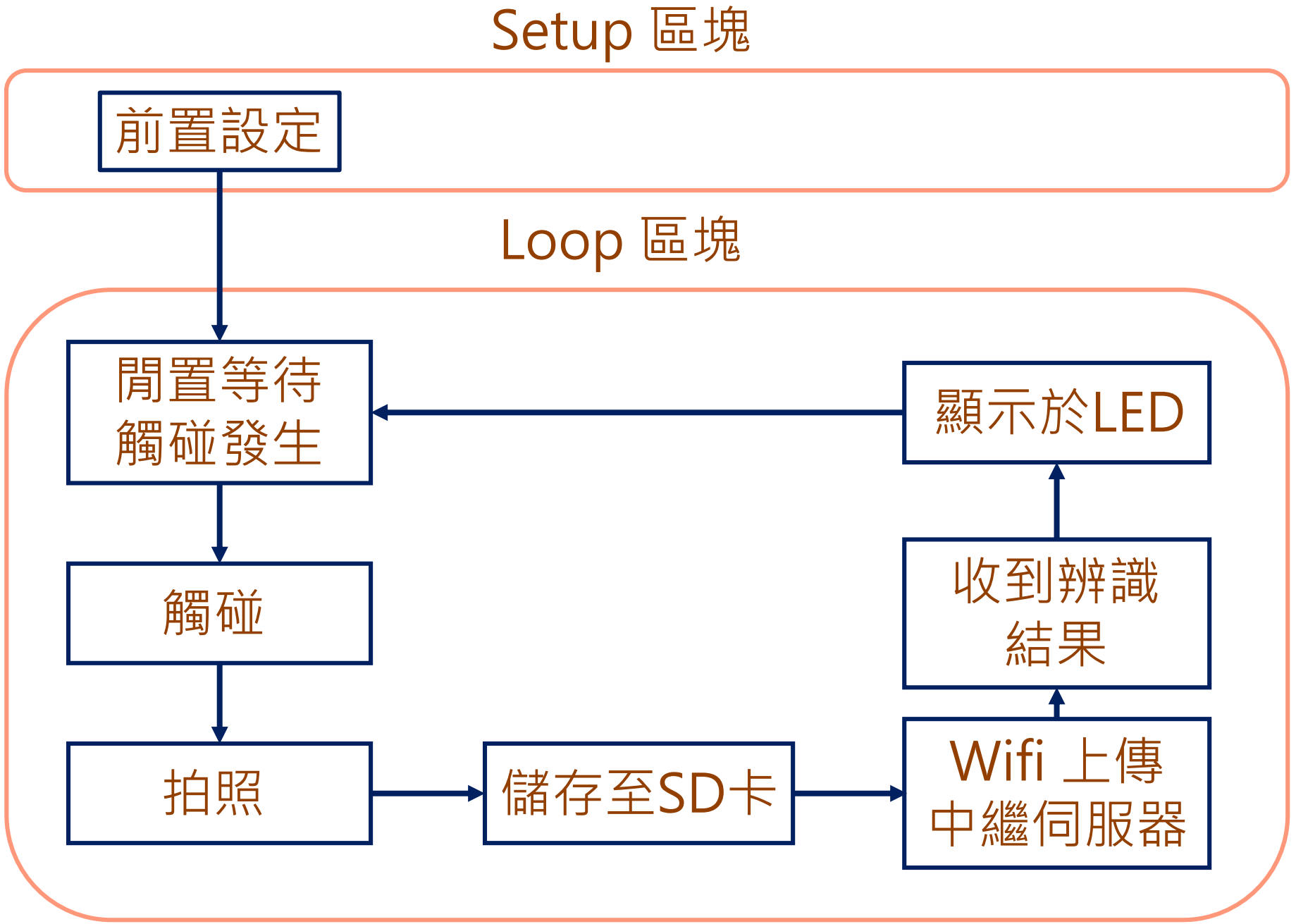
RX GPIO12

TX GPIO13

5V 接高電位



Adruino
程式部份



Arduino Program

Part 1

- // 使用到函式庫
- #include <SD.h>
- #include <HardwareSerial.h>
- #include <WiFi.h>
- #include <WiFiMulti.h>

- ////////////////
- // 變數設定
- // 命令宣告
- ////////////////

Arduino Program

Part 2

```
// 初始化所有用到元件
void setup() { ... }

// 重複動作
void loop()
{
    // 網路連線成功
    if((WiFiMulti.run() == WL_CONNECTED))
    {
        // 有觸碰
        if (digitalRead(touchpin))
        {
            // 開始拍照並上傳
        }
    }
}
```

Arduino Program

Part 3

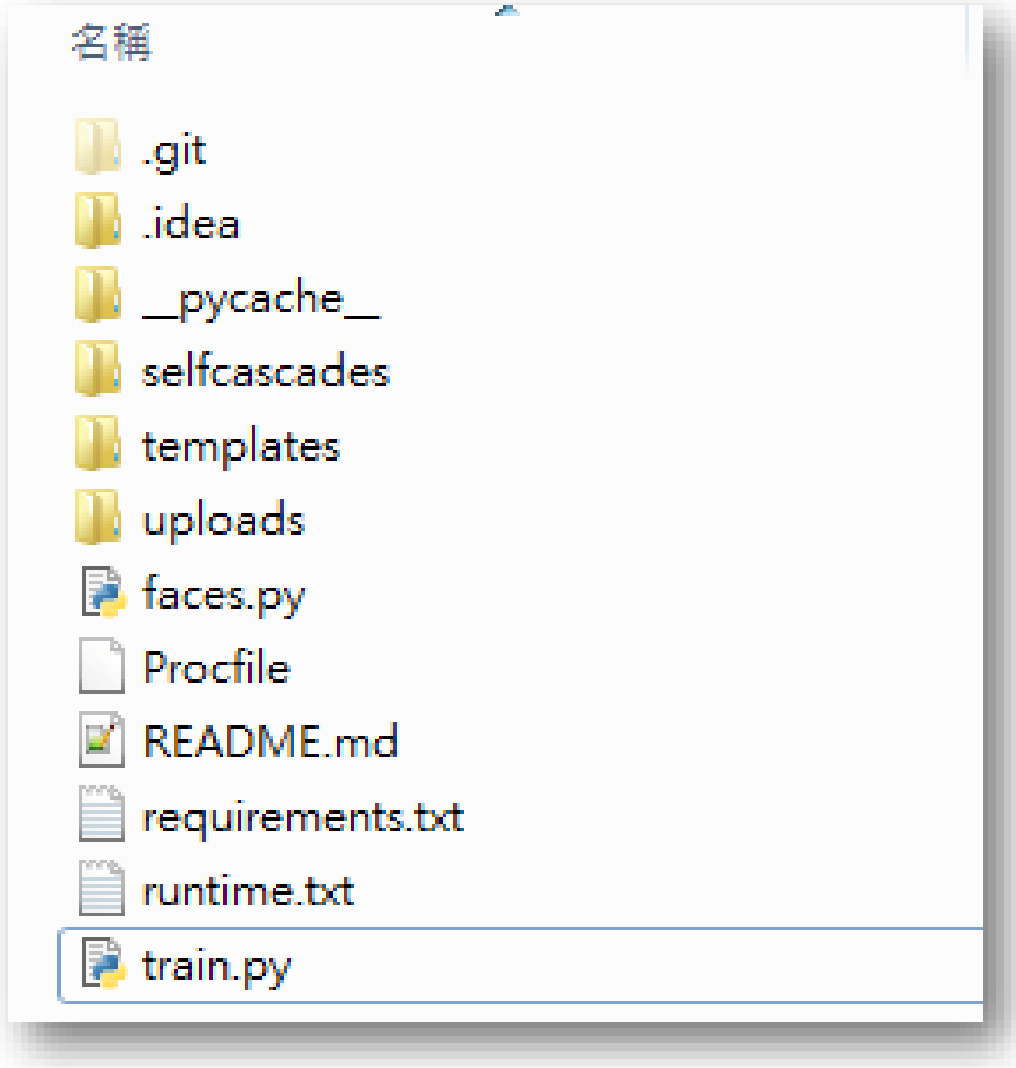
```
// 透過wifi上傳圖檔
void wifisendfile(String filename){ ... }

// 以下相機命令

// 重設相機
void SendResetCmd() { ... }

// 設定拍照圖片size
void SetImageSizeCmd(String size) { ... }

// 設置波特率 似乎沒有用
void SetBaudRateCmd() { ... }

// 拍照指令
void SendTakePhotoCmd() { ... }

// 讀圖檔指令
void SendReadDataCmd() { ... }

// 停止拍照指令
void StopTakePhotoCmd() { ... }
```

中繼伺服器

face.py

train.py

- face.py：利用先前上課做的模型做識別
- Train.py：中繼伺服器，用來接收Arduino上傳檔案，呼叫face.py或者Watson做辨識，再回傳到Arduino或送到linebot做顯示



中繼伺服器

Python server

- 下載 iot-line.zip 解壓縮到 D:/，或你的自訂目錄下
- 打開CMD，移動到目錄下 (cd PATH)
- 打開虛擬環境 (activate deepface)
- 安裝所需套件 (pip install -r requirements.txt)
- 將你的模型檔 predict_face.xml 放到 /selfcascades
- 打開 jupyter (jupyter notebook)
- 打開 Server.ipynb 做一些修改
- 執行 Server.ipynb
  - 跳出 `Running on http://0.0.0.0:80/ (Press CTRL+C to quit)` ，代表成功，在網址列輸入localhost可以打開測試網頁

中繼伺服器

Server.ipynb

```
# -*- coding: utf-8 -*-
#__author__ = 'Robin Kuo'
from flask import Flask, flash, request, redirect, render_template ,make_response , jsonify
from werkzeug.utils import secure_filename
import json , os
import requests
import faces
from ibm_watson import VisualRecognitionV3

#檔案上傳路徑
UPLOAD_FOLDER = './uploads/'
ALLOWED_EXTENSIONS = set(['txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'])

# 設定自己的 line token & 接收者，資料都在 https://developers.line.biz/en/ 登入後尋找
recipient = “ooooooooooooooooo"
line_token = “XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXxx”
# line_token = ""
```

中繼伺服器

```
app = Flask(__name__)
# web service 框架

def allowed_file(filename):
    # 允許檔案類型
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

@app.route('/')
def upload_form():
    # 在 localhost 打開一個網頁

@app.route('/upload', methods=["GET",'POST'])
def web_fileupload():
    # (ip)/upload
    # 上傳並辨識圖片 可以選擇使用 opencv 或 watson 辨識
```

```
def opencv_face(file_name):
    # opencv VR

def watson_face(file_name):
    # watson VR

def posttoline(sendmessage):
    # 用機器人發辨識結果

if __name__ == '__main__':
    # 在 local 開始一個 server
    app.run(host='0.0.0.0',port=80)
```

中繼伺服器

Python server

- 調整程式變成串接Arduino使用

```
line_token = "vFkzuCFanieNHh9eeAZl9cl31jdbl
# line_token = ""
```

```
# line_token = "vFkzuCFanieNHh9eeAZl
line_token = ""
```

```
    if request.method == 'POST':
        try:
#             type=request.args.get("message")
#             line_token = request.args.get("Bearer")
            type = "1"
```

```
    if request.method == 'POST':
        try:
            type=request.args.get("message")
            line_token = request.args.get("Bearer")
#             type = "1"
```

TRY IT!

Q & A

北祥股份有限公司 Michael