

# 111B 人工智慧及其應用

## 期末專題報告 Part I

組別：電子學實驗小組

報告日期：2023-06-13

組員：謝在宥<sup>(P1)</sup>、李泳諒<sup>(P2)</sup>、施家峻<sup>(P2)</sup>

主題：ESP32 氣象站

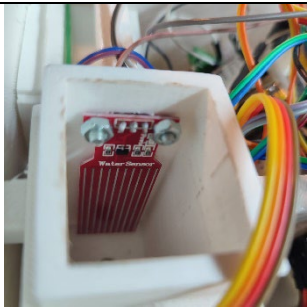
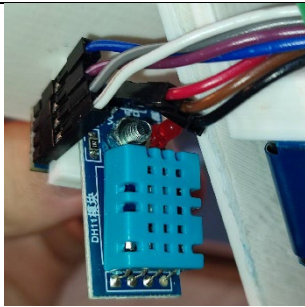
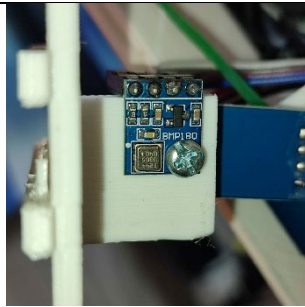
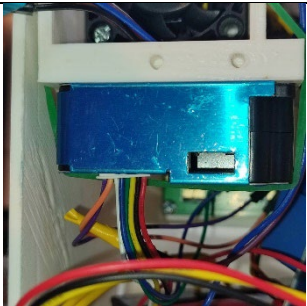
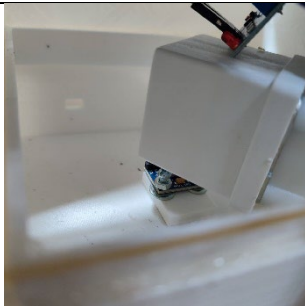
貳、設計規劃 .....	2
一、感測器規劃 .....	2
二、模組規劃 .....	3
三、設計圖 .....	4
參、實際組裝 .....	7
一、外殼數計(使用 Blender 3D 建模) .....	7
二、切片(轉檔 Gcode) .....	8
三、3D 列印 .....	8
四、組裝流程、組裝影片 五、3D 列印物件 .....	8
肆、成品 .....	10
一、ESP32 腳位選定 .....	10
二、降雨量計算(小時累計雨量) .....	10
三、I <sup>2</sup> C 推薦使用 .....	10
四、未來展望 .....	11
伍、資料監看 .....	12
一、上傳欄位列表 .....	12
二、ThingSpeak IoT 平臺 .....	12
陸、附錄 .....	13
附錄一、外觀照片 .....	13
附錄二、製作過程縮時影片連結 .....	13
附錄三、製作資訊分享 .....	14

## 壹、設計動機

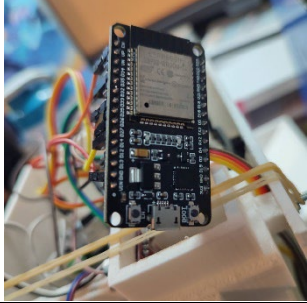
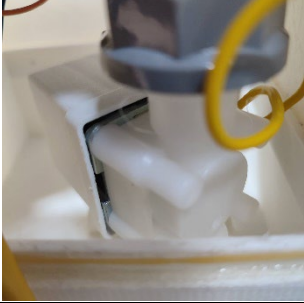
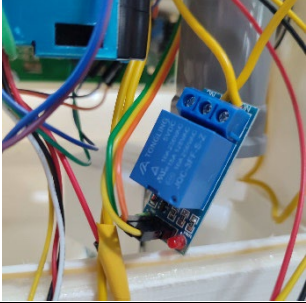



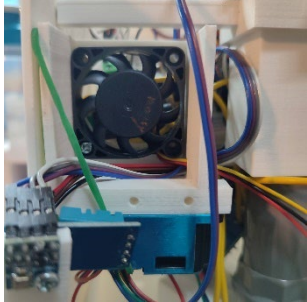

現行氣象站測量降雨量較無自動化的方式，因此想製作個能測量累計(小時)雨量的裝置；附加其他氣象資料，溫度、濕度、氣壓、加速度(地震)、風速、風向、懸浮微粒。考量到防水因素與短時間製作因素，風向與風速並未納入本次專題，未來有機會進行後續發展。

## 貳、設計規劃

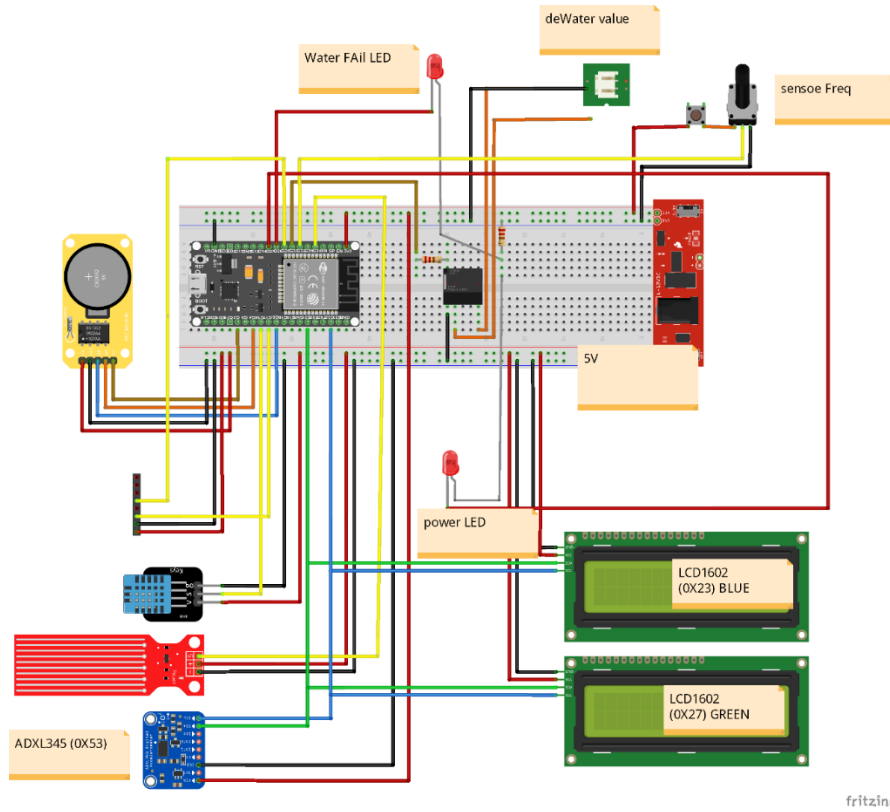
### 一、感測器規劃

			
型號	水位傳感器?	DHT11	BMP180
價格	10	30	25
測值 1	水位高度	溫度(精度低)	大氣壓力
測值 2		濕度	溫度
傳輸方式	analogRead	Digital Signal	I2C
			
型號	PMS3003	ADXL345	
價格	600	70	
測值	懸浮微粒 pM1.0、pM2.5、pM10	加速度 X、Y、Z	
通訊方式	Digital Signal	I2C	

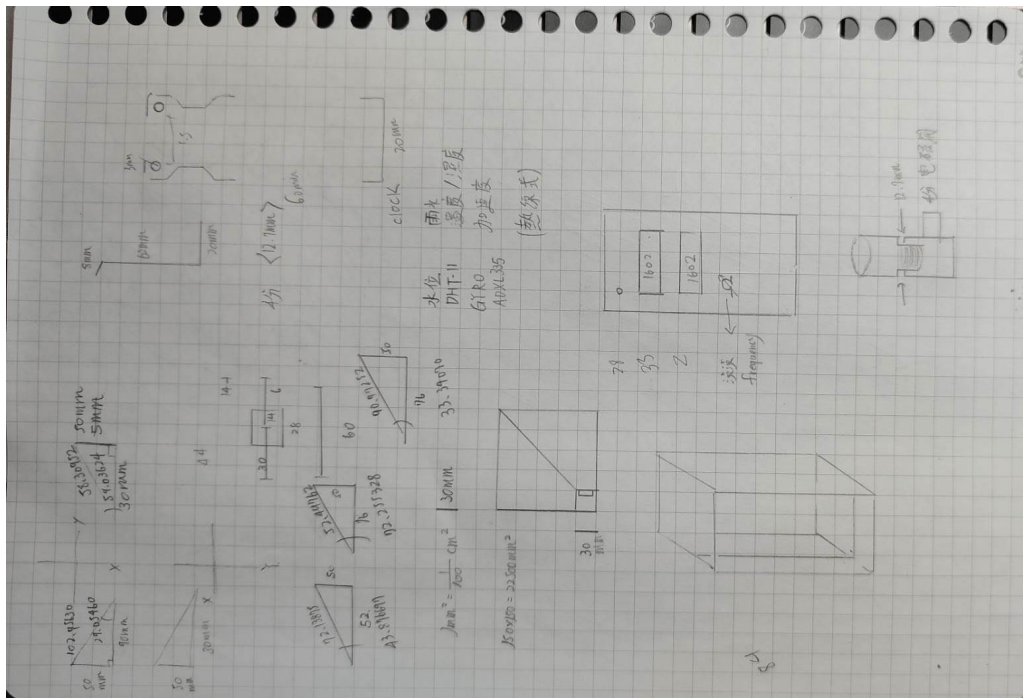
## 二、模組規劃

			
名稱	ESP32	排水電磁水閥 12v	電磁水閥繼電器
價格	195	120	22
			
名稱	LCD1602*2 個	DS3231(時鐘)	三串 18650 (4.2v) 12.6v 電池組
價格	120*2	85	35+100*3
			
名稱	12v(4cm)風扇	5V、3v3 電源供應器	
價格	拆舊品	23	

## 1. 電路圖(初步規劃)

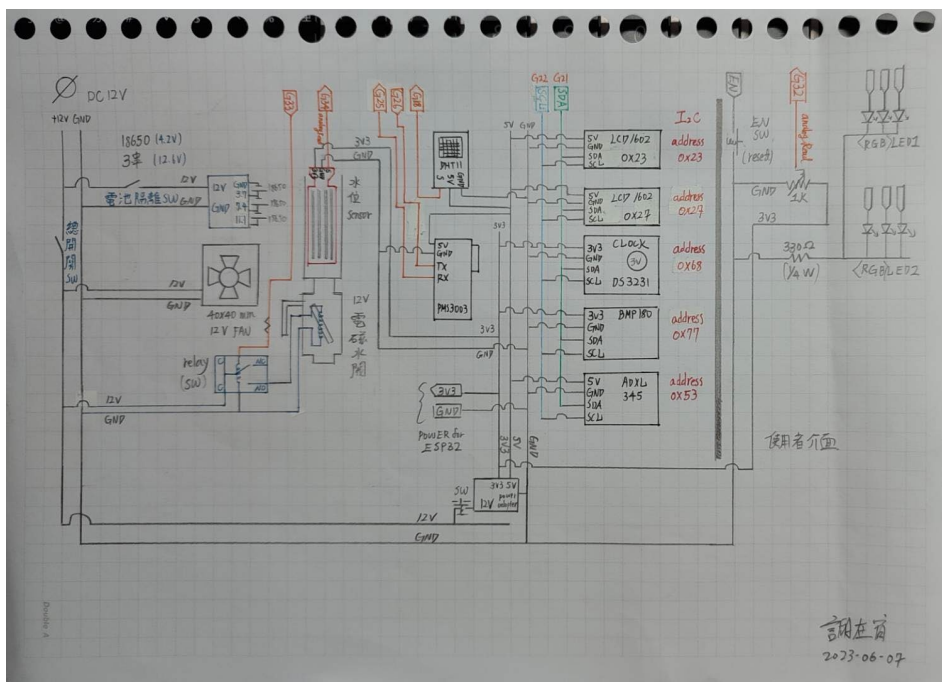
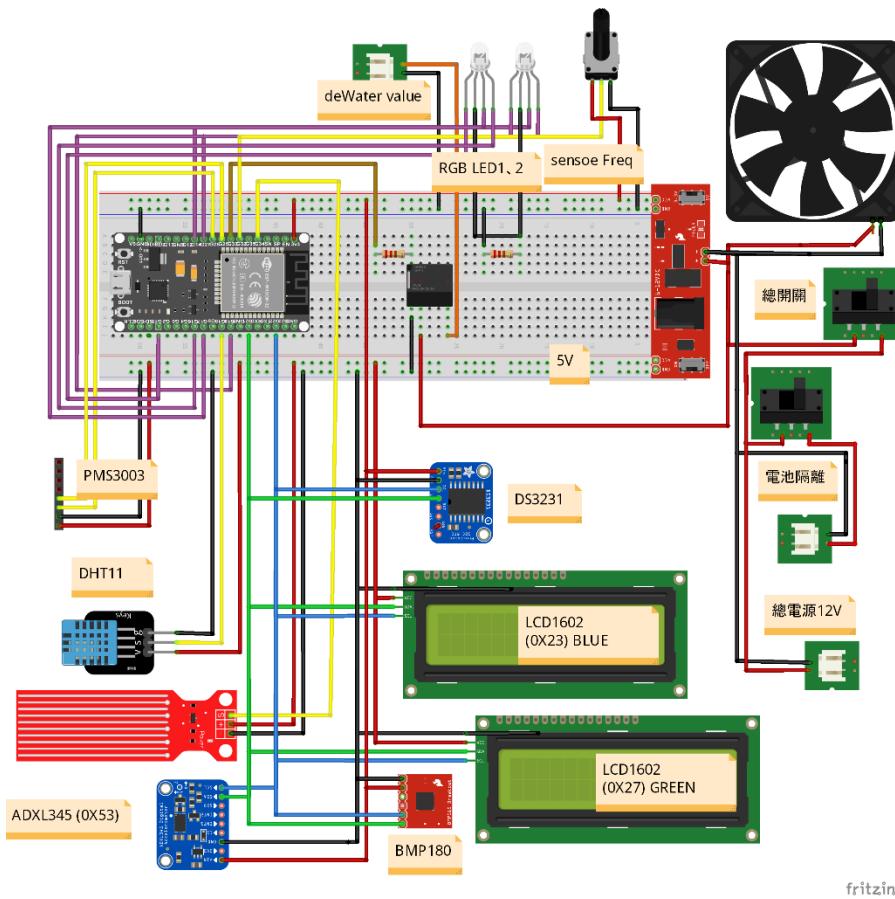


## 2. 外觀設計(初步規劃)



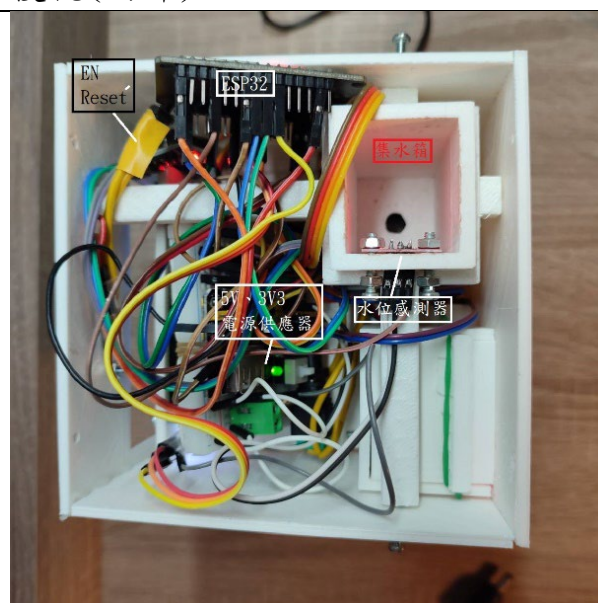
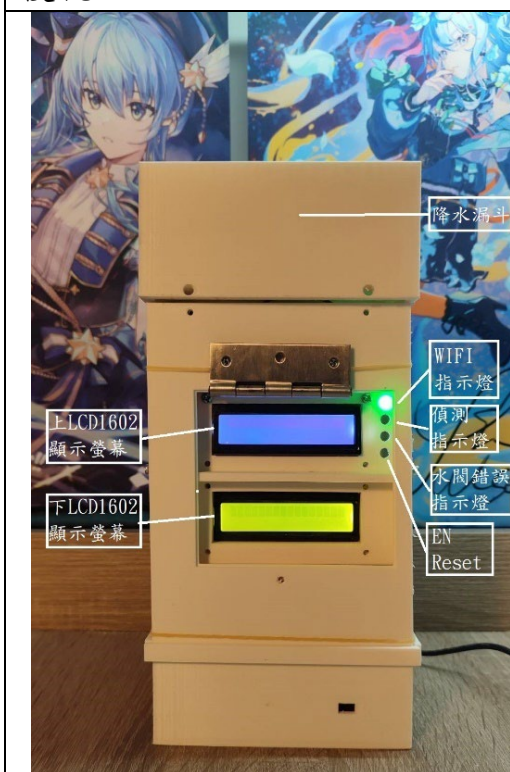
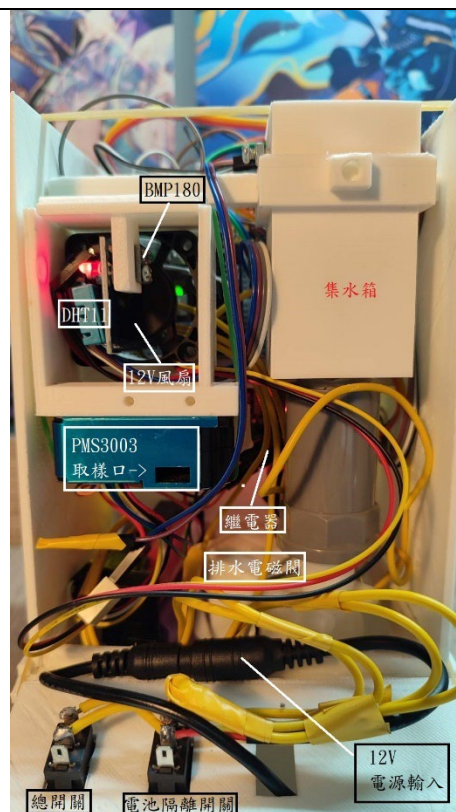
### 3. 電路圖(完整版)

註 6/8: ESP32 電壓源改 5V 輸入 VIN 腳(取消 3V3)





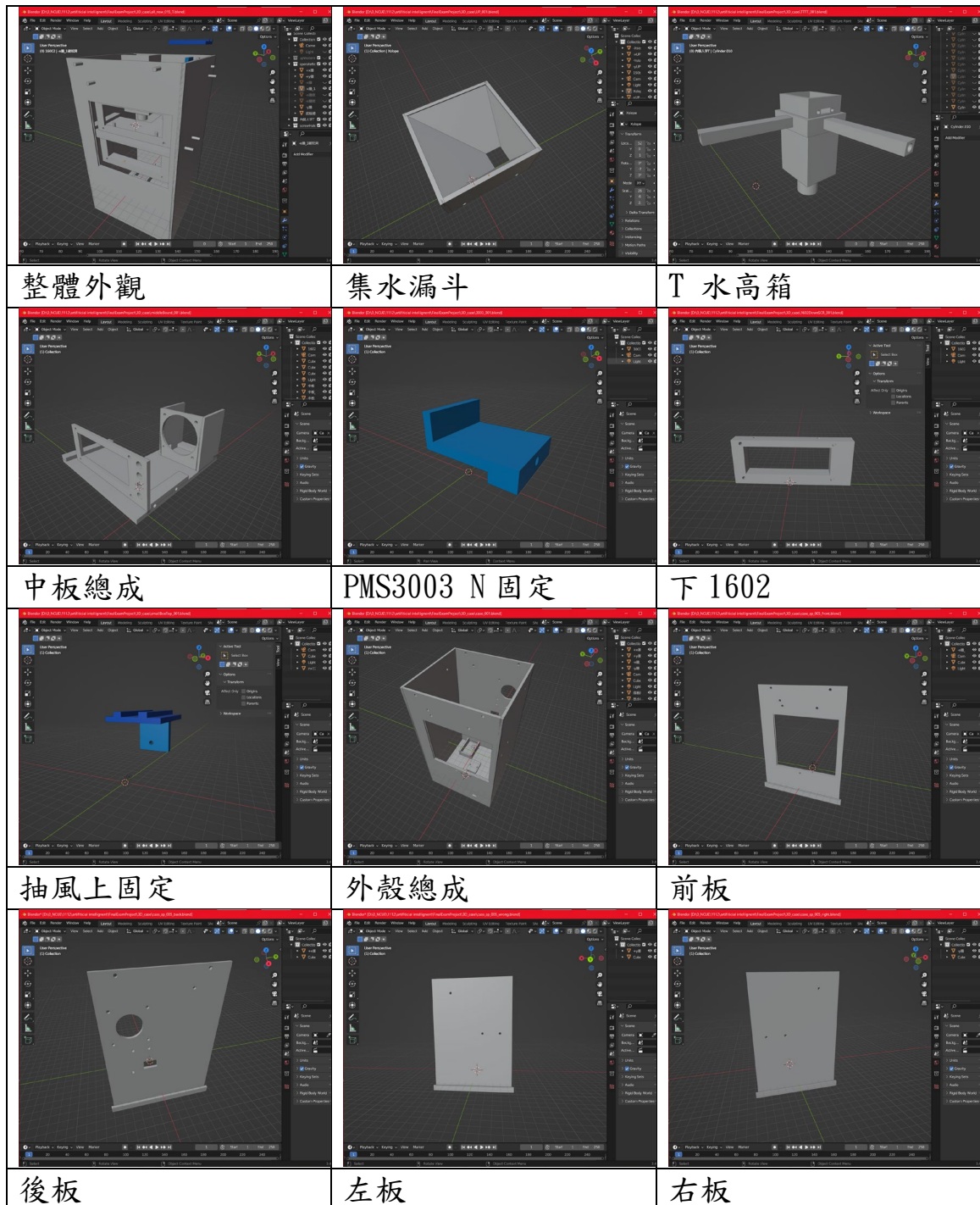
#### 4. 設備介紹



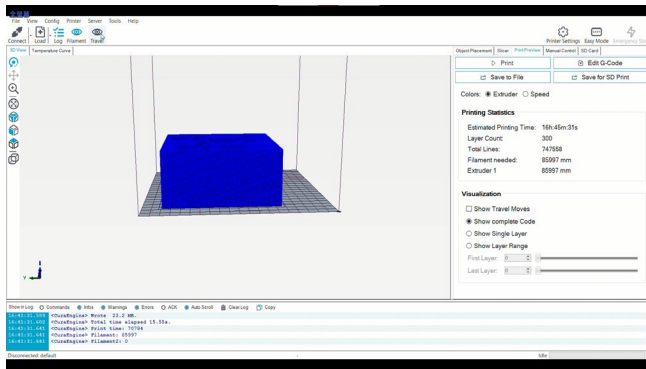
← 前視(操作、顯示介面)

參、實際組裝

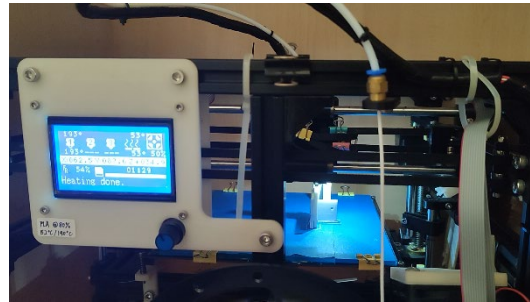
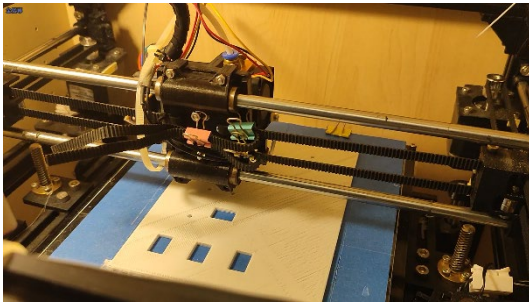
一、外殼數計(使用 Blender 3D 建模)



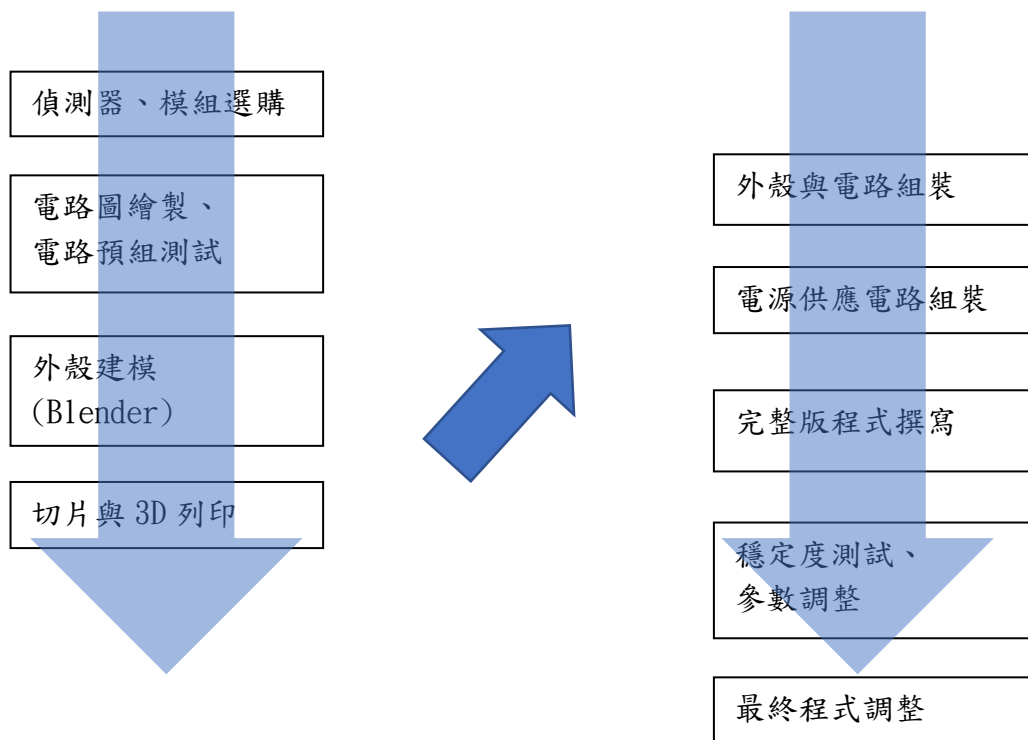
## 二、切片(轉檔 Gcode)



## 三、3D 列印

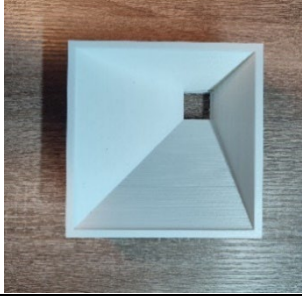
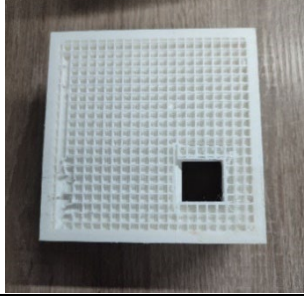



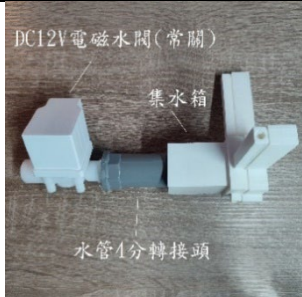

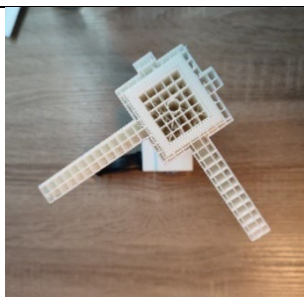
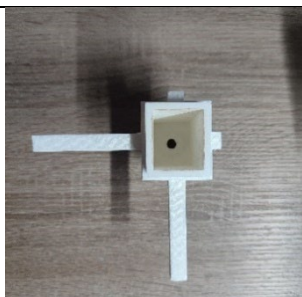
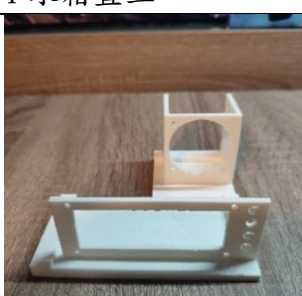


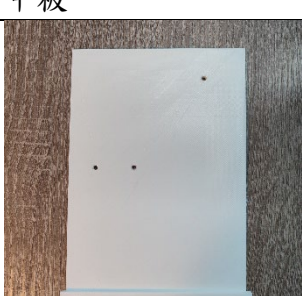
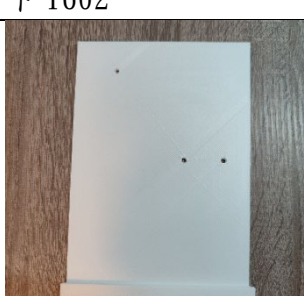
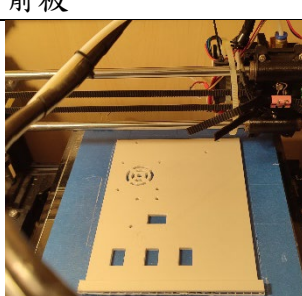


## 四、組裝流程、組裝影片



## 五、3D 列印物件



		
集水漏斗	集水漏斗(底)	集水漏斗(側)
		
箱底(上視)	箱底(前視)	T 水箱介紹
		
T 水箱直立	原始列印(未拆支架)	拆支架後
		
中板	下 1602	前板
		
左板	右板	後板

## 肆、成品

### 一、ESP32 腳位選定

參考網址: <https://zerotech.club/esp32-gpio/>

```
ESP32 pinFunction: Namepad
File Edit Format View Help
G18 DHT11 digitalRead (VIN 5V) 88
G34 waterSensor analogRead (VIN 5V) 88
G33 waterValue HIGH sig (水閘) 88
//G32 freqResistance (5V)
//G27 powerLight LED
G27 waterERROR LED
G25 PMS3003 Serial G26
//G17 PMS3003 sleep (standby0/operate1)
G5 CLK/SCLK 5
G4 IO/DAT 4
G2) RS/CET 2 DS1302 clock
G16 時間設定用 只有在與電腦連接時接3V3

17LED
16LED
-----
I2C
1602
ADXL345(加速度) 83(0X53)
I2C SDA 21 green / blue
SCL 22 blue //purple
analog in 34 35 36 39
```

### 二、降雨量計算(小時累計雨量)

降雨量測定為「單位面積的累計降水高度」。設計透過上方漏斗接取降水(接取底面積 115\*115mm)，透過斜面流入水箱(底面積 32\*28mm)；以體積=底面積\*高換算，面積縮小使高度變化量增大，將測量解析度放大。

註: ESP32 analogRead 於 3V3 值為 4095。

註: 檢測高度指水位高度在水箱內變化時可被測量到的總高度區間。

### 計算公式

`analogRead(waterPin) = (realWaterVal/4095)*40*(32*28-20*8)/(115*115) ; //輸出期間水位高度(毫米 mm) = realWaterVal/4095 * 檢測高度 * (水箱底面積-水位傳感器截面積)/(接取底面積)`

### 三、I<sup>2</sup>C 推薦使用

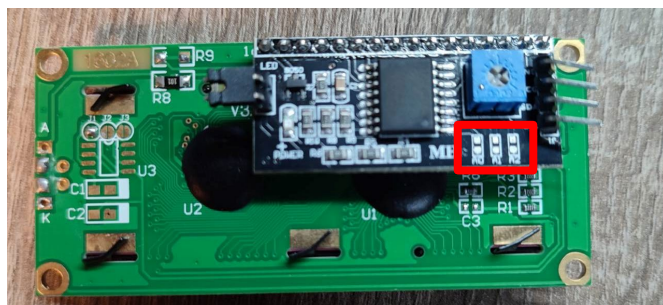
I<sup>2</sup>C 位址模組列表

參考網址: <https://learn.adafruit.com/i2c-addresses/the-list>

LCD1602 I<sup>2</sup>C 位址改寫

參考網址: <https://dotblogs.com.tw/JWPL102216/2016/12/10/213049>

1. 為何可同時接兩個 LCD1602 螢幕? 位址不會沖到嗎?



	A0	A1	A2
0x27	0	0	0
0x26	1	0	0
0x25	0	1	0
0x24	1	1	0
0x23	0	0	1
0x22	1	0	1
0x21	0	1	1
0x20	1	1	1

## 四、未來展望

### 1. 風速、風向測定

使用熱線式風速儀：

透過風速不同導致熱線線圈為維持溫度而提供的加熱功率換算為風速，易受環境背景溫度影響。

### 2. 外觀改良

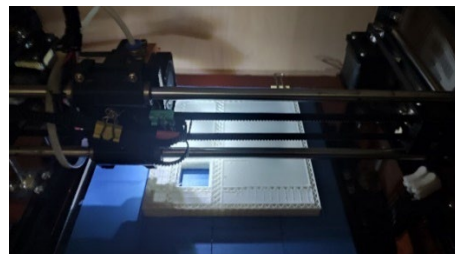
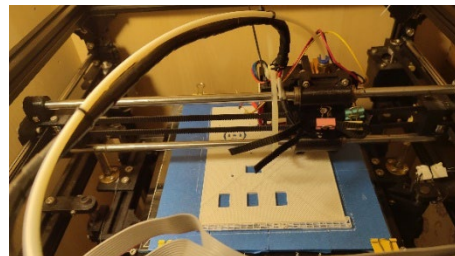
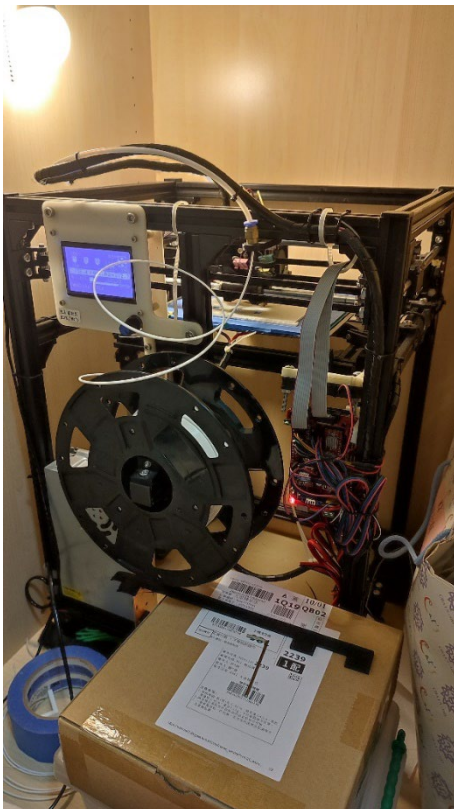
#### A. 3D 列印熱脹冷縮的問題

下 1602 與中板 1602 螢幕架完成列印時，螺孔與螢幕方框吻合，但材料 (PLA) 冷卻後導致方框變大、螺孔位移，無法四個螺絲皆固定。

改良：

- a. 建模時預留位置(需靠經驗累積~)。
- b. 尋找可計算溫度影響的切片軟體。
- c. 趁尚未降溫時固定。

**註：**3D 列印時為確保列印物件與平臺暫時固定，會在平臺下方加熱(俗稱熱床)，溫度約控制在 50 度(實際因平臺面積過大、散熱、加熱因素，設定於 53 度)，完成物件列印降溫後，平臺即與物件分離。也因為減少散熱，我將 3D 列印機放在衣櫃裏面，列印時將門關上。這臺是我國二時自組的，因此外觀上有些雜亂~~





## B. 前掀蓋與擋水板設計

因為時間不足沒做前掀蓋擋板，未來有機會可用透明 3D 列印材料。  
若要在戶外實際使用，空氣取樣孔與線孔皆應裝設防雨擋板。

## C. 充電孔獨立開孔

## D. 進線電源隔離開關

目前 DC12V 充電孔使用進線電源隔離開關的預留位置，未來可設計為固定式的 12V(外 5.5 內 2.1)的母接頭，使電源供應器可分離。並加裝一個電源供應器的進線隔離開關，使操作上更為方便。

# 伍、資料監看

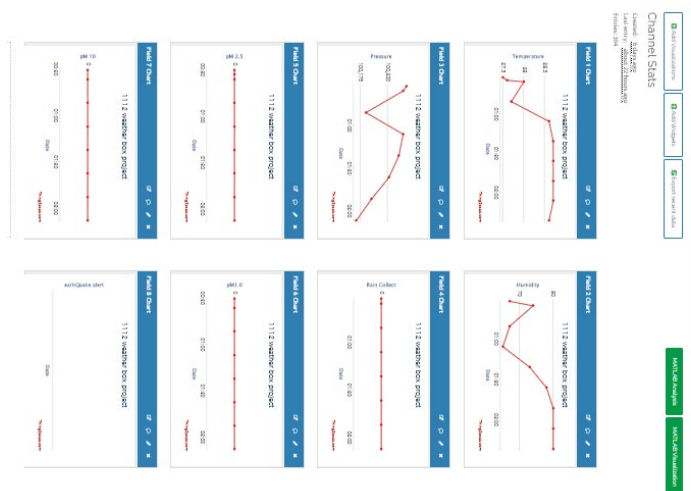
## 一、上傳欄位列表

欄位	Label 1	Label 2	Label 3	Label 4
項目	溫度(BMP180)	濕度(DHT11)	氣壓(BMP180)	降雨量(水位)
備註				
欄位	Label 5	Label 6	Label 7	Label 8
項目	pM2.5(PMS3003)	pM1.0(PMS3003)	pM10(PMS3003)	地震加速度(ADXL345)
備註	softSerial 相容測試中 (baudrate 9600 不同)			未來延伸用

## 二、ThingSpeak IoT 平臺

Channel ID: 2167549

<https://thingspeak.com/channels/2167549>



電腦板監看畫面(網頁)排版因素倒著



手機 APP 版監看畫面



## 陸、附錄

### 附錄一、外觀照片



### 附錄二、製作過程縮時影片連結

<https://youtu.be/B3E9KYg2nBQ>



## 附錄三、製作資訊分享

### 一、檔案與程式碼

檔案連結：

[https://github.com/NCUE-S1151026/111B\\_Artificial-Intelligence-and-Its-Application\\_finalProject](https://github.com/NCUE-S1151026/111B_Artificial-Intelligence-and-Its-Application_finalProject)



### 二、Arduino C 程式碼

```
/*
Author:          Josh Hsieh
well done Date:  2023-06-08
Description:      For NCUE 111B Technology Application and Innovation-00258
Advisor:
*/

#include <WiFi.h>
#include "ThingSpeak.h"
#include <RTClib.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>
#include "DHT.h"
#include <Adafruit_ADXL345_U.h>
#include <LiquidCrystal_PCF8574.h>
#include <SoftwareSerial.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
//用於電源不穩不重開機

#define spareTime 15
//間格測資時間

#define waterPin 34
//analogRead
// #define errorPin 27
//error LED
#define drainPin 33
//電磁水閥 relay
#define dht11Pin 18
//DHT11 G18
#define DHTTYPE DHT11

#define PMS_RXPin 25
#define PMS_TXPin 26
//PMS3003
#define LED1RPin 15
#define LED1GPin 16
#define LED1BPin 17
#define LED2RPin 19
#define LED2GPin 27
#define LED2BPin 14
```

```

/*-----
RGB LED pin/ light staus
mode 1 -- WIFI connecting          1RG
mode 2 -- WIFI upload details      1B
mode 3 -- WIFI unload all success   1G
mode 4 -- WIFI upload details error 1R
mode 5 -- getACCleration            1RB
mode 6 --
mode 7 -- setup                     2RG
mode 8 -- setup success              2RGB
mode 9 -- PMS3003                    2B
mode 10 -- getWaterHight             2G
mode 11 -- getBM                     2RB
mode 12 -- getDH                     2BG
mode 13 -- getWaterHight error       2R
mode 14 --
-----
*/

//3003
SoftwareSerial PmsSerial(PMS_RXPin, PMS_TXPin); // RX, TX
// SoftwareSerial WifiSerial(4, 5); // RX, TX
int nFirstTime=1 ;
static unsigned int pm_cf_10;          //定義全域變數
static unsigned int pm_cf_25;
static unsigned int pm_cf_100;

//water
int baseWaterVal = 0 ; //基本底線水位
const int maxWaterVal = 2700 ; //最高上限水位
const int lowWaterVal = 500 ; //排水停止水位
int dyanWaterVal = 0 ; //動態偵測水位
int lastWaterVal = 0 ; //前一次水位值
int realWaterVal = 0 ; //本次實際水位值
float waterHight = 0 ; //輸出期間水位高度(毫米 mm)

//3231
RTC_DS3231 rtc;
char t[32];

//BMP180
Adafruit_BMP085 bmp;
float bmpTemper = 0 ;
float bmpPreas = 0 ;

//ADXL345
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
float accX = 0 ; //x 加速度
float accY = 0 ; //y 加速度
float accZ = 0 ; //z 加速度

//DHT11
DHT dht(dht11Pin, DHTTYPE); // constructor to declare our sensor

```

```

float dhTemper = 0 ;
float dhHimi = 0 ;

//1602
LiquidCrystal_PCF8574 lcdB(0x23);
LiquidCrystal_PCF8574 lcdG(0x27);

//WIFI
const char* ssid = "----WIFI_SSID----"; // your network SSID (name)
const char* password = "----PASSWORD----"; // your network password
// const char* ssid = "TANetRoaming";
//ThingSpeak
unsigned long myChannelNumber = ----ThingSpeak_Channel_ID----;
const char * myWriteAPIKey = "----ThingSpeak_API_Key----";

//upload 資料整理
float uploadLabel[8] = {bmpTemper, dhHimi, bmpPreas, waterHight, pm_cf_25,
pm_cf_10, pm_cf_100};
String uploadLabelName[8] = {"bmpTemper", "dhHimi", "bmpPreas", "waterHight",
"pm_cf_25", "pm_cf_10", "pm_cf_100"};
unsigned int testCount = 1 ;
int cccc = 0;

WiFiClient client;

void setup() {
    // put your setup code here, to run once:
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //關閉電源不穩就重開機的設定
    ledSatus(7,1);
    PmsSerial.begin(9600);
    PmsSerial.setTimeout(1500);
    Serial.begin(115200);
    setup1602();
    rtc.begin(); // Initialize DS3231
    Wire.begin(); // DS3231 use
    //rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
    //set Ds3231 time//set time use ||
    //remember to mask after set time!!!!!!!!!!!!!!

    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client); // Initialize ThingSpeak

    setupLED() ;
    dht.begin() ; // Initialize DHT11
    bmp.begin() ; // Initialize BMP180
    setupWaterHight();
    ledSatus(7,0);
    ledSatus(8,1);

    lcdG.setCursor(0, 0);
    lcdG.print(printDateTen());
    lcdG.setCursor(0, 1);
    lcdG.print(printTimeEig());

```



```

    lcdB.setCursor(0, 0);
    lcdB.print("setup success");
    lcdB.setCursor(0, 1);
    lcdB.print(WiFi.macAddress());

    delay(1500);
    ledSatus(8,0);
    lcdB.clear();
    lcdB.setCursor(0, 0);
    lcdB.print("WIFI:"+ String(ssid));
}

void setup1602(){
    lcdB.begin(16, 2); // initialize the lcd
    lcdG.begin(16, 2); // initialize the lcd
    lcdB.setBacklight(200);
    lcdB.home();
    lcdB.clear();
    lcdG.setBacklight(200);
    lcdG.home();
    lcdG.clear();
}

void setupLED(){
    pinMode(LED1RPin, OUTPUT);
    pinMode(LED1GPin, OUTPUT);
    pinMode(LED1BPin, OUTPUT);
    pinMode(LED2RPin, OUTPUT);
    pinMode(LED2GPin, OUTPUT);
    pinMode(LED2BPin, OUTPUT);
}

void setupWaterHight(){
    pinMode(waterPin, INPUT);
    pinMode(drainPin, OUTPUT);

    baseWaterVal = analogRead(waterPin); //設定基礎水位

    while(baseWaterVal < lowWaterVal){ //液位面不足，加水
        //show error message on 1602//
        lcdB.clear();
        lcdB.setCursor(0, 0);
        lcdB.print("water drain error");
        baseWaterVal = analogRead(waterPin);
        ledSatus(13,0);
        delay(4000);
        ledSatus(13,0);
        delay(500);
    }
    lastWaterVal = baseWaterVal;
}

void getWaterHight(){
    ledSatus(10,1);
    dyanWaterVal = analogRead(waterPin);
}

```

```

realWaterVal = dyanWaterVal - lastWaterVal ; //本次實際水位值(小時累計 4095)
waterHight = realWaterVal*0.0005539727 ;
//輸出期間水位高度(毫米 mm) = realWaterVal/4095 *60 *(28*28)/(144*144)
lastWaterVal = dyanWaterVal ; //更新本次

if(dyanWaterVal > maxWaterVal) { //執行排水
    while(dyanWaterVal > lowWaterVal){
        digitalWrite(drainPin, HIGH);
        delay(200) ; //校正!!!! 續測時間
    }
    digitalWrite(drainPin, LOW); //關閉水閥
    delay(30000);
    baseWaterVal = analogRead(waterPin); //設定基礎水位
    lastWaterVal = baseWaterVal;
}
Serial.print(waterHight) ;
Serial.println("waterHight") ;
ledSatus(10,0);
}

void getBM(){
    ledSatus(11,1);
    bmpTemper = bmp.readTemperature() ;
    bmpPreas = bmp.readPressure() ;
    //bmp.readAltitude() ;
    //bmp.readSealevelPressure() ;
    Serial.print(bmpTemper) ;
    Serial.println("bmpTemper") ;
    Serial.print(bmpPreas) ;
    Serial.println("bmpPressure") ;
    ledSatus(11,0);
}

void getDH(){
    ledSatus(12,1);
    dhHimi = dht.readHumidity();
    //Read the moisture content in %.
    dhTemper = dht.readTemperature();
    //uploadLabel[?] = dhTemper ;
    //Read the temperature in degrees Celsius
    // = dht.readTemperature(true);
    // true returns the temperature in Fahrenheit
    Serial.print(dhHimi) ;
    Serial.println("dhHimi") ;
    Serial.print(dhTemper) ;
    Serial.println("dhTemper") ;
    ledSatus(12,0);
}

void getAccleration(){
    ledSatus(5,1);
    /* Get a new sensor event */
    sensors_event_t event;
    accel.getEvent(&event);

```

```

/* Display the results (acceleration is measured in m/s^2) */
accX = event.acceleration.x ;
accY = event.acceleration.y ;
accZ = event.acceleration.z ;
Serial.print(accX) ;
Serial.println("accX") ;
Serial.print(accY) ;
Serial.println("accY") ;
Serial.print(accZ) ;
Serial.println("accZ") ;
//Serial.println("m/s^2 ");
ledSatus(5,0);
}

void getG5(unsigned char ucData)//獲取 G5 的值
{
    static unsigned int ucRxBuffer[250];
    static unsigned int ucRxCnt = 0;
    ucRxBuffer[ucRxCnt++] = ucData;
    if (ucRxBuffer[0] != 0x42 && ucRxBuffer[1] != 0x4D)//資料頭判斷
    {
        ucRxCnt = 0;
        return;
    }

    if (ucRxCnt > 16)//資料位元數判斷//G5T 為 16
    {
        pm_cf_10=(int)ucRxBuffer[4] * 256 + (int)ucRxBuffer[5];
        //大氣環境下 PM2.5 濃度計算
        pm_cf_25=(int)ucRxBuffer[6] * 256 + (int)ucRxBuffer[7];
        pm_cf_100=(int)ucRxBuffer[8] * 256 + (int)ucRxBuffer[9];
        // pm_at_10=(int)ucRxBuffer[10] * 256 + (int)ucRxBuffer[11];
        // pm_at_25=(int)ucRxBuffer[12] * 256 + (int)ucRxBuffer[13];
        // pm_at_100=(int)ucRxBuffer[14] * 256 + (int)ucRxBuffer[15];

        if (pm_cf_25 > 999 || pm_cf_10 > 999 || pm_cf_100 >999)
            //如果 PM2.5 數值>1000，返回重新計算
        {
            ucRxCnt = 0;
            return;
        }

        ucRxCnt = 0;
        return;
    }
}

void get3003(){
    //422230P
    ledSatus(9,1);
    PmsSerial.listen();
    if (PmsSerial.isListening())
    {

```

```

    Serial.println("PmsSerial.isListening");
}else
{
    Serial.println("PmsSerial.is not Listening");
}

while (PmsSerial.available()) {
    getG5(PmsSerial.read());
}

// this part is for retrieving useful values
if (pm_cf_10 ==0 && pm_cf_25 ==0 && pm_cf_100 ==0)
{
    Serial.println("All are 0s");
    // new added; restart
    nFirstTime=1;
    delay(1000); // avoid to read too many times
    // return; //2256 爛啟用
}
ledSatus(9,0);
/////////42223ED
}

String printTimeEig(){
    DateTime now = rtc.now();
    String TTprint = "";

    if(int(now.hour())<10){
        TTprint = "0"+ String(now.hour()) + ":";
    }
    else
    {
        TTprint = String(now.hour()) + ":";
    }

    if(int(now.minute())<10){
        TTprint = TTprint + "0"+ String(now.minute()) + ":";
    }
    else{
        TTprint = TTprint + String(now.minute()) + ":";
    }

    if(int(now.second())<10){
        TTprint = TTprint + "0"+ String(now.second());
    }
    else{
        TTprint = TTprint + String(now.second());
    }

    return(TTprint);
}

String printDateTen(){

```



```

DateTime now = rtc.now();
String TDprint = (String(now.year()) + "/");

if(int(now.month())<10){
    TDprint = TDprint + "0"+ String(now.month()) + "/";
}
else{
    TDprint = TDprint + String(now.month()) + "/";
}

if(int(now.day())<10){
    TDprint = TDprint + "0"+ String(now.day());
}
else{
    TDprint = TDprint + String(now.day());
}

return(TDprint);
}

void ledSatus(int modew, bool boo){ //which mode? / on or off
    if(boo == 1){ //ON
        switch(modew){
            case 1:
                digitalWrite(LED1RPin, HIGH);
                digitalWrite(LED1GPin, HIGH);
                break;
            case 2:
                digitalWrite(LED1BPin, HIGH);
                break;
            case 3:
                digitalWrite(LED1GPin, HIGH);
                break;
            case 4:
                digitalWrite(LED1RPin, HIGH);
                break;
            case 5:
                digitalWrite(LED2RPin, HIGH);
                digitalWrite(LED2BPin, HIGH);
                break;
            case 7:
                digitalWrite(LED2RPin, HIGH);
                digitalWrite(LED2GPin, HIGH);
                break;
            case 8:
                digitalWrite(LED2RPin, HIGH);
                digitalWrite(LED2GPin, HIGH);
                digitalWrite(LED2BPin, HIGH);
                break;
            case 9:
                digitalWrite(LED2BPin, HIGH);
                break;
            case 10:

```

```

        digitalWrite(LED2GPin, HIGH);
        break;
    case 11:
        digitalWrite(LED2RPin, HIGH);
        digitalWrite(LED2BPin, HIGH);
        break;
    case 12:
        digitalWrite(LED2BPin, HIGH);
        digitalWrite(LED2GPin, HIGH);
        break;
    case 13:
        digitalWrite(LED2RPin, HIGH);
        break;
    }
}
else{ //Off
    switch(modeW){
        case 1:
            digitalWrite(LED1RPin, LOW);
            digitalWrite(LED1GPin, LOW);
            break;
        case 2:
            digitalWrite(LED1BPin, LOW);
            break;
        case 3:
            digitalWrite(LED1GPin, LOW);
            break;
        case 4:
            digitalWrite(LED1RPin, LOW);
            break;
        case 5:
            digitalWrite(LED2RPin, LOW);
            digitalWrite(LED2BPin, LOW);
            break;
        case 7:
            digitalWrite(LED2RPin, LOW);
            digitalWrite(LED2GPin, LOW);
            break;
        case 8:
            digitalWrite(LED2RPin, LOW);
            digitalWrite(LED2GPin, LOW);
            digitalWrite(LED2BPin, LOW);
            break;
        case 9:
            digitalWrite(LED2BPin, LOW);
            break;
        case 10:
            digitalWrite(LED2GPin, LOW);
            break;
        case 11:
            digitalWrite(LED2RPin, LOW);
            digitalWrite(LED2BPin, LOW);
            break;
        case 12:
            digitalWrite(LED2BPin, LOW);

```

```

        digitalWrite(LED2GPin, LOW);
        break;
    case 13:
        digitalWrite(LED2RPin, LOW);
        break;
    }
}
}

void loop() {
    DateTime now = rtc.now();
    if(((now.minute()) % spareTime) == 0 || testCount == 1){ //整點或第一次測資
        // put your main code here, to run repeatedly:

        // Connect or reconnect to WiFi
        if(WiFi.status() != WL_CONNECTED){
            lcdB.setCursor(0, 1);
            lcdB.print("try connect WIFI");
            Serial.print("Attempting to connect");
            while(WiFi.status() != WL_CONNECTED){
                ledSatus(1,1); //WIFI connect on
                WiFi.begin(ssid, password);
                delay(5000);
            }
            Serial.println("\nConnected.");
            lcdB.setCursor(0, 1);
            lcdB.print("WIFI connect OK ");
            ledSatus(1,0); //WIFI connect off
            delay(1500);
        }

        lcdB.clear();
        lcdG.clear();
        lcdB.setCursor(0, 0);
        lcdB.print(printDateTen());
        lcdB.setCursor(11, 0);
        lcdB.print("C"+String(testCount)); //益位注意
        lcdB.setCursor(0, 1);
        lcdB.print(printTimeEig());
        lcdB.setCursor(9, 1);
        lcdB.print(String(myChannelNumber));

        get3003();
        getWaterHight() ;
        getBM() ;
        getDH() ;

        //upload to thingSpeak(loop for 8 fields)
        uploadLabel[0] = bmpTemper ;
        uploadLabel[1] = dhHimi ;
        uploadLabel[2] = bmpPreas ;
        uploadLabel[3] = waterHight ;
        uploadLabel[4] = pm_cf_25 ;
        uploadLabel[5] = pm_cf_10 ;
        uploadLabel[6] = pm_cf_100 ;
    }
}

```

```

for(int i=1; i<8; i++){
    Serial.println(i);
    Serial.println(uploadLabel[i-1]);
    ledSatus(2,1);
    lcdG.clear();
    lcdG.setCursor(0, 0);
    lcdG.print(uploadLabelName[i-1]);
    lcdG.setCursor(1, 1);
    lcdG.print(uploadLabel[i-1]);
    int Fs = ThingSpeak.writeField(myChannelNumber,i, uploadLabel[i-1],
myWriteAPIKey);
    cccc = 0;
    while(Fs != 200){
        ledSatus(4,1);
        Serial.println(" error channel. HTTP error code " + String(Fs));
        lcdG.setCursor(11, 0);
        lcdG.print("error");
        lcdG.setCursor(11, 1);
        lcdG.print(String(Fs));
        ledSatus(2,0);
        delay(5000);
        lcdG.setCursor(11, 1);
        lcdG.print("cE>" + String(cccc));
        delay(10000);
        ledSatus(4,0);
        cccc=cccc+1;

        if(cccc == 20){
            break;
        }
    }
    Serial.println(" upload SUCCESS");
    lcdG.setCursor(14, 0);
    lcdG.print("UP");
    lcdG.setCursor(14, 1);
    lcdG.print("OK");
    ledSatus(2,0);
    delay(15000);
}
Serial.println("ALL SUCCESS");
lcdG.clear();
lcdG.setCursor(0, 0);
lcdG.print("C" + String(testCount)); //益位注意
lcdG.setCursor(3, 1);
lcdG.print("ALL success");
testCount = testCount + 1;
delay(10000);
}
//隨時循環顯示
for(int i=1; i<8; i++){
    lcdG.clear();
    lcdG.setCursor(0, 0);
    lcdG.print(uploadLabelName[i-1]);
    lcdG.setCursor(3, 1);

```



```
        lcdG.print(uploadLabel[i-1]);  
        delay(3000);  
    }  
    lcdG.clear();  
    lcdG.setCursor(0,0);  
    lcdG.print("recTime");  
    lcdG.setCursor(8,1);  
    lcdG.print(printTimeEig());  
    delay(3000);  
    lcdG.clear();  
    lcdG.setCursor(0,0);  
    lcdG.print("tset Freq");  
    lcdG.setCursor(12,1);  
    lcdG.print(String(spareTime) + "M");  
    delay(3000);  
}
```