

臺北市立大安高級工業職業學校

電子科

專題報告

IOT小型氣象觀測站

IOT small weather observatory.

組長：劉沛勳

組員：洪聿民

組員：徐丞霆

組員：賴霆安

指導老師：薛元陽

中華民國 112 年 1 月

中文摘要

在農業中，許多人需要一些關於天候氣象的數值，來確保自農作物環境。

本專題使用不同的感測器，偵測不同的天候數值，偵測到的數值能夠傳輸到攜帶性裝置上，讓我們可以直接偵測到當下的氣象數值，有了數值再去做運算與推論就可以得知可能的氣候，並對農作物作相對應的保護，即可避免重大災害來臨時的損失。

關鍵字：arduino，物聯網，太陽能，ESP32，氣象觀測站

Abstract

In many agriculture industries. They need instant or regular meteorological data for taking care of their crops or plants. If you have WIFI, you can get the latest meteorological data on your mobile devices. When you notice that the crops or plants may get harmed from the weather, you can instantly prevent it.

This topic uses different sensors to detect different weather values, and the detected values can be transmitted to the portable device, so that we can directly detect the current weather values, and then do calculations and inferences with the values. possible climate.

Keywords: arduino, internet of things, solar energy, ESP32, weather observatory

目錄

第 1 章 前言	1
1-1 專題製作背景及目的	1
1-1-1 製作背景	1
1-1-2 製作目的	1
1-2 預期成果	1
第 2 章 理論探討	2
2-1 硬體	2
2-1-1 74LS148	2
2-1-2 ESP32 (NodeMCU-32s)	2
2-1-4 霍爾感測器	3
2-1-5 攀藤 PMS3003	4
2-1-6 GUVVA-S12SD	4
2-1-7 3D 列印機	4
2-1-8 雷射切割機	5
2-2 軟體	6
2-2-1 Arduino	6
2-2-2 Tinkercad	6
2-2-3 Altium Designer	6
2-2-4 Autodesk Inventor 3D	7
第 3 章 專題設計	7
3-1 系統架構圖	7
3-2 甘特圖	8
3-3 硬體架構	8
3-3-1 方塊圖	8
3-3-2 整體電路圖	9
3-3-3 74LS148 和風向八顆霍爾感測器	10

3-3-4 ESP-32 及其他感測器輸入	10
3-4 軟體架構.....	11
3-5 機構.....	22
3-5-1 風速計	22
3-5-2 風向計	24
3-5-3 百葉箱	25
第 4 章專題成果.....	27
第 5 章結論與建議.....	28
5-1 討論.....	28
5-2 建議.....	28
參考文獻.....	29
附錄.....	29
附錄 1 設備清單.....	30
附錄 2 材料清單.....	30
附錄 3 參與學生簡介.....	31

圖目錄

圖 1 74LS148 接腳圖.....	2
圖 2 74LS148 真值表.....	2
圖 3 ESP32 實體圖.....	3
圖 4 DHT22.....	3
圖 5 霍爾感測器模組.....	4
圖 6 霍爾感測器模組.....	4
圖 7 GUVA-S12SD.....	4
圖 8 3D列印機.....	5
圖 9 雷射切割機.....	6
圖 10 Arduino logo.....	6
圖 11 Tinkercad logo.....	6
圖 12 Altium Designer logo.....	7
圖 13 Autodesk Inventor 3D logo.....	7
圖 14 系統架構圖.....	8
圖 15 甘特圖.....	8
圖 16 硬體架構方塊圖.....	9
圖 17 整體電路圖.....	9
圖 18 硬體PCB layout.....	9
圖 19 硬體PCB 3D.....	10
圖 20 74LS148 及風向電路圖.....	10
圖 21 ESP-32 及其他感測器輸入電路圖.....	11
圖 22 風速計 3D設計圖.....	22
圖 23 風速計爆炸圖.....	23
圖 24 風速計實體照.....	23
圖 25 風向計 3D設計圖.....	24
圖 26 風向計爆炸圖.....	25
圖 27 風向計實體照.....	25
圖 28 百葉箱 3D設計圖.....	26
圖 29 百葉箱實體照.....	26
圖 30 LINE Notify顯示畫面.....	27
圖 31 整體外觀.....	28

第 1 章 前言

1-1 專題製作背景及目的

1-1-1 製作背景

在農業中，許多人需要一些關於天候氣象的數值，來確保自農作物環境。

1-1-2 製作目的

為了避免氣候對農業的損害，所以製作本專題 IOT 小型氣象觀測站，來偵測不同天氣數值，並且傳送資料給使用者。

1-2 預期成果

本專題預期完成一個小型氣象觀測站，能夠利用 LINE NOTIFY 來傳輸所感測到的數據，可顯示當前的風向、風速、PM2.5 濃度、溫度、濕度、紫外線強度、並且使用太陽能蓄電池供電可不用外接電源。

第 2 章 理論探討

2-1 硬體

2-1-1 74LS148

74LS148 是 8 線—3 線優先編碼器，將 8 條數據線 (0—7) 進行 3 線(4-2-1) 二進制 (八進制) 優先編碼，即對最高位數據線進行解碼。利用選通端 (EI) 和輸出選通端 (EO) 可進行八進制擴展。

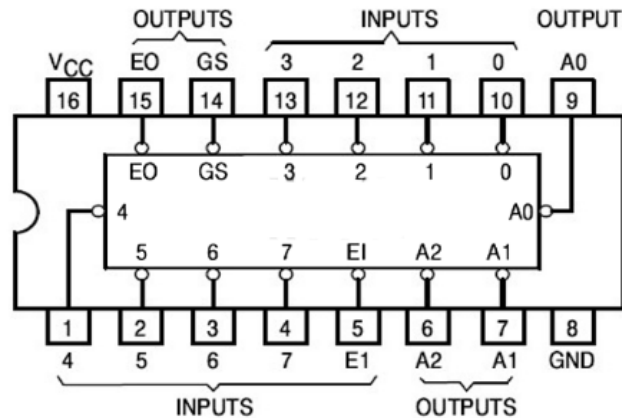


圖 1 74LS148 接腳圖

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

H = high logic level, L = low logic level, X = irrelevant

圖 2 74LS148 真值表

2-1-2 ESP32 (NodeMCU-32s)

ESP32 是一系列由中國上海樂鑫科技(Espressif)推出的晶片、模組、開發板，採用 Tensilica Xtensa LX6 微處理器，包括雙核心和單核變體，支援 Arduino 開放性架構，內建 WiFi 及低功率藍芽 BLE，可用腳位也擴增到約 26 根，大部分腳位同時支援類比及數位，內建霍爾、溫度、觸控感測器，主頻高達 240MHz 的雙核心 CPU，傳輸界面 SPI、I2C、UART 等強大的功能。



圖 3 ESP32 實體圖

2-1-3 DHT22

DHT22 (又稱 AM2302) 濕敏電容數位溫濕度模組是一款含有已校準數位信號輸出的溫濕度複合感測器。它應用專用的數位模組採集技術和溫濕度傳感技術，確保產品具有極高的可靠性與卓越的長期穩定性。感測器包括一個電容式感濕元件和一個高精度測溫元件，並與一個高性能 8 位單片機相連接。



圖 4 DHT22

2-1-4 霍爾感測器

霍爾磁性感測器模組也稱霍爾傳感器，是一個換能器，將變化的磁場轉化為輸出電壓的變化。霍爾傳感器首先是用來測磁場的，此外還可以被用來測量產生和影響磁場的物理量，例如，位置測量，轉速測量，和電流測量設備。

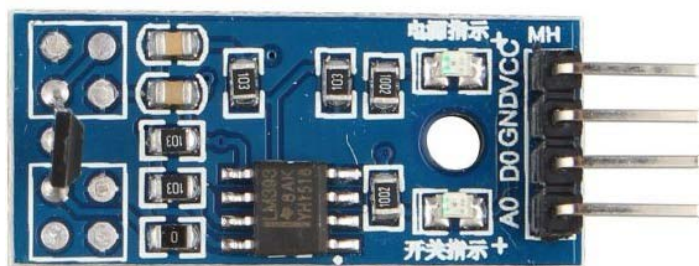


圖 5 霍爾感測器模組

2-1-5 攀藤 PMS3003

PMS3003 是一款數字式通用顆粒物濃度傳感器，可以用於獲得單位體積內空氣中懸浮顆粒物的質量，即顆粒物濃度，並以數字接口形式輸出。本傳感器可嵌入各種與空氣中懸浮顆粒物濃度相關的儀器儀表或環境改善設備，為其提供及時準確的濃度數據。本傳感器採用激光散射原理。即令激光照射在空氣中的懸浮顆粒物上產生散射，同時在某一特定角度收集散射光，得到散射光強隨時間變化的曲線。進而微處理器利用基於米氏（MIE）理論的算法，得出顆粒物的等效粒徑及單位體積內不同粒徑的顆粒物數量。



圖 6 霍爾感測器模組

2-1-6 GUVA-S12SD

GUVA-S12SD UV 感測器是適合於檢測在太陽光的 UV 輻射。它可以用於任何需要監視 UV 光量的應用，並且可以很容易地連接到任何微控制器。GUVA-S12SD 太陽紫外線強度感測器 UV 檢測波長為 200 – 370nm 輸出校準的模擬電壓，隨 UV 光強度而變化，因此，基本上所有您需要做的是將 GUVA-S12SD 其連接到 ADC 輸入並讀取該值。



圖 7 GUVA-S12SD

2-1-7 3D 列印機

3D 列印機是根據您要創建的设计模型(3D 模型檔)進行三個維度列印的機器。

了 3D 列印機的列印技術各式各樣，3D 列印機的種類也多種可以選擇，從簡單的可以在家使用的列印機，到用於製造業等的工業用列印機。



圖 8 3D 列印機

2-1-8 雷射切割機

雷射切割是一種可用於切割不同材料厚度的金屬與非金屬材料的切割工藝。經過引導、成型與聚合的雷射束為此奠定基礎。其擊中工件後，對材料進行加熱以使其熔化或蒸發。全部雷射功率集中於直徑通常小於半毫米的一點。若在該位置輸入的熱量多於因熱傳導而流失的熱量，則雷射束將完全穿透材料——切割過程開始。在其他工藝中重型工具向板材施加巨大外力，而雷射束是無接觸地完成其工作。因此，工具本身不會磨損，工件上也不會產生變形或損壞。



圖 9 雷射切割機

2-2 軟體

2-2-1 Arduino

Arduino 專案始於 2005 年，作為義大利伊夫雷亞地區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，以建立使用感測器與環境相互作用的裝置執行器。適用於初學者愛好者的此類裝置的常見範例包括感測器、簡單機器人、恆溫器和運動檢測器。



圖 10 Arduino logo

2-2-2 Tinkercad

Tinkercad 是一套免付費的雲端建模軟體，它將原本複雜的 3D 建模過程簡單化，並提供多種常用的幾何圖形元件，就算是剛進入 3D 列印的使用者，也能快速上手，成為 3D 列印建模達人！適用對象：想加入 3D 列印鑑模的設計者。不需要特別的電腦背景，只要有興趣，就能在最短時間內，學會如何建模。



圖 11 Tinkercad logo

2-2-3 Altium Designer

Altium designer，簡稱 AD，是一款電子設計自動化軟體，用於電路原理圖、PCB、FPGA 設計，由在澳大利亞的軟體公司 Altium Ltd.開發。它結合了板級設計與 FPGA 設計。收購來的 PCAD 及 TASKING 成為了 altium designer 的一部分。在高速電路板設計領域，它可進行差分對布線。



圖 12 Altium Designer logo

2-2-4 Autodesk Inventor 3D

Autodesk Inventor 產品為 3D 機械設計、產品模擬、模具建立與設計溝通提供一套彈性且全方位的軟體。Inventor 能讓您設計、視覺化及模擬產品，超越 3D 進化至數位原型製作。Autodesk Inventor 產品符合成本效益且易學，為 3D 機械設計、CAD 生產力、設計溝通、產品模擬、佈線系統與鑄模設計提供一套彈性的軟體，本專題使用其製作 3D 爆炸圖。



圖 13 Autodesk Inventor 3D logo

第 3 章 專題設計

3-1 系統架構圖

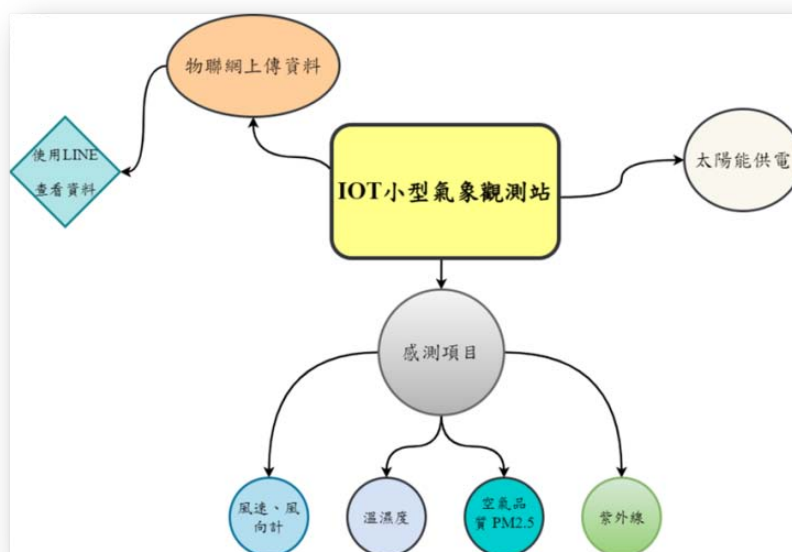


圖 14 系統架構圖

3-2 甘特圖

週次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	負責成員	
工作項目																				
資料查詢	■	■	■			■		■	■	■										全員
理論研究	■	■	■	■			■	■												全員
材料購買		■	■				■													劉沛勳
硬體設計		■	■	■	■			■	■	■	■	■								徐丞霆、賴霆安
3D 列印及雷切		■	■		■	■	■													徐丞霆、洪聿民
硬體測試					■	■	■	■	■											劉沛勳、徐丞霆
程式設計	■	■	■			■		■	■	■										洪聿民、賴霆安
軟體測試		■	■			■			■	■	■									洪聿民、劉沛勳
物聯網									■	■	■	■	■							洪聿民、賴霆安
整體測試											■	■	■	■						全員
外殼製作													■	■	■					全員
期末報告撰寫					■	■	■	■	■	■			■	■	■	■				全員
口頭報告			■		■			■		■		■		■		■				全員
預定進度	5	10	15	20	25	30	35	40	45	50	60	65	70	75	85	90	95	10		累積百分比%

圖 15 甘特圖

3-3 硬體架構

3-3-1 方塊圖

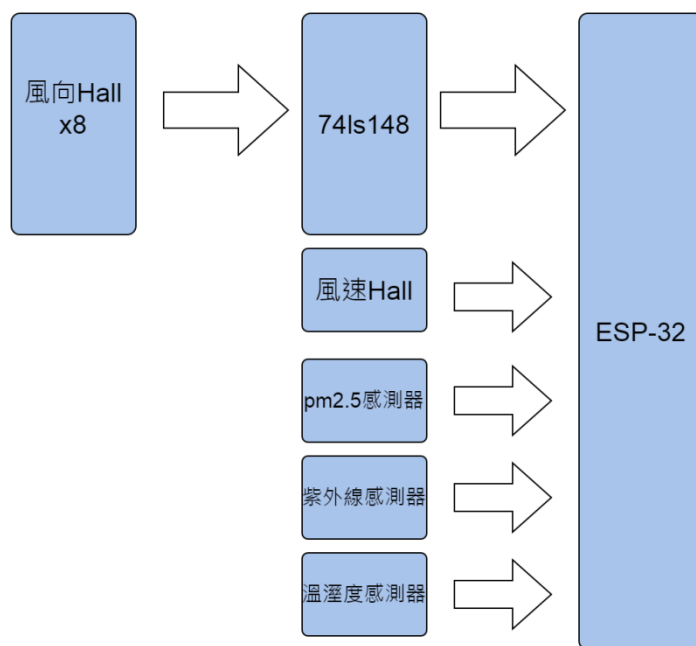


圖 16 硬體架構方塊圖

3-3-2 整體電路圖

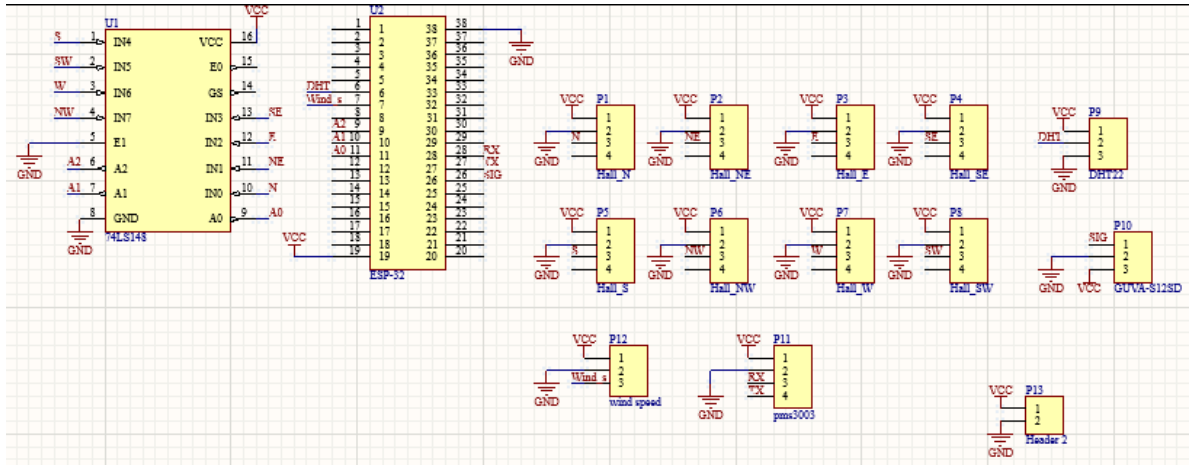


圖 17 整體電路圖

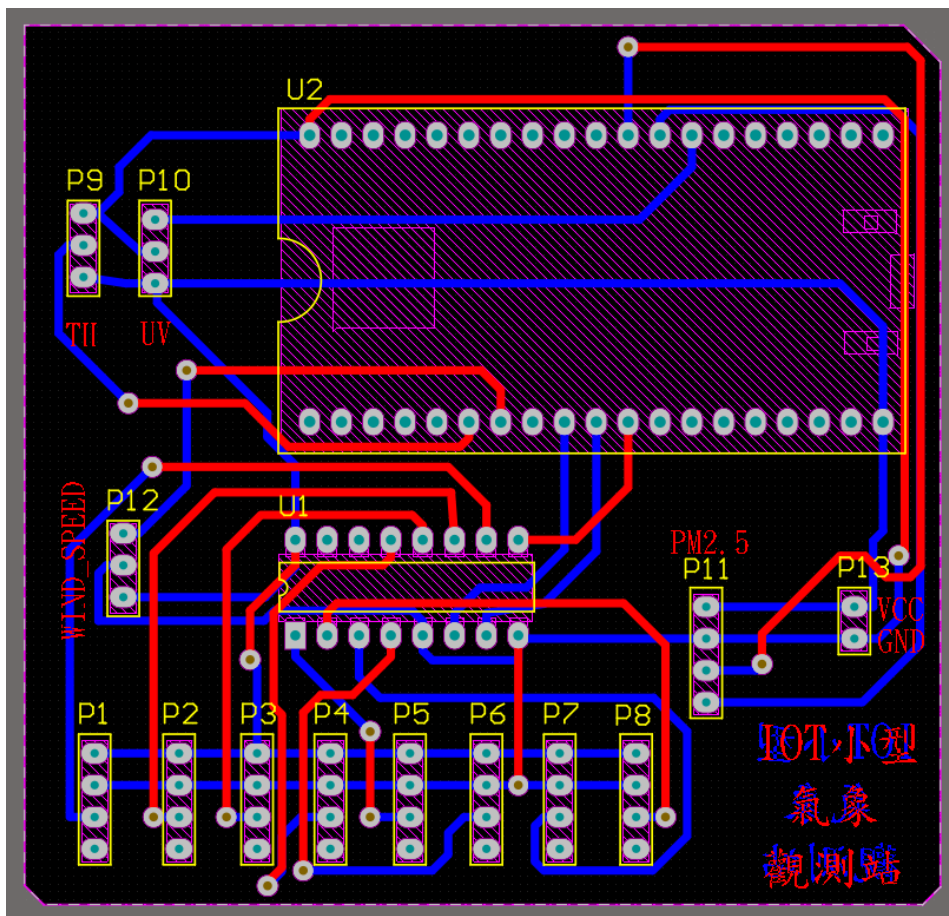


圖 18 硬體 PCB layout

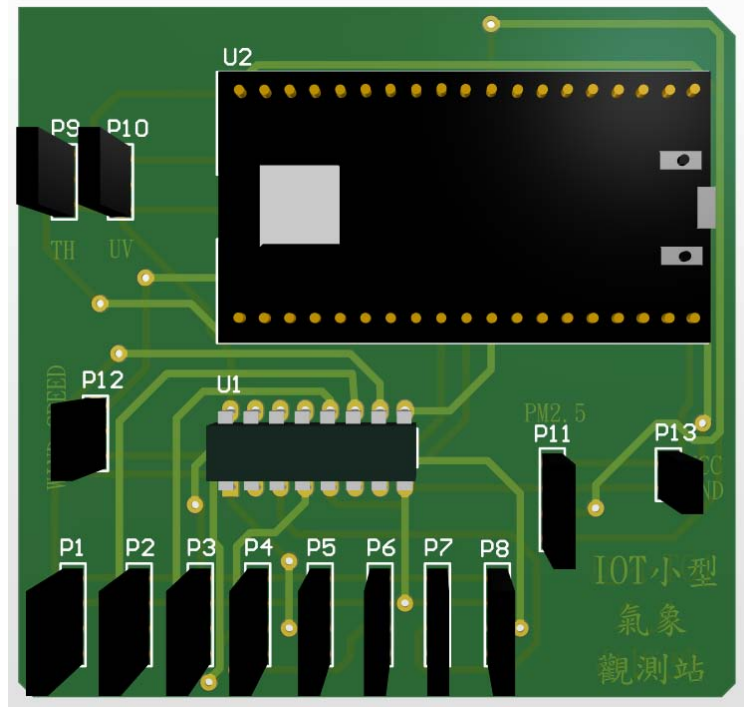


圖 19 硬體 PCB 3D

3-3-3 74LS148 和風向八顆霍爾感測器

因為八個不同輸入會使硬體電路太複雜且軟體編寫上也較繁瑣，所以我們使用 74LS148，8 對 3 優先編碼器將原本的 8 個輸出編碼為 3 位二進制碼。

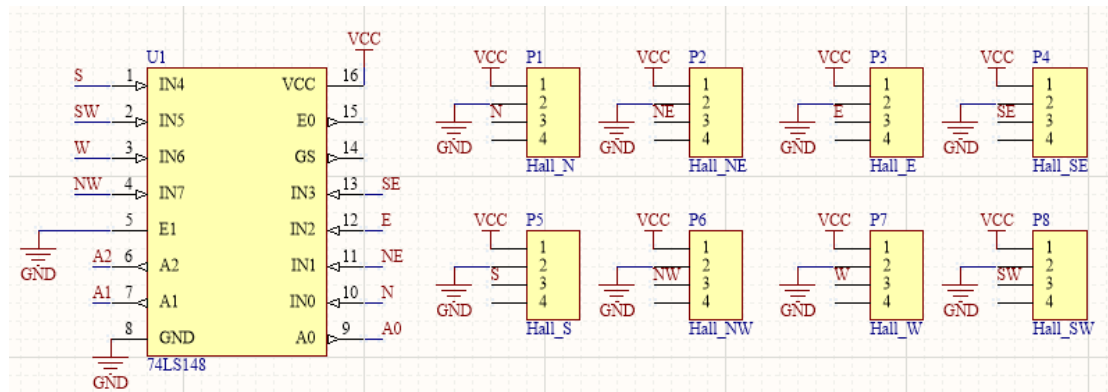


圖 20 74LS148 及風向電路圖

3-3-4 ESP-32 及其他感測器輸入

溫溼度感測器輸出接腳(DHT)，紫外線感測器輸出接腳(SIG)，PM2.5 感測器使用 TX、RX，74LS148 輸出(A0、A1、A2)，風速一隻輸入，共八個輸入至 ESP-32。

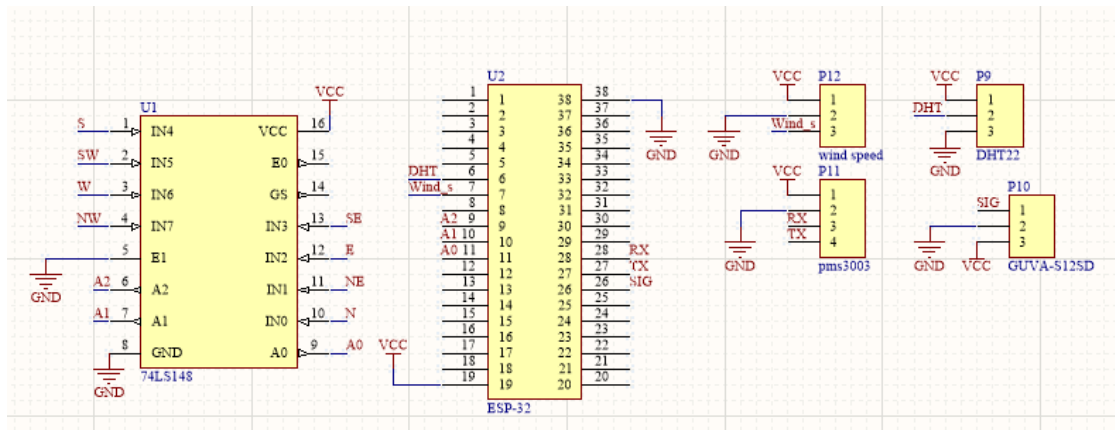


圖 21 ESP-32 及其他感測器輸入電路圖

3-4 軟體架構

程式使用的函示庫

```
#include "DHT.h"
```

```
#include <WiFi.h>
```

```
#include <WiFiClientSecure.h>
```

```
#include <HTTPClient.h>
```

```
#include "PMS.h"
```

```
WiFiClientSecure client
```

全域變數設定

```
#define DHTPIN 4 //溫溼度腳位設置
```

```
#define DHTTYPE DHT22
```

```
PMS pms(Serial2);
```

```
PMS::DATA data;
```

```
const char ssid[]="到底什麼閃現"; //修改為你家的 WiFi 網路名稱
```

```
const char pwd[]="88888888"; //修改為你家的 WiFi 密碼
```

```
String Linetoken = "MchAcvxna3VBim73pEVAqgmsAoWQLNBcUWohogNCvQ4";
```

```

char host[] = "notify-api.line.me";//LINE Notify API 網址

String url = "https://api.thingspeak.com/update?api_key=BDH0DJ6OKSNMKPYS";
DHT dht(DHTPIN, DHTTYPE);
int hsp1 = 25;    //74LS148 腳位
int hsp2 = 26;
int hsp3 = 27;
int state1 = 0;  //74LS148 狀態
int state2 = 0;
int state3 = 0;
float windspeed; //宣告風速
const int hal =32; //風速腳位
const int INT_Number = 0; //宣告中斷
volatile int count = 0;
float sensorVoltage;
float sensorValue; //宣告 UV
int wd;
int pm;
int lastime =0;

```

設置風速風向腳位及 ESP-32wifi 設定

```

void setup() {
  Serial.begin(9600);
  Serial2.begin(9600);
  pinMode(hsp1, INPUT);    //風向腳位模式設定
  pinMode(hsp2, INPUT);
  pinMode(hsp3, INPUT);
  pinMode(hal, INPUT);    //風速腳位模式設定
  attachInterrupt(hal, halcount, FALLING);    //風速霍爾產生之電壓負
緣觸發中斷
  WiFi.mode(WIFI_STA); //設置 WiFi 模式
  WiFi.begin(ssid,pwd);

  Serial.print("WiFi connecting");

  //當 WiFi 連線時會回傳 WL_CONNECTED，因此跳出迴圈時代表已成功連線

```

```

while(WiFi.status()!=WL_CONNECTED){
  Serial.print(".");
  delay(500);
}

Serial.println("");
Serial.print("IP 位址:");
Serial.println(WiFi.localIP()); //讀取 IP 位址
Serial.print("WiFi RSSI:");
Serial.println(WiFi.RSSI()); //讀取 WiFi 強度

dht.begin();
client.setInsecure();
}

```

主程式 UV 值計算

```

void loop() {
  sensorValue = analogRead(35);           //UV 腳位設定
  sensorVoltage =sensorValue/1024*5;     //UV 計算
}

```

風向判斷

```

state1 = digitalRead(hsp1);           //
state2 = digitalRead(hsp2);
state3 = digitalRead(hsp3);
// Wait a few seconds between measurements.
if (pms.read(data)){                  //Pm2.5 接收決策訊息
  if (state1 == LOW && state2 == LOW && state3 == LOW){      //74is148 輸出
    狀態判定
    Serial.println("aspect = WN");
    wd = 7;
  }
  else if (state1 == LOW && state2 == LOW && state3 == HIGH) {
    Serial.println("aspect = W");
    wd = 6;
  }
}

```

```

else if (state1 == LOW && state2 == HIGH && state3 == LOW){
    Serial.println("aspect = SW");
    wd = 5;
}
else if (state1 == LOW && state2 == HIGH && state3 == HIGH){
    Serial.println("aspect = S");
    wd = 4;
}
else if (state1 == HIGH && state2 == LOW && state3 == LOW){
    Serial.println("aspect = ES");
    wd = 3;
}
else if (state1 == HIGH && state2 == LOW && state3 == HIGH){
    Serial.println("aspect = E");
    wd = 2;
}
else if (state1 == HIGH && state2 == HIGH && state3 == LOW){
    Serial.println("aspect = EN");
    wd = 1;
}
else if (state1 == HIGH && state2 == HIGH && state3 == HIGH){
    Serial.println("aspect = N");
    wd = 0;
}

```

風速計算

```

windspeed = (14*3.14159)*count/1/100; //風速計算公式
Serial.print("wind speed :");
Serial.print(windspeed);
Serial.println(" m/s");
count = 0;

```

各項數值數值輸出到 serial.print

// Reading temperature or humidity takes about 250 milliseconds!

```

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();

Serial.print(F("Humidity: "));
Serial.println(h);
Serial.print(F("Temperature: "));
Serial.println(t);
Serial.print("UV = ");
Serial.print(sensorVoltage);
Serial.println(" V");

Serial.print("PM 1.0 (ug/m3): ");
Serial.println(data.PM_AE_UG_1_0);

Serial.print("PM 2.5 (ug/m3): ");
Serial.println(data.PM_AE_UG_2_5);
pm = data.PM_AE_UG_2_5;

Serial.print("PM 10.0 (ug/m3): ");
Serial.println(data.PM_AE_UG_10_0);

Serial.println();

UploadData();

```

每 6 小時上傳資料至 LINE 即發生異常氣候立即通知

```

if((millis()- lastime) > 3600000){           //每六個小時執行一次
    lastime += 3600000;
    UploadLine();           //上傳資料至 LINE
    WsEmergency();         //風速異常
    PmEmergency();         //PM2.5 異常
    UvEmergency();         //UV 值異常
    TemperEmergency();     //溫度異常

```

```
}  
}
```

風速計數

```
void halcount() { //風速計數  
  count = count + 1;  
}
```

將資料上傳 ThinkSpeak

```
//將資料上傳 ThinkSpeak  
void UploadData(){  
  float h = dht.readHumidity();  
  // Read temperature as Celsius (the default)  
  float t = dht.readTemperature();  
  
  //建立一個網頁  
  HTTPClient http;  
  
  //把所有數值數值寫入網址  
  String url1=  
url+"&field1="+wd+"&field2="+windspeed+"&field3="+t+"&field4="+h+"&field5="+senso  
rVoltage+"&field6="+pm;  
  //控制資料輸出變數  
  //提交網址  
  http.begin(url1);  
  
  //讀取網頁內容  
  int httpCode = http.GET();  
  if(httpCode == HTTP_CODE_OK){
```

```

String payload = http.getString();
Serial.print("網頁內容=");
Serial.println(payload);
}
else{
    Serial.println("網路傳送失敗");
}

//關閉網頁
http.end();
}

```

將資料上傳 LINE

```

void UploadLine(){
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    String message;
    message += "\n 溫度=" + String(((float)t) + " *C";
    message += "\n 濕度=" + String(((float)h) + " H";
    message += "\n 風速=" + String(((float)windspeed) + " m/s";
    message += "\n 風向=" + String(((float)wd));
    message += "\n 紫外線=" + String(((float)sensorVoltage) + " UV";
    message += "\npm2.5=" + String(((float)pm) + " ug/m3";
    Serial.println(message);
    if (client.connect(host, 443)) {
        int LEN = message.length();
        //傳遞 POST 表頭 //傳輸至 line
        String url = "/api/notify";
        client.println("POST " + url + " HTTP/1.1");
        client.print("Host: "); client.println(host);
        //權杖
    }
}

```

```

client.print("Authorization: Bearer "); client.println(Linetoken);
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: "); client.println( String((LEN + 8)) );
client.println();
client.print("message="); client.println(message);
client.println();
//等候回應
delay(2000);
String response = client.readString();
//顯示傳遞結果
Serial.println(response);
client.stop(); //斷線，否則只能傳 5 次
}
else {
//傳送失敗
Serial.println("connected fail");
}
}
}

```

緊急傳送溫度異常

```

void TemperEmergency(){
float t = dht.readTemperature();
if(t > 33 || t < 15){
String message = "溫度警報!";
message += "\n 溫度=" + String(((float)t) + " *C");
Serial.println(message);
if (client.connect(host, 443)) {
int LEN = message.length();
//傳遞 POST 表頭
String url = "/api/notify";
client.println("POST " + url + " HTTP/1.1");
client.print("Host: "); client.println(host);
//權杖
client.print("Authorization: Bearer "); client.println(Linetoken);
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: "); client.println( String((LEN + 8)) );

```



```

client.println();
client.print("message="); client.println(message);
client.println();
//等候回應
delay(2000);
String response = client.readString();
//顯示傳遞結果
Serial.println(response);
client.stop(); //斷線，否則只能傳 5 次
}
else {
//傳送失敗
Serial.println("connected fail");
}
}

```

緊急傳送 PM2.5 值異常

```

void PmEmergency(){
pm = data.PM_AE_UG_2_5;
if(pm > 54){
String message = "細懸浮微粒過高!";
message += "\npm2.5=" + String(((float)pm)) + " ug/m3";
Serial.println(message);
if (client.connect(host, 443)) {
int LEN = message.length();
//傳遞 POST 表頭
String url = "/api/notify";
client.println("POST " + url + " HTTP/1.1");
client.print("Host: "); client.println(host);
//權杖
client.print("Authorization: Bearer "); client.println(Linetoken);
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: "); client.println( String((LEN + 8)) );
client.println();
client.print("message="); client.println(message);
client.println();
//等候回應

```

```

    delay(2000);
    String response = client.readString();
    //顯示傳遞結果
    Serial.println(response);
    client.stop();//斷線，否則只能傳 5 次
  }
else {
  //傳送失敗
  Serial.println("connected fail");
}
}
}

```

緊急傳送 UV 值異常

```

void UvEmergency(){
  sensorValue = analogRead(35);
  sensorVoltage = sensorValue/1024*5;
  if(sensorVoltage> 7){
    String message = "紫外線警報!";
    message += "\n 紫外線=" + String(((float)sensorVoltage)) + " UV";
    Serial.println(message);
    if (client.connect(host, 443)) {
      int LEN = message.length();
      //傳遞 POST 表頭
      String url = "/api/notify";
      client.println("POST " + url + " HTTP/1.1");
      client.print("Host: "); client.println(host);
      //權杖
      client.print("Authorization: Bearer "); client.println(Linetoken);
      client.println("Content-Type: application/x-www-form-urlencoded");
      client.print("Content-Length: "); client.println( String((LEN + 8)) );
      client.println();
      client.print("message="); client.println(message);
      client.println();
      //等候回應
    }
  }
}

```

```

    delay(2000);
    String response = client.readString();
    //顯示傳遞結果
    Serial.println(response);
    client.stop(); //斷線，否則只能傳 5 次
  }
else {
  //傳送失敗
  Serial.println("connected fail");
}
}
}

```

緊急傳送颱風警報

```

void WsEmergency(){
  if(windspeed> 20){
    String message = "颱風警報!";
    message += "\n 風速=" + String(((float)windspeed)) + " m/s";
    Serial.println(message);
    if (client.connect(host, 443)) {
      int LEN = message.length();
      //傳遞 POST 表頭
      String url = "/api/notify";
      client.println("POST " + url + " HTTP/1.1");
      client.print("Host: "); client.println(host);
      //權杖
      client.print("Authorization: Bearer "); client.println(Linetoken);
      client.println("Content-Type: application/x-www-form-urlencoded");
      client.print("Content-Length: "); client.println( String((LEN + 8)) );
      client.println();
      client.print("message="); client.println(message);
      client.println();
      //等候回應
      delay(2000);
      String response = client.readString();
    }
  }
}

```

```
//顯示傳遞結果
Serial.println(response);
client.stop();//斷線，否則只能傳 5 次
}
else {
  //傳送失敗
  Serial.println("connected fail");
}
```

3-5 機構

3-5-1 風速計

使用 1 顆霍爾感測器，1 顆培林，1 根壓克力棒，3 個剖半乒乓球，1 顆磁鐵。

使用 3D 列印自製培林座、防水罩。

使用雷射切割做出下方小圓盤、上方旋轉盤。

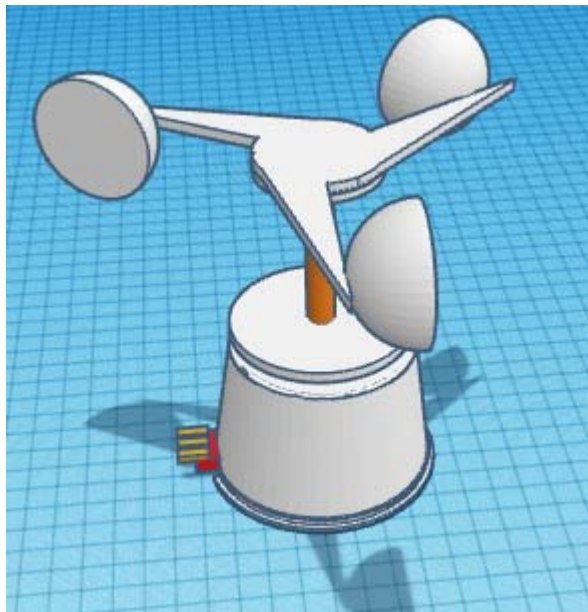


圖 22 風速計 3D 設計圖



圖 23 風速計爆炸圖



圖 24 風速計實體照

3-5-2 風向計

使用 8 顆霍爾感測器，1 顆培林，1 根壓克力棒，1 顆磁鐵。

使用 3D 列印自製培林座、防水罩。

使用雷射切割做出風向標。

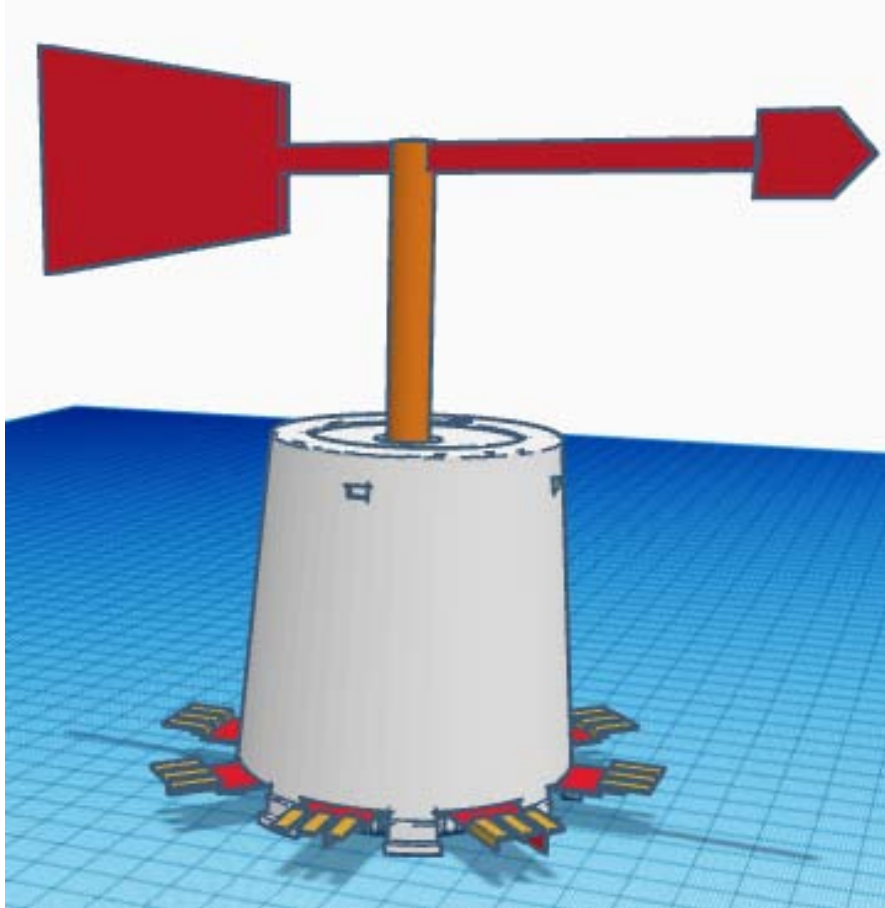


圖 25 風向計 3D 設計圖



圖 26 風向計爆炸圖

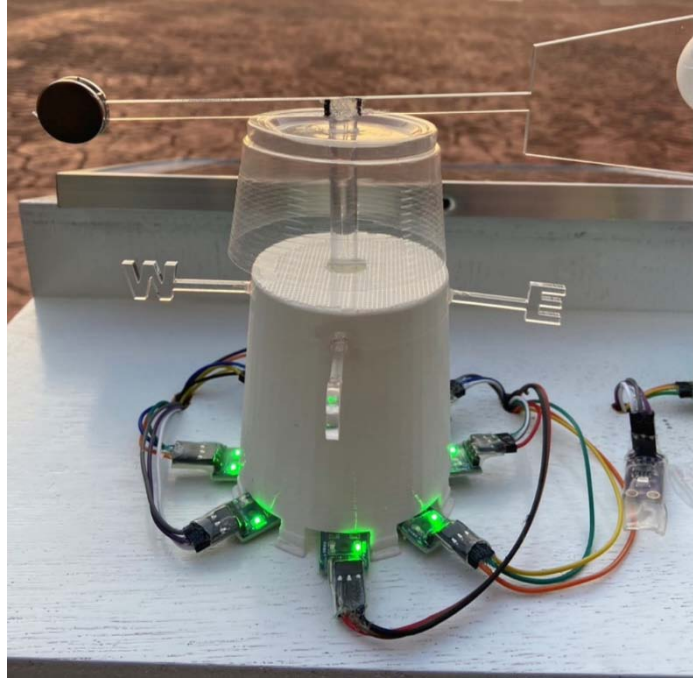


圖 27 風向計實體照

3-5-3 百葉箱

繪製完 3D 圖後交由木匠進行製作



。

圖 28 百葉箱 3D 設計圖



圖 29 百葉箱實體照

第 4 章 專題成果

本專題使用 LINE Notify 來傳送訊息給使用者，每六小時傳送訊息通知，以及當天氣發生異常時傳送緊急通知下圖(圖 30)為方便測試將傳送時間間隔調為 10 秒 1 次。

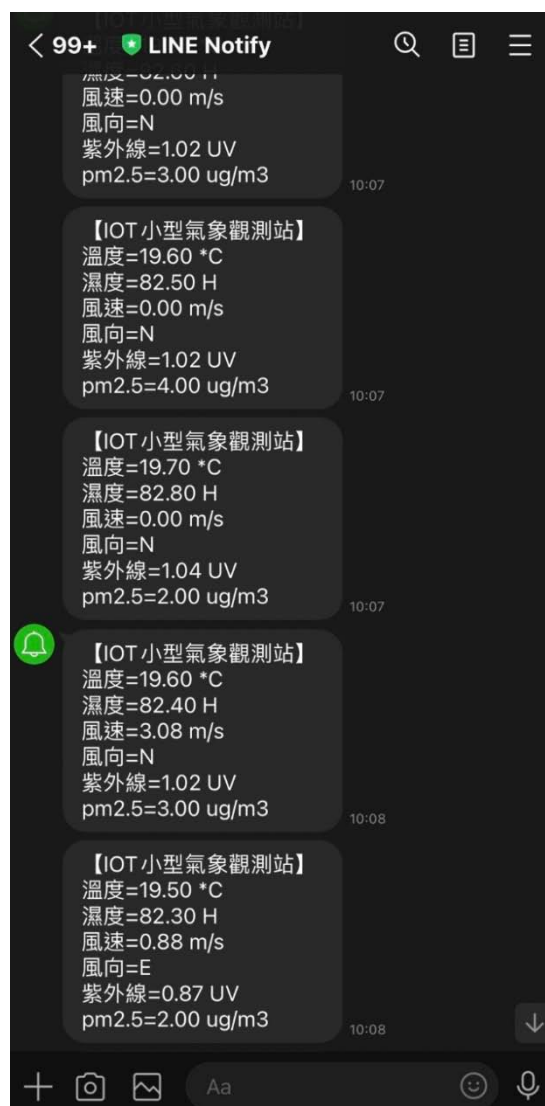


圖 30 LINE Notify 顯示畫面



圖 31 整體外觀

第 5 章 結論與建議

5-1 討論

在製作專題的過程中遇到了許多困難，也有大大小小的爭吵，不過在一起經歷過一切之後，我們成長了很多，也學習到了很多，學到了 IoT 物聯網的運用，繪製 3D 圖利用 3D 列印機印製專題外殼，使用雷射切割機，之前有學過的 pcb 板製作也有運用在這次的專題，在簡報、書面報告也訓練了我們文書處理的能力，而且我們這組還很幸運的在最後的發表會使用全英文報告，也大幅提升我們英文口說的能力。

專題也讓我們有了團隊合作的經驗，有了這次專題的經驗，讓我們以後有類似大型的報告的時候，能夠更容易的完成。

5-2 建議

1. pm2.5 感測器 TX、RX 腳不要插反會無法燒錄程式。

參考文獻

一、

(N.d.). 百科知識. <https://www.jendow.com.tw/wiki/74LS148>

二、

Ej. (2021, September 22). *ESP32_DAY7 介紹 ESP32*. IT 邦幫忙. <https://ithelp.ithome.com.tw/articles/10269937?sc=iThomeR>

三、

霍爾磁性感測器模組 3144E 開關型 電機測速模組 轉速計數模組 霍爾測速 模 組 . (n.d.).

TIS. <https://www.taiwansensor.com.tw/product/%E9%9C%8D%E7%88%BE%E7%A3%81%E6%80%A7%E6%84%9F%E6%B8%AC%E5%99%A8%E6%A8%A1%E7%B5%84-3144e-%E9%96%8B%E9%97%9C%E5%9E%8B-%E9%9B%BB%E6%A9%9F%E6%B8%AC%E9%80%9F%E6%A8%A1%E7%B5%84/>

四、

攀藤 G3 PMS3003 PM1.0 PM2.5 PM10 濃度及顆粒數 感測器 .

(n.d.). TAIWANIOT. <https://www.taiwaniot.com.tw/product/%E6%94%80%E8%97%A4-g3-pms3003-pm1-0-pm2-5-pm10-%E6%BF%83%E5%BA%A6%E5%8F%8A%E9%A1%86%E7%B2%92%E6%95%B8-%E6%84%9F%E6%B8%AC%E5%99%A8/>

五、

GUVA-S12SD 太陽紫外線強度感測器 偵測紫外線強度 . (n.d.).

TAIWANIOT. <https://www.taiwaniot.com.tw/product/guva-s12sd-%E5%A4%AA%E9%99%BD%E7%B4%AB%E5%A4%96%E7%B7%9A%E5%BC%B7%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8-%E5%81%B5%E6%B8%AC%E7%B4%AB%E5%A4%96%E7%B7%9A%E5%BC%B7%E5%BA%A6/>

六、

Arduino. (n.d.). 維基百科. <https://zh.m.wikipedia.org/zh-tw/Arduino>

七、

Vedat Ozan Oner(2020)。 *ESP32 物聯網專題製作實戰寶典*。臺灣：基峰資訊。

八、

林家德(2020)。 *單晶片微處理機實習 全(乙版) Arduino Uno 使用 iPOE M3 多功能實驗板*。臺灣：台科大圖書。

附錄

附錄 1 設備清單


類別	設備、軟體名稱	應用說明
軟體	Arduino	撰寫程式
軟體	Altium Designer	電路設計
軟體	Laser box	繪製雷射切割圖
軟體	MS office	報告製作
軟體	Tinkercad	製作 3D 模型
軟體	Autodesk Inventor 3D	製作爆炸圖
硬體	激光寶盒	雷射切割
硬體	個人電腦	簡報書面靠告製作
硬體	3D 列印機	3D 列印

附錄 2 材料清單

類別名稱	材料名稱	單位	數量	應用說明	備註
微處理機	ESP32 (NodeMCU-32s)	個	1	微處理機	
線材	USB 傳輸線	條	1	傳輸 Arduino 程式	
線材	杜邦線	條	數條	連接電路傳輸信號	
感測器	霍爾感測器	個	9	感測磁力運用在風向及 風速	
感測器	溫溼度感測器 DHT22	個	1	感測溫溼度	

感測器	紫外線感測器 GUVA-S12SD	個	1	感測紫外線	
感測器	PM2.5 感測器	個	1	感測 PM2.5	
編碼器	74LS148	個	1	編碼	
模組	太陽能鋰電池充 電模組	個	1	太陽能鋰電池充電模組	
耗材	壓克力板	片	數片	外殼及風速風向使用	
耗材	電池盒	盒	1	裝電池	
耗材	鋰電池 18650	顆	2	儲存電能	
耗材	太陽能板	片	1	將太陽能轉換成電能	
箱子	百葉箱	個	1	將所有元放入或至於上 方	訂做

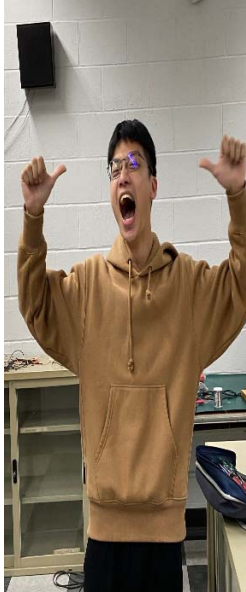
附錄 3 參與學生簡介

姓名	洪聿民	班級	電子三乙	
曾修習 專業科目	1.基本電學(含實習) 2.電子學(含實習) 3.數位邏輯 4.可程式邏輯實習 5.微處理機 6.單晶片微處理器實習 7.電腦輔助設計實習 8.行動裝置應用實習 9.電子電路實習 10.程式設計實習			

參與專題 工作項目	1.3D 列印、雷射切割 2.軟體測試 3.程式設計 4.專題簡報設計
經歷簡介	曾擔任大安吉他社社長 曾擔任一學期班長 曾擔任國文小老師 工業電子丙級技術士證照

姓名	徐丞霆	班級	電子三乙	
會修習 專業科目	1.基本電學(含實習) 2.電子學(含實習) 3.數位邏輯 4.可程式邏輯實習 5.微處理機 6.單晶片微處理器實習 7.電腦輔助設計實習 8.行動裝置應用實習 9.電子電路實習 10.程式設計實習			
參與專題 工作項目	1.硬體測試 2.3D 列印、雷射切割 3.材料購買 4.機構設計			

經歷簡介	工業電子丙級技術士證照			
姓名	劉沛勳	班級	電子三乙	
會修習 專業科目	<ol style="list-style-type: none"> 1.基本電學(含實習) 2.電子學(含實習) 3.數位邏輯 4.可程式邏輯實習 5.微處理機 6.單晶片微處理器實習 7.電腦輔助設計實習 8.行動裝置應用實習 9.電子電路實習 10.程式設計實習 			
參與專題 工作項目	<ol style="list-style-type: none"> 1.材料購買 2.軟、硬體測試 3.硬體設計 4.書面告製作 			
經歷簡介	曾擔任風紀股長 曾擔任大安熱音活動 工業電子丙級技術士證照			

姓名	賴霆安	班級	電子三乙	
曾修習 專業科目	1.基本電學(含實習) 2.電子學(含實習) 3.數位邏輯 4.可程式邏輯實習 5.微處理機 6.單晶片微處理器實習 7.電腦輔助設計實習 8.行動裝置應用實習 9.電子電路實習 10.程式設計實習			
參與專題 工作項目	1.軟體測試 2.3D 列印、雷射切割 3.材料購買 4.專題簡報設計			
經歷簡介	曾擔任物理小老師 工業電子丙級技術士證照			