

112-2 數位電子學實務期末專題

低成本空氣品質偵測器

第八組

物理二 B11202006 陳佑軒

物理四 B09202068 劉雨恩

摘要

隨著空氣污染問題日益嚴重，而為了引起對空氣品質的重視，本項目旨在設計並實作一個低成本的PM2.5偵測器，幫助民眾實時監測空氣品質。

這份報告描述了一個低成本的空氣品質（煙霧）偵測器的設計與實作。偵測器使用 ESP32 作為主控板，透過 HTU21D 感測器測量空氣濕度和溫度，並利用 ESP32 抓取網路上最近空氣品質測站的實時數據顯示於 LCD1602 上。此外，偵測器使用伺服馬達（SG 90）來控制光敏電阻的角度，從 0 度到 180 度每隔 10 度進行一次散射訊號測量，這種不同角度的散射訊號量測在現有研究中尚未見到。

雖未完全成功，但觀察到煙霧通過時訊號受散射影響，驗證了 Rayleigh 散射理論的應用潛力。這種低成本設計的優點在於其易於實現和普及，特別適合於資源有限的地區進行空氣品質監測。

1. 背景介紹(WHY):

隨著全球工業化進程的不斷推進，空氣品質問題日益嚴峻。根據統計數據顯示，去年全球僅有七個國家符合聯合國對 PM2.5 濃度的標準。而台灣在冬季尤為明顯，常見嚴重的霧霾問題。

目前，我們主要依靠氣象局提供的空氣品質指數來決定日常活動，例如是否進行戶外運動。然而，經常出現空氣品質指數顯示良好，但實際觀察卻霧茫茫的情況。通過研究，我們發現板橋區僅有一個空氣品質監測點，這表明氣象局發布的空氣品質數據在空間解析度上存在顯著不足。這一發現與 Badura 等人的研究結果一致，他們的研究強調了目前監測點布局的不足。

這種監測不足的問題在世界上許多未開發地區更加嚴重，這些地區由於缺乏必要的監測設備，無法準確評估空氣品質狀況。因此，我們希望通過此項目，改進空氣品質監測的覆蓋範圍和精度，以便更好地了解 and 應對空氣污染問題。

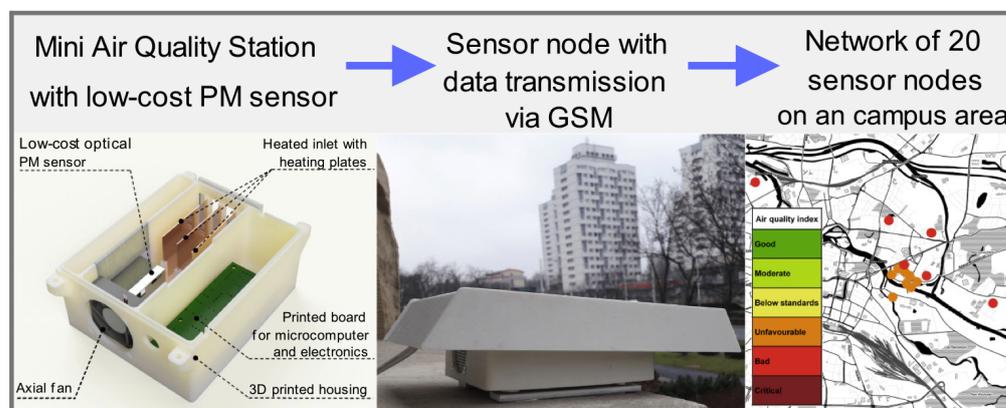
2. 研究目的(HOW):

近年來，“Dense Sensor Network”這一概念受到了廣泛關注，其核心在於通過高密度部署低成本空氣品質偵測器來提高空氣品質監測的覆蓋率和精度。例如，Badura等人的研究證實了高密度部署空氣品質偵測器的重要性，並展示了偵測器密度對數據精準度的影響。

我們目前面臨的挑戰主要在於偵測器部署密度不足，特別是在發展中國家和地區，這些地方缺乏必要的監測設備，無法及時獲取空氣品質數據。針對這一問題，我們的研究目標是設計並製作低成本且高精度的空氣品質偵測器，這些偵測器能夠大量生產並在重點區域高密度部署。

Badura等人的研究：

這篇論旨在評估密集感測器在大學校園區域監測PM2.5的有效性。作者們在波蘭弗羅茨瓦夫科技大學的校園內部署了20個低成本光學PM2.5感測器節點，以提高空氣質量測量的時空解析度。



使用低成本光學傳感器（PMS A003）構建迷你空氣質量站（MAQS）收集PM2.5濃度數據。研究發現PM2.5濃度在校園內部和周邊區域存在顯著的空間變異性，而且在溫度逆溫條件下出現分層現象，建築物上部的PM2.5濃度比地面高出2-4倍。

3. 實驗方法(WHAT):

我們的目標是製作一個低成本的空氣品質偵測器，用以測量儀器周圍的空氣品質。我們設計了一個裝置，利用光敏電阻測量散射光的強度，並顯示空氣品質指數（AQI）和PM2.5 濃度。

a. 外殼設計：

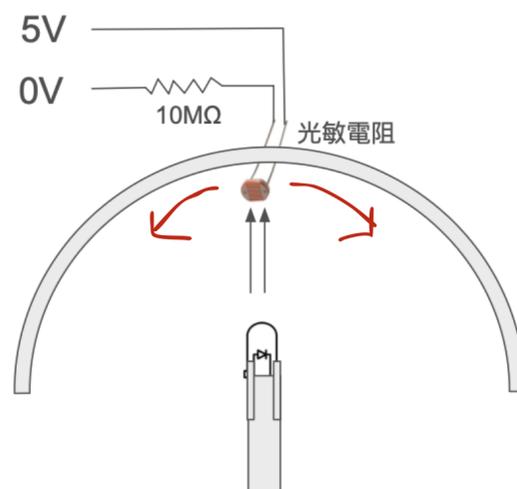
- 使用3D列印技術印一個黑色的10cm x 10cm x 10cm的外殼。
- 中間會有分隔設計，下層放電路及電池，上層是測量光散射訊號的地方。
- 外殼的左側和右側各設置一個風扇，分別用於進氣和排氣，促進氣體交換。
- 前方設置一個LCD1602顯示器，顯示溫濕度和空氣品質數據。

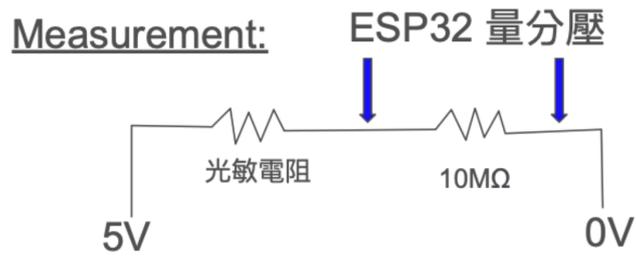


b. 光源與散射測量：

- 使用一種顏色的LED燈作為光源，固定在90度位置（正對光敏電阻）。
- 光敏電阻用於接收散射光訊號。
- 使用SG90伺服馬達來控制光敏電阻的角度，從0度到180度，每10度變化一次，每隔10秒讀取一次數據。

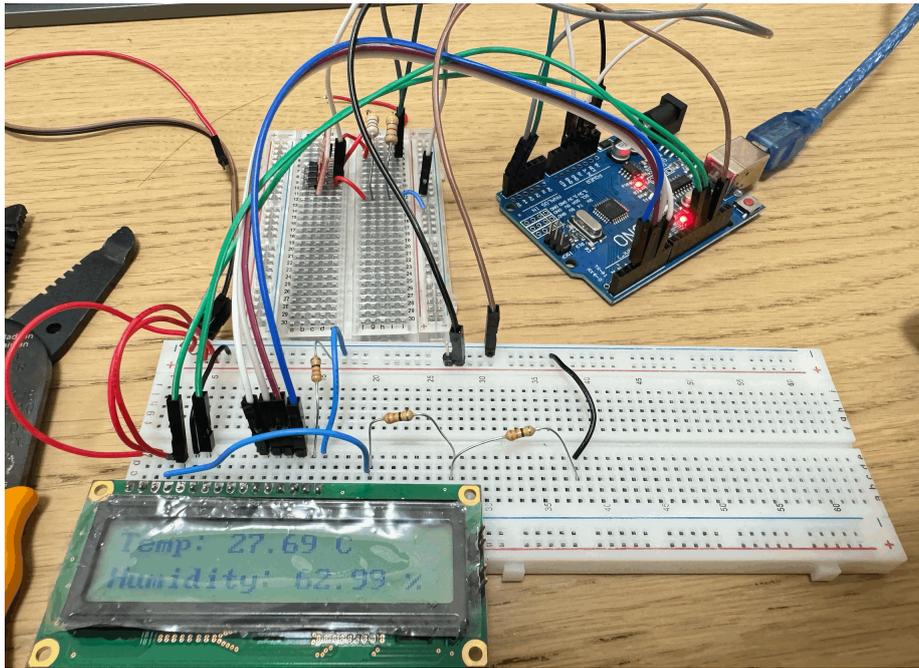
這是我們裝置的示意圖：



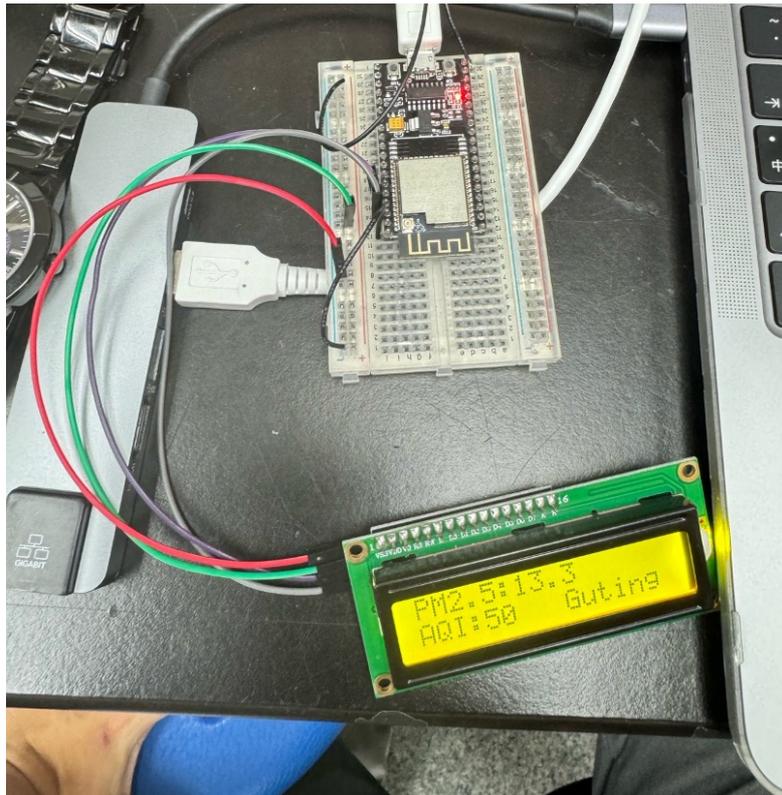


c. 數據讀取與顯示：

- HTU21D感測器測量空氣的溫度和濕度，數據顯示在LCD1602上。



- 從網路上抓取最近監測站的即時AQI值和PM2.5數據，同樣顯示在LCD1602上。



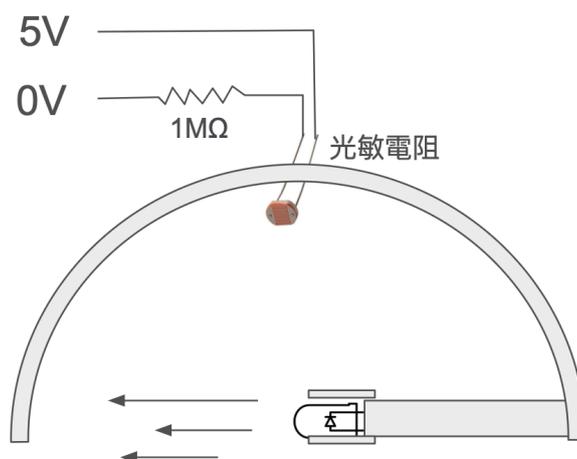
d. 煙霧模擬：

我們使用液態氮或者線香燃燒來產生煙霧，前者是容易取得，隔壁實驗室可以要到，一次可以產生大量煙霧，後者燃燒產生的懸浮微粒、化學成分與真實PM2.5類似，並曾在Nature Communications上的一篇論文作為空氣品質淨化器的測試。

4. 實驗結果:

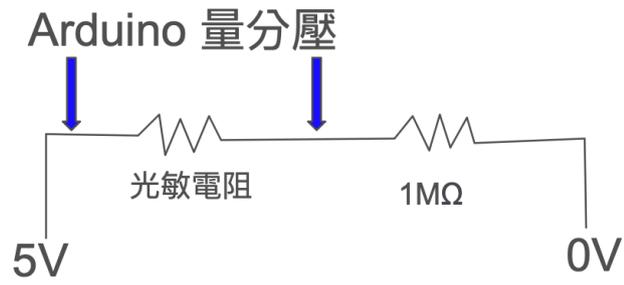
我們期中曾測試過，光敏電阻通過由液態氮產生能作為一個煙霧偵測的成功案例。

當時測試的示意圖：

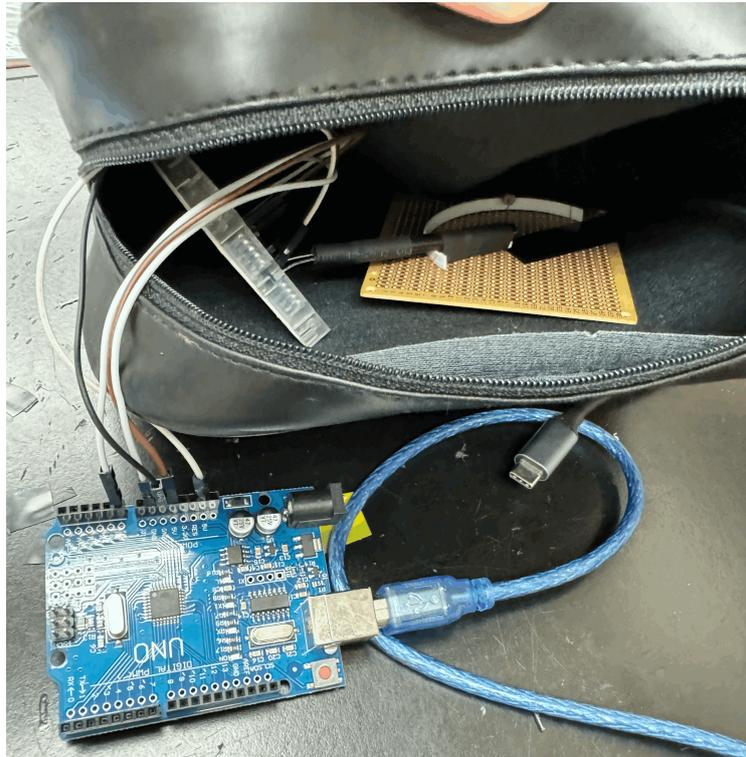


用Arduino UNO板量測光敏電阻分壓：

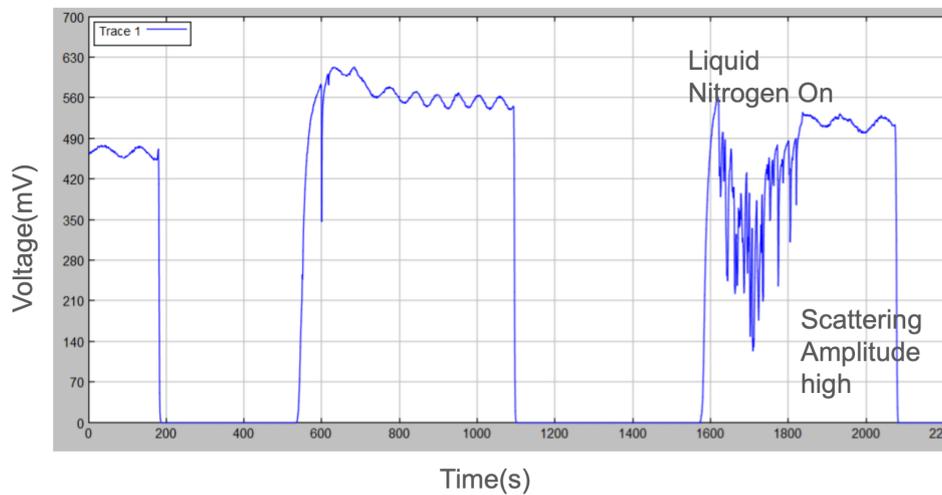
Measurement:



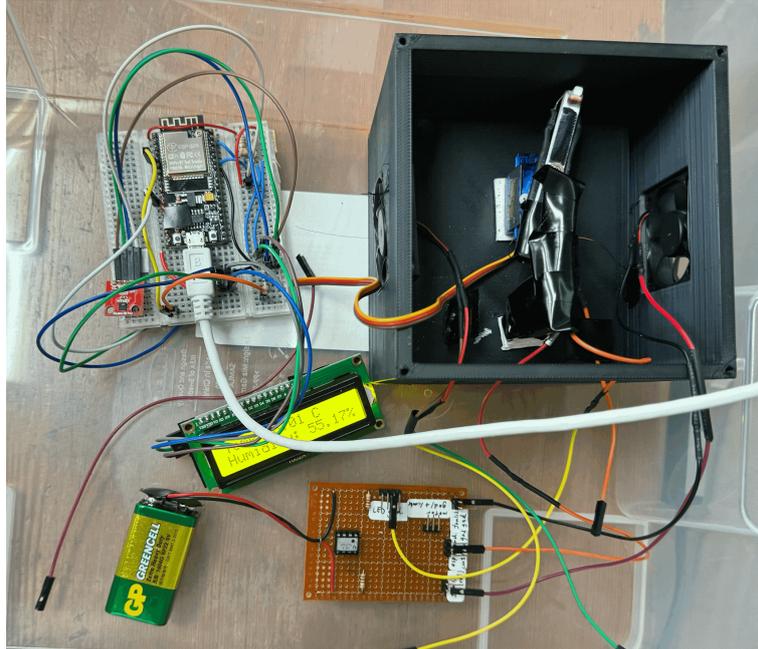
實際環境：將液態氮煙霧通入黑色包包中。



量測到的訊號：通入液態氮後量到的分壓迅速下降。

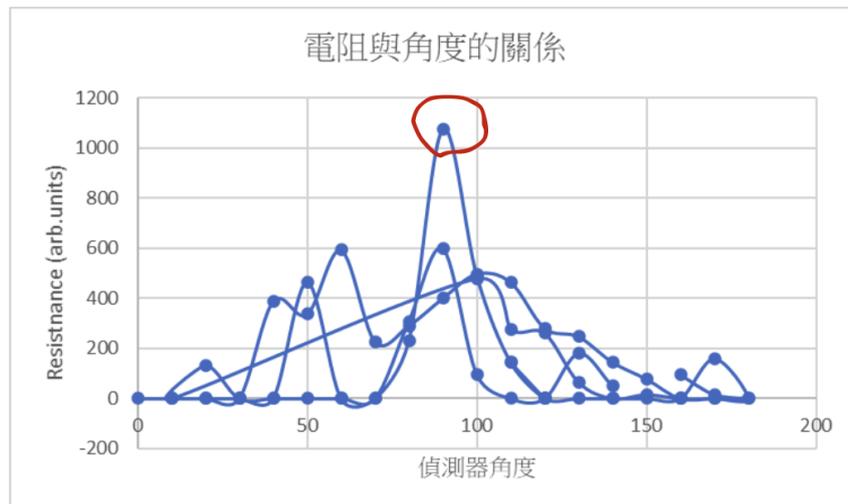


我們接續期中的想法完成，這是我們的成品圖，為demo展示將電路從下層移出來，光敏電阻旋轉請參考影片。

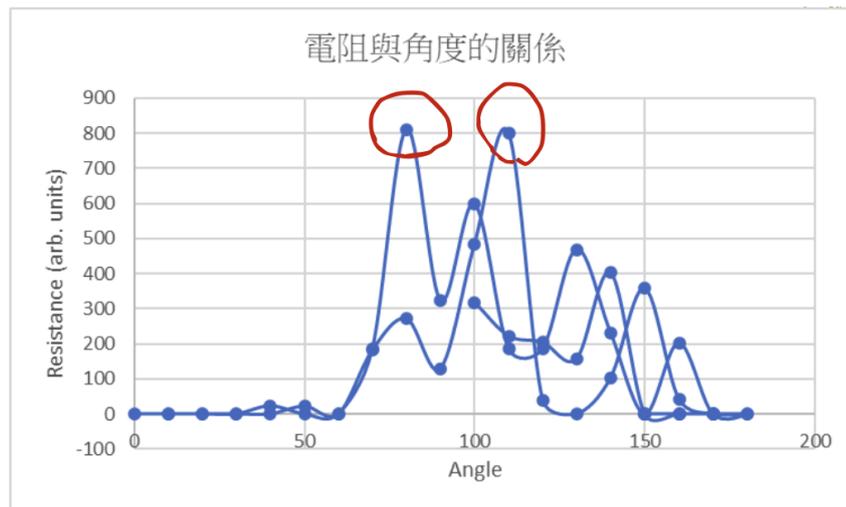


實驗結果顯示，當濃煙通過感測器時，訊號出現明顯波動，這表明散射現象對訊號有影響。儘管未能準確測量PM2.5濃度，但驗證了散射現象的存在。

這張圖是一般環境下量測到的數據，LED在90度正對光敏電阻，ESP32量到最大的訊號。



這是點香後量測到的數據，LED發出的光遇到線香產生的懸浮微粒而產生散射，ESP32在80度和110度的地方量到最大訊號。



5. 成果討論:

在我們開始這個項目時，遇到了許多預料之外的困難，很多地方與我們的預期設想大相徑庭。

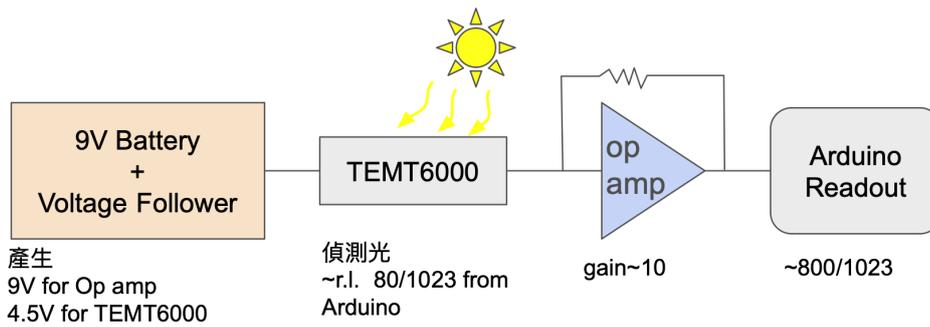
首先，在顯示器選擇方面，我們最初選用了七段顯示器。然而，七段顯示器存在共陽和共陰極的差異，這增加了設計程式碼的複雜性，事實上控制七段顯示器顯示數據所需的程式碼比我們後來使用的LCD1602顯示屏複雜得多。為了解決這個問題，我們從實驗室取用了WG14432B顯示屏。然而，由於這並非主流LCD顯示器，我們在網絡上找不到參考資料，因此花費了大量時間進行研究和測試，最終才成功顯示數據。但是，當我們嘗試切換顯示內容時又遇到新的問題，最後我們還是選用了LCD1602顯示器。

其次，我們在選擇和使用馬達方面也面臨挑戰。最初，我們使用MG996R馬達，但後來發現其不適合我們的需求，於是更換為SG90（360度）馬達。然而，SG90（360度）馬達也未能完全滿足要求，最終我們轉換為SG90（180度）馬達。在這一過程中，由於我們對各種元件不夠熟悉，導致耗費了大量時間進行研究和調整。

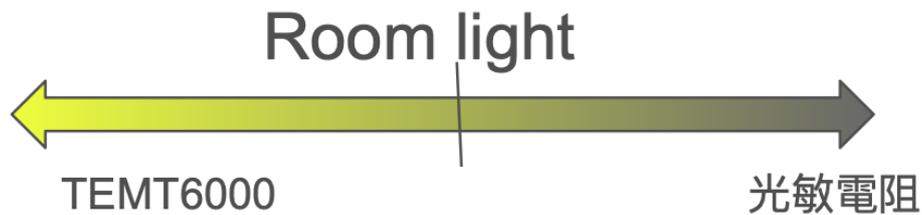
我們發現，許多問題的根源在於我們想要實現“低成本”設計。這一挑戰極大地考驗了我們的技術和知識。我們選擇了便宜的元件進行組裝，但這也帶來了一系列問題：電路焊接不良、焊點脫落、電路無法導通、元件精度不足等。這些問題不僅增加了我們的工作量，也延長了開發的時間。

a. 與初期目標的差異：

- 我們原先打算使用光偵測器TEMT6000讀取散射訊號，不過有幾種缺點，第一是他讀訊號的Range很小，第二是他量到的幾乎都是雜訊（~500/1023）。

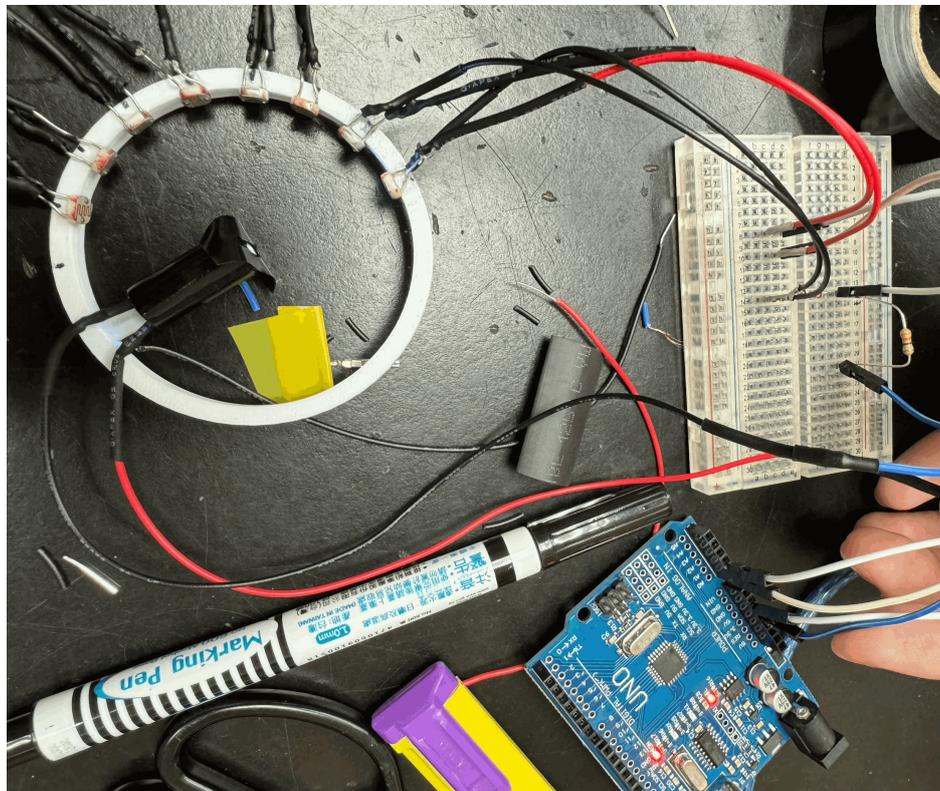


- 經過測試，我們認為光敏電阻比TEMT6000更適合在低於室光的環境，適合在我們的盒子內使用。



- 我們有嘗試使用多個光敏電阻量測在不同角度的訊號，但是使用多個光敏電阻會讓訊號線非常多，而且每一個光敏電阻出廠的內部電阻不一樣，要校正這個不是容易的事，所以我們後來用馬達控制單一個光敏電阻旋轉。

原先的構想設計圖：



- 因為空氣中的懸浮粒子會因為不同波長的光產生不一樣的Rayleigh 散射，所以我們一開始想使用不同顏色的光觀察，最後因為盒子不夠大，使得我們在電路設計上只能使用一個LED。
- 時程規劃比較：
原本安排：

Week 11 (05/03)	初步組裝電路和元件焊接 完成樣品外觀 測試電路功能
Week 12 (05/10)	組裝光路與光學系統的校準 測試光學系統的穩定性
Week 13 (05/17)	編寫微控制器的程式碼 測試軟體功能和與硬體的整合
Week 14 (05/24)	期末專案期中口頭報告: 完成一個初階可運作版本
Week 15 (05/31)	對照我們儀器偵測的數值與 真正的AQI偵測器數值的正確度
Week 16 (06/07)	參考測試結果和同學反饋 對設計進行改進
Week 17 (06/14)	期末專案成果演示

實際：

Week 11 (05/03)	測試溫濕度感測器
Week 12 (05/10)	測試光學系統
Week 13 (05/17)	數據顯示、demo準備
Week 14 (05/24)	期末專案期中口頭報告: 完成一個初階可運作版本
Week 15 (05/31)	焊接電路
Week 16 (06/07)	列印外殼、裝上風扇、馬達，做實際測試
Week 17 (06/14)	期末專案成果演示

- b. 未來展望：要再稍微拓寬盒子，因為現階段的設計空間不足，很多電線容易因拉扯而電路不通。固定LED的支架需要改成3D列印的才夠穩固。這些問題對於光路的穩定十分重要，需要調整完這些後才能根據散射強度對空氣品質的校正。

c. 我們成品的特別之處：不同角度的測量只有我們嘗試過，而且我們挑戰在隨手可得的低成本材料下完成。

6. Q&A:

問題主要是demo影片沒有看清楚。

7. 課程心得:

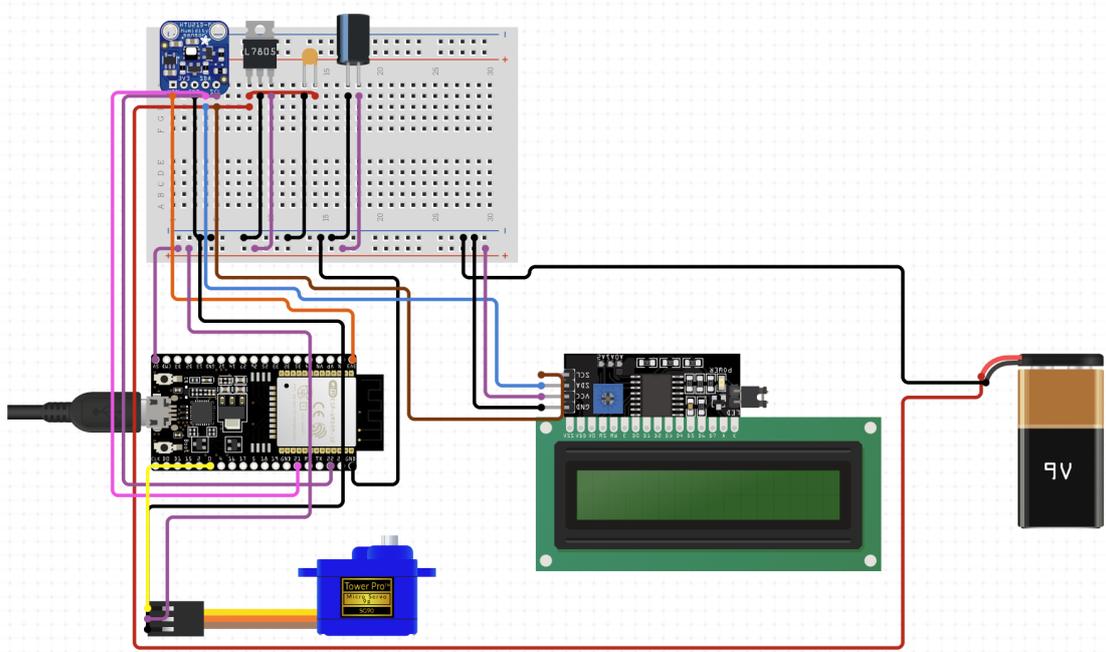
過去沒有嘗試過設計一個東西並製造出來的經驗，在修這門課之前也是僅僅聽過Arduino這個名字而已，在完成專題的過程中學到的就是要怎麼從0開始一步一步獨立解決問題，並透過團隊合作分享自己的知識。

8. 附錄:

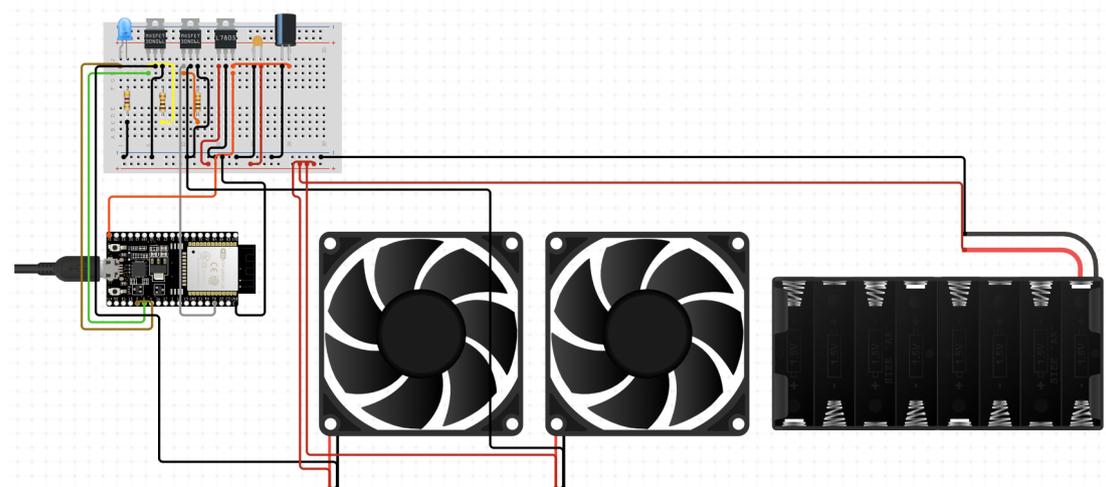
a. 電路圖:

以下是用Circuit.io畫製的電路圖。

第一張電路圖用於控制HTU21D、SG90和LCD1602，需補上光敏電阻一端插在esp32的5V位置，另一端接esp32的第34腳位以接收訊號，並串一個10M Ohm的電阻再接地。



第二張電路圖用於控制風扇和LED，實際上我們採用9V電池做供電，可以不使用到ESP 32。



b. 程式碼:

- 此部分控制LCD顯示。

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <Adafruit_HTU21DF.h>

// 設置 LCD 的 I2C 位址
LiquidCrystal_I2C lcd(0x27, 16, 2);
Adafruit_HTU21DF htu = Adafruit_HTU21DF();

// WiFi 設置
const char* ssid = "iPhone"; //R207
const char* password = "tttttttt"; //33665107

// API URL
const char* serverName = "
https://data.moenv.gov.tw/api/v2/aqx_p_432?format=json&limit=5&api_key=
037a82df-7f8f-4022-8b87-24c74e4f49dd&filters=SiteName,EQ,古亭
";

// 計時器相關變數
unsigned long previousDisplayMillis = 0;
unsigned long previousFetchMillis = 0;
const long displayInterval = 10000; // 30秒
const long fetchInterval = 600000; // 10分
bool showWeather = true; // 鐘
float temperature = 0;
float humidity = 0;
float pm25 = 0;
int aqi = 0;
```

```
void setup() {
  // 初始化 LCD
  lcd.init();
  lcd.backlight();

  // 初始化 HTU21D 感測器
  if (!htu.begin()) {
    while (1); // 停止程式運行，如果 HTU21D 初始化失敗
  }

  // 連接 WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("Connecting...");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi connected");

  // 初次獲取並顯示數據
  fetchData();
  displayData();
}
```

```

void loop() {
    // 獲取當前時間
    unsigned long currentMillis = millis();

    // 每隔30秒切換顯示數據
    if (currentMillis - previousDisplayMillis >= displayInterval) {
        previousDisplayMillis = currentMillis;
        showWeather = showWeather; // 切換顯示模式
        displayData();
    }

    // 每隔10分鐘獲取一次網站數據
    if (currentMillis - previousFetchMillis >= fetchInterval) {
        previousFetchMillis = currentMillis;
        fetchData();
    }

    delay(1000);
}

void fetchData() {
    // 顯示天氣數據
    temperature = htu.readTemperature();
    humidity = htu.readHumidity();

    // 顯示古亭站的資料
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverName);
        int httpStatusCode = http.GET();

        if (httpStatusCode > 0) {
            String payload = http.getString();

            // 解析 JSON 數據
            DynamicJsonDocument doc(2048);
            deserializeJson(doc, payload);

            // 獲取 PM2.5 和 AQI 值
            const char* pm25_str = doc["records"][0]["pm2.5_avg"];
            const char* aqi_str = doc["records"][0]["aqi"];
            pm25 = String(pm25_str).toFloat();
            aqi = String(aqi_str).toInt();
        }

        http.end(); // 關閉 HTTP 連接
    }
}

```

```
void displayData() {
  lcd.clear();
  if (showWeather) {
    lcd.setCursor(0, 0);
    lcd.print("Temp:");
    lcd.print(temperature, 2);
    lcd.print(" ");
      C
    lcd.setCursor(0, 1);
    lcd.print("Humidity: ");
    lcd.print(humidity, 2);
    lcd.print("%");
  } else
    // 格式化 PM2.5 值顯示到小數點後一位
    char pm25_display[6];
    dtostrf(pm25, 4, 1, pm25_display);

    lcd.setCursor(0, 0);
    lcd.print("Guting AQI:");
    lcd.print(aqi);
    lcd.setCursor(0, 1);
    lcd.print("PM2.5:");
    lcd.print(pm25_display);
  }
}
```

- 此部分控制馬達旋轉，並讀數值到監視窗，紀錄數據可另外使用CoolTerm做處理。

```
#include <ESP32Servo.h>

Servo myservo; // 建立一個 servo 物件

int pos = 0;    // 設定 Servo 位置的變數
const int photoPin = 34; // 光敏電阻連接的引腳

void setup() {
  Serial.begin(115200); // 初始化序列通
  myservo.attach(14); // 將 servo 物件連接到 GPIO 14
  myservo.write(0); // 將 servo 位置設置為 0 度
}

void loop() {

  // 正轉 180度，從 0 度旋轉到 180 度，每次 10 度
  for pos = 0; pos < 180; pos += 10) {
    myservo.write(pos); // 告訴 servo 走到 'pos' 的位置
    delay(1000); // 等待 1000ms 讓 servo 走到指定位置
    int lightLevel = analogRead(photoPin); // 讀取光敏電阻的
    Serial.print("Position,"); // 值
    Serial.print(pos);
    Serial.print(",Light Level,");
    Serial.println(lightLevel);
  }

  // 反轉 180度，從 180 度旋轉到 0 度，每次 10 度
  for pos = 180; pos > 0; pos -= 10) {
    myservo.write(pos); // 告訴 servo 走到 'pos' 的位置
    delay(1000); // 等待 1000ms 讓 servo 走到指定位置
    int lightLevel = analogRead(photoPin); // 讀取光敏電阻的
    Serial.print("Position,"); // 值
    Serial.print(pos);
    Serial.print(",Light Level,");
    Serial.println(lightLevel);
  }
}
```

c. 產品花費（只考慮最後成品的使用）：

來源為光華商場的今華電子或實驗室。

名稱	數量	總價
進氣風扇	2	140
溫度濕度感測器	1	80
LED	1	5
光敏電阻	1	10
ESP32	1	350(實驗室)
螢幕 (LCD 1602)	1	85
SG90	1	150(實驗室)
電阻、跳線、麵包 板、電池...	X	實驗室

若包含實驗室的材料總價約莫是在1000元台幣(30美元)。

d. 參考資料:

1. Badura, M., Sówka, I., Szymański, P., Batog, P. (2020). "Assessing the usefulness of dense sensor network for PM2.5 monitoring on an academic campus area." *Science of The Total Environment*, Volume 722, 137867. ISSN 0048-9697. <https://doi.org/10.1016/j.scitotenv.2020.137867>
2. Our World in Data (2024). "Air Pollution." <https://ourworldindata.org/air-pollution#all-charts>
3. PR Newswire (2024). "2023 IQAir World Air Quality Report Finds Only 7 Countries Meet WHO PM2.5 Air Pollution Guideline."
4. YU, T. (2020). "Design of Air Quality Monitoring System based on Light Scattering Sensor." *IOP Earth and Environmental Science*, Volume 647.
5. Liu, C., Hsu, P. C., Lee, H. W., Ye, M., Zheng, G., Liu, N., Li, W., & Cui, Y. (2015). "Title of the Paper." *Nature Communications*, 6, 6205.

6. Ameba IoT (n.d.). "HTU21D." <https://www.amebaiot.com/zh/htu21d/>