中華民國第 60 屆中小學科學展覽會

作品說明書

高級中等學校組　環境學科

(鄉土)教材獎

052607

空汙報你知-守護你我檢測燈塔

學校名稱：國立苗栗高級農工職業學校

作者：	指導老師：
職三　徐振哲	黃學志
職三　田宇桓	羅昌財
職三　邱柏勛	

關鍵詞：空汙、燈塔、物聯網

摘要

我們的學校兩旁是農田且附近有大型化學肥料工廠，有時空氣品質不佳，可是空氣品質的旗幟卻是插著綠色良好的標誌，不禁讓我們懷疑它的準確性及即時性。在市售的空氣盒子中，都是小型而且數據無法即時觀看，也沒有明顯燈號顯示，不適合學校單位使用，因此我們想做出既美觀又可以隨時知道空氣品質的測定器。

作品利用 3D 繪圖軟體自行設計，繪製亮麗吸睛的外型再加上每分鐘測定一次，準確性遠超過空氣品質旗幟，再加入溫溼度感測器並將數據上傳至物聯網，並可以顯示「符合行政院環保署空氣品質指標(AQI)與健康影響之狀態色塊」(表 1)的 LED 燈示及語音廣播，讓大家都能了解當時的空氣品質及時提醒及防護，以維護師生和民眾的健康。

表 1.環保署空氣品質指標與狀態色塊

對健康影響	良好	普通	對敏感族群不健康	對所有族群不健康	非常不健康	危害
空氣品質指標(AQI)	0~50	51~100	101~150	151~200	201~300	301~500
PM2.5 空氣汙染指標	0~15.4	15.5~35.4	35.5~54.4	54.5~150.4	150.5~250.4	250.5~500
狀態色塊						

壹、研究動機

工業發達，工廠林立，隨之而來的廢氣也危害人的健康。近年來注重環保及健康的呼聲愈來愈高漲，細懸浮微粒 PM2.5 儼然是全世界必須正視的問題，不管是本地污染源，又或者是鄰國的污染，這些都是在這片土地上的我們必須去面對的，去年至今不時的在新聞上看到「紫爆」這個詞，而我們其實都不知道現在的空氣品質的好壞，多半都是用肉眼來看。

以我們的學校為例，學校位於兩旁是農田及附近有大型化學肥料工廠的環境之下，有時空氣品質變差，但每天插的空氣品質旗號依然沒有變，不禁讓我們懷疑它的準確性及即時性。在市售的空氣 PM2.5 偵測盒子中，都是小型而且數據無法即時觀看，也沒有明顯燈號顯示，不適合學校單位使用，因此我們想做出既美觀又可以隨時知道空氣品質的「空汙報你知-守護你我檢測燈塔」

我們希望做出可以將數據上傳至環境物聯網的作品外，還希望它可以顯示「符合行政院環保署空氣品質指標(AQI)與健康影響之狀態色塊」的 LED 燈及語音廣播，讓經過的所有人都可以快速、簡單、即時的得知當地的空氣品質情況，保護自我及他人健康，為環保盡上一份心力。

貳、研究目的

一、運用 Arduino 程式設計及電路設計，創作出能即時測定及更新目前空氣品質之燈號。

二、用明顯的燈號、LED 液晶顯示及語音廣播讓大家可以在遠處就能清楚了解到現在的空氣品質。

三、利用 3D 電腦繪圖軟體設計把 PM2.5 檢測顯示器轉化成裝置藝術品，將其融入生活周遭，達到簡單清楚美觀的設計。

四、利用無線網路把裝置上所測到的數據傳送到行政院環保署環境物聯網，讓即將前往裝置所在地的人能提前知道當地的空氣品質。

參、研究設備及器材

表 2.研究設備

名稱	數量	單位	備註
3D 列印機	5	台	YING ANGEL,UP mini
雷射切割機	1	台	台灣三軸科技 DC-9060S
電腦	3	台	設計與繪圖用

表 3.研究工具

名稱	數量	單位	備註
銼刀	3	把	有粗、中、細三種不同尺寸
砂紙	5	張	去毛邊用
3M 耐熱膠帶	1	捲	貼於 3D 列印機底板上
游標卡尺	1	把	量測尺寸
SolidWorks 軟體	2	套	設計與繪圖用
Cura 轉檔軟體	1	套	轉成 3D 列印機的檔案
SD 記憶卡	4	個	儲存檔案以供列印機列印

表 4.研究材料

名稱	數量	單位	備註
線材	5	捲	紅,黑,橘
螺絲	多	個	材料
燈條及控制模組	2	組	電機模組
電線	1	捲	材料
LED 螢幕面板	1	個	電機模組
Arduino ESP32 開發版	1	個	電機模組
PMSA003 粉塵感測器	1	個	電機模組
溫溼度感測器	1	個	電機模組
風扇	1	個	電機模組
揚聲器	1	個	電機模組
電源供應器	1	個	電機模組
2mm 半透明壓克力板	1	片	材料
3mm 透明壓克力板	1	片	材料
5mm 木板	1	片	材料
5mm 白色壓克力板	2	片	材料

肆、研究過程或方法

一、 研究過程

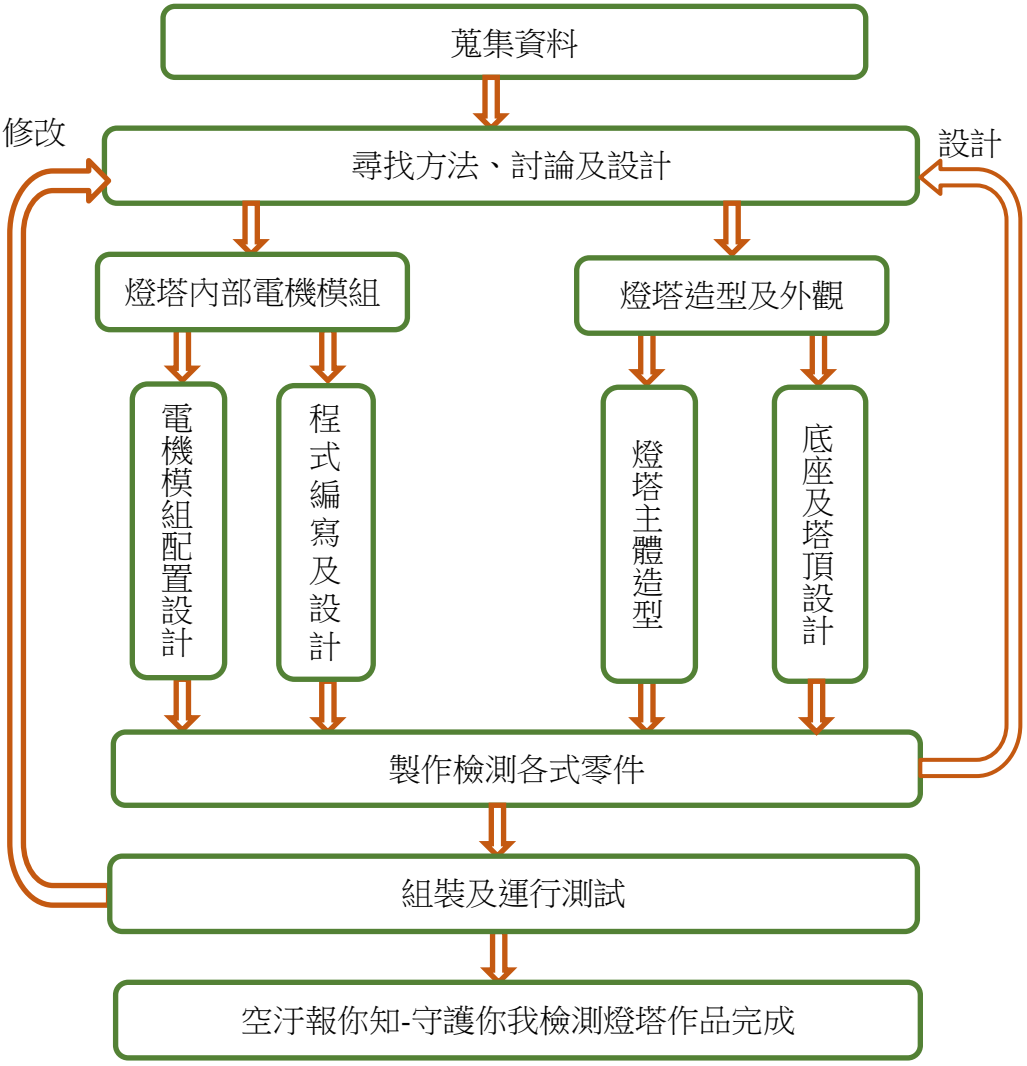


圖 1.研究過程圖

二、蒐集資料

校園空汙旗幟及市售檢測器之優缺點比較(如表 5)

(一)校園空氣品質旗幟

這種旗幟在校園內很常見雖然價格便宜但功能性不佳，無即時性且準確度低。

(二)空氣品質檢測儀 1

此產品比旗幟來得準確且功能較多，但沒有語音功能，只有數據沒有顏色燈示，無法了解目前空氣品質之狀態。

(三)空氣品質檢測儀 2

此產品價格偏高能測得的數據只有 PM2.5，雖然它是攜帶式但功能偏少。

表 5.市售空氣品質偵測器之比較表

	空氣品質旗幟	空氣品質檢測儀 1	空氣品質檢測儀 2
圖片			
空氣品質偵測功能	無	PM2.5/甲醛/TVOC 揮發性有機物	PM2.5
外觀設計	普通	佳	佳
數據更新頻率	1 天	5 分鐘	3 分鐘
語音功能	無	超標有警告聲	無
明顯燈號	無	無	無
傳數據至雲端	無	無	無
機體顯示測得數據	無	有	有
室外防水	無	無	無
總評	差	普通	普通

三、作品設計方向

我們蒐集及分析各種市售產品後，發現其優缺點，綜合以上優缺點我們訂出以下設計方向:

(一)亮麗吸睛的外型

作品想要表達『守護』你我健康的理念，於是決定使用燈塔的外型來詮釋此理念，而且要符合亮麗的現代感，所以利用 3D 繪圖軟體來繪製其外觀。

(二)環保材料

近年來環保意識抬頭，所以作品也儘量採用 PLA 環保塑料的 3D 列印製成，因為 ABS 塑料是由石油提煉而成，在加熱 時會產生有毒氣體，PLA 環保塑料則是由玉米粉提煉，環保且兼具防水功能。塔身的部分，我們採用壓克力材料，因為燈示需有透光材質的特性，製程方面採用雷射切割，因為可以快速又準確的製成零件。

(三)燈號及語音廣播

運用 Arduino 程式設計及電路設計，創作出能即時測定及更新目前空氣品質(AQI)之燈號。用明顯的燈號、LED 液晶顯示及語音廣播讓民眾可以在遠處就能清楚了解到現在的空氣品質。

四、設計原理

(一)積層製造

積層製造技術是一種層層堆疊成型技術，此技術從早期被稱為快速原型（RP）轉變成快速製造（RM），直至最近由 2010 年後，創客運動風起雲湧，因應低價低耗材與低營運成本需求，而發展眾多桌上型技術，被廣泛稱為 3D 列印。而本作品的底座及屋頂的部份由 3D 列印製成，因為設計中之底座、屋頂有許多鏤空、斜面及曲面的部份，用一般傳統加工不易完成，而 3D 列印積層製造對這些難項，卻可輕鬆完成，故積層製造技術製程是本作品最佳的選擇。

(二)雷射雕刻及切割

其技術原理為：雷射光束經激發後，從雷射管經由擴束鏡以一定的角度照射到反射鏡(1)，再反射到反射鏡(2)，最後通過 f-θ 聚焦鏡，照射到打標工作上，使工件受高熱燒穿燒斷(圖 2)。

作品中本體的部份安裝燈條，燈光需要用可穿透性的材質，透明及半透明壓克力為我們的最佳選擇，而目前能切斷壓克力的最佳設備就是雷射切割機了。故組員們於本體製造選用雷射切割，切割 230 片零件並組裝完成。

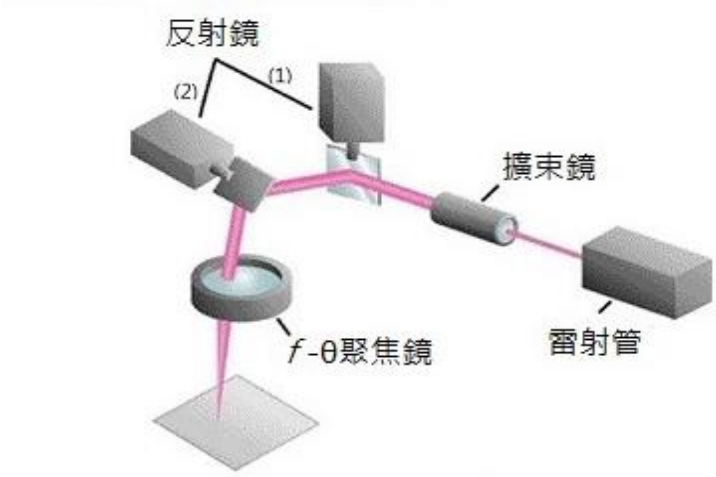


圖 2.雷射雕刻原理 PMSA003

圖 3.粉塵感測器反面實體圖

(三)電路設計及製作

1.PMSA003 粉塵感測器

PMSA003 粉塵感測器是一款基於鐳射散射原理的數位式通用顆粒物濃度感測器，可連續採集並計算單位體積內空氣中不同粒徑的懸浮顆粒物個數，即顆粒物濃度分佈，進而換算成為品質濃度，並以通用數位介面形式輸出（如圖 3、表 6、圖 4）。

表 6. PIN 腳之功用介紹

PIN1	VCC	電源正 5V
PIN2	VCC	電源正 5V
PIN3	GND	電源負
PIN4	GND	電源負
PIN5	RESET	模組重定信號/TTL 電平@3.3V，低復位
PIN6	NC	
PIN7	RX	串口接收管腳/TTL 電平@3.3V
PIN8	NC	
PIN9	TX	串口發送管腳/TTL 電平@3.3V
PIN10	SET	設置管腳 /TTL 電平@3.3V，高電平或懸空

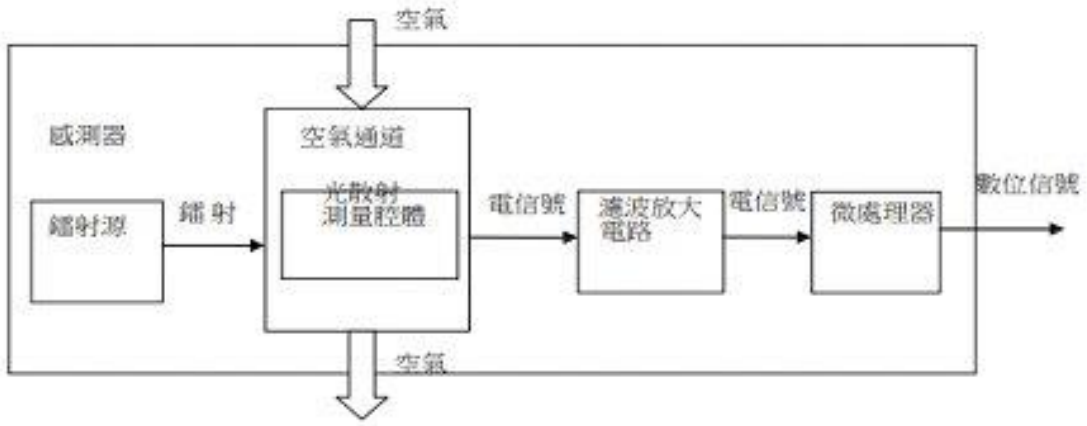


圖 4. PMSA003 感測器原理圖

PMSA003 粉塵感測器可測出 PM 1.0、 PM 2.5、PM 10 與粉塵顆粒高精度感測器。粉塵感測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。

2. ESP32 開發板

ESP32 是一款 WiFi 和藍牙系統級晶片 (SoC)，ESP32 完全符合 WiFi 802.11b/g/n/e/i 和藍牙 4.2 的標準，集成了 WiFi/藍牙/BLE 射頻和低功耗技術，並且支持開放性的即時操作系統

3.燈條、ESP32 開發板、粉塵感測器等電機元件整體接線

(1)將燈條分別與 ESP32 開發板之 IO22(PIN22)、IO23(PIN21)、IO34(PIN5)、

IO35(PIN6)連接。

(2) ESP32 開發板 EXT_5V(PIN19)、PMSA003 粉塵感測器(PIN1)、溫濕度感測器(PIN1)與電源 5V 連接。

(3) ESP32 開發板 GND3(PIN20)、PMSA003 粉塵感測器 GND(PIN3)、溫濕度感測器 GND (PIN4)與電源負連接。

(4) PMSA003 粉塵感測器與 ESP32 開發板之 RX、TX 連接。

(5) 溫濕度感測器 DATA (PIN2)與 ESP32 開發板 IO21 (PIN25) 連接。

將 ESP32 開發板燒入 Arduino 程式後，再依接線連接正確，即可測試，若是有錯誤則再檢查 Arduino 程式偵錯，直到沒有錯誤，即可與硬體安裝組合。（圖 5）

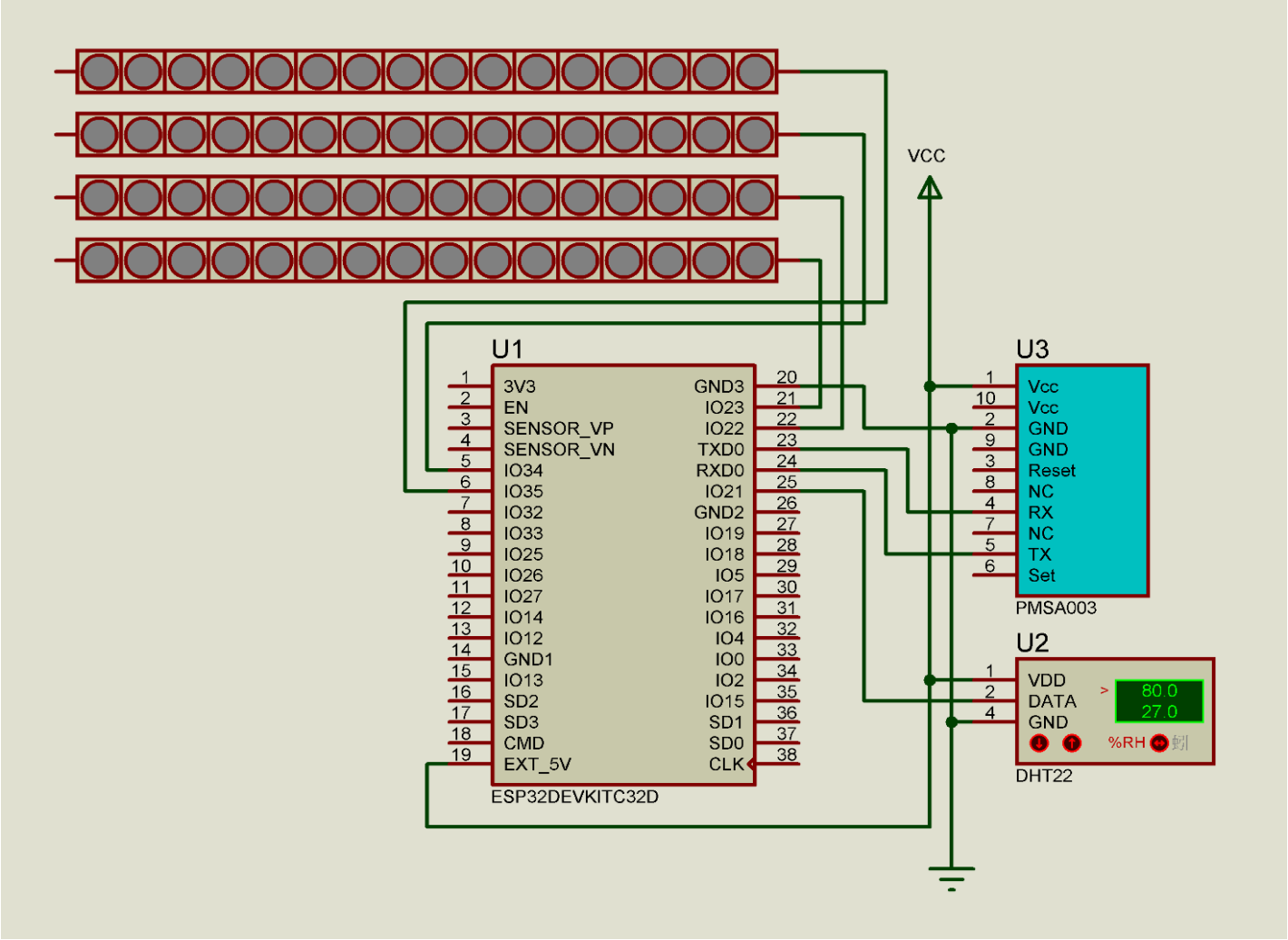


圖 5. 四條燈條、ESP32 開發板及粉塵感測器整體接線圖

五、電機系統架構及設計圖

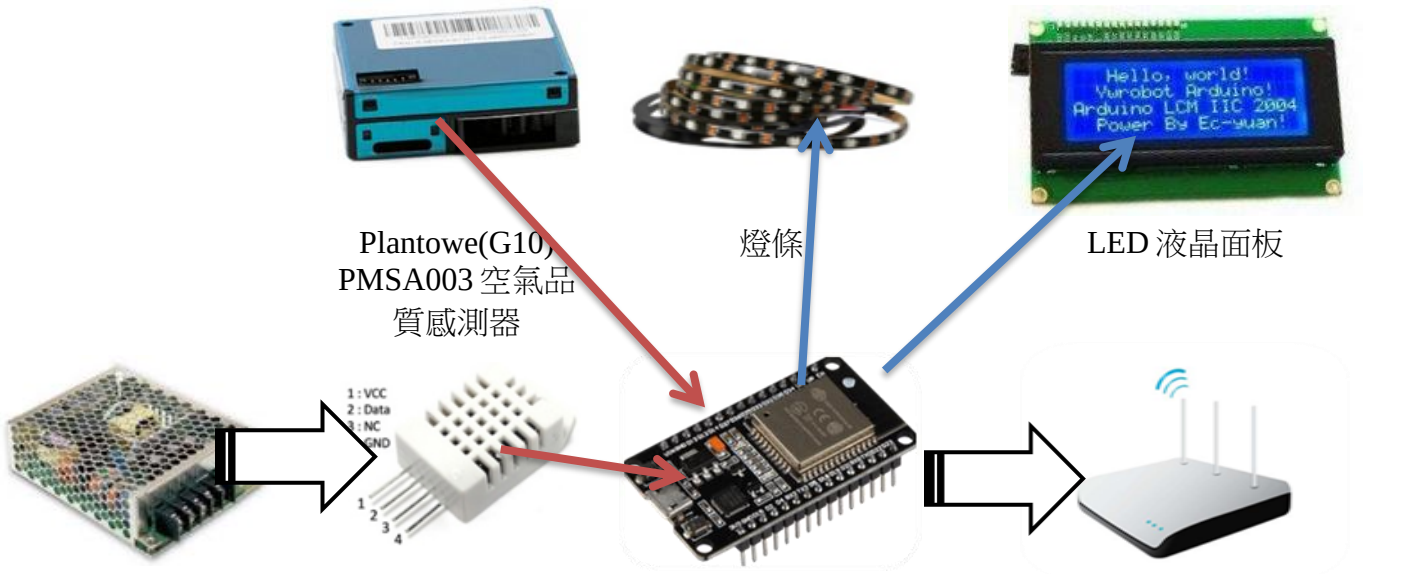


圖 6.系統架構設計圖

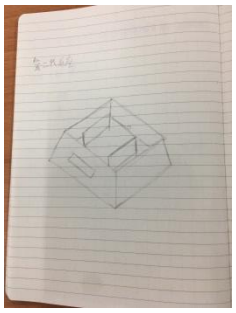
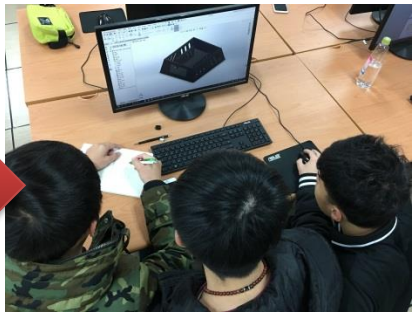
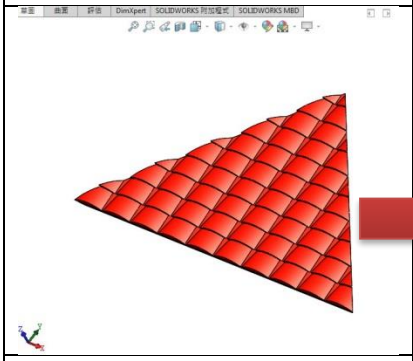
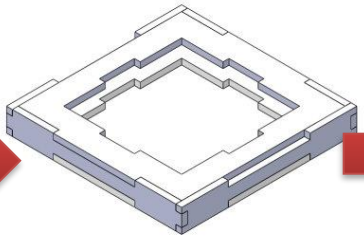
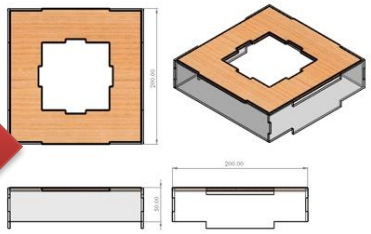
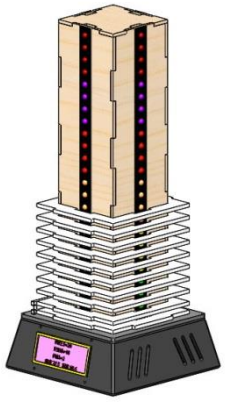
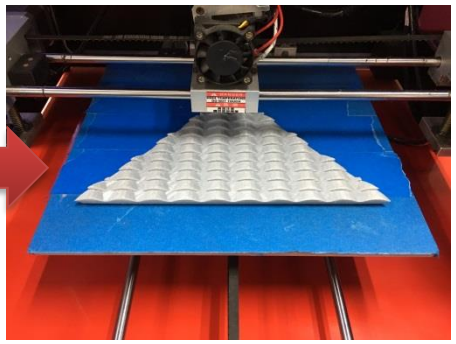
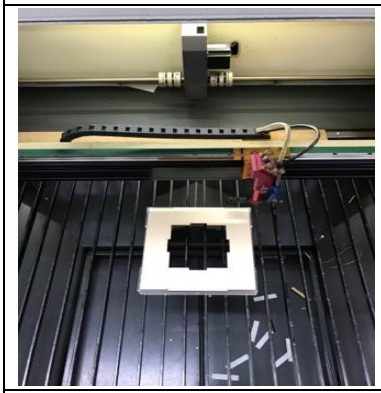
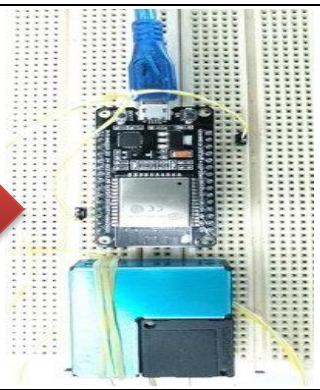
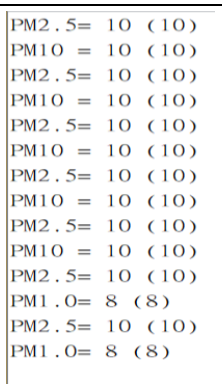
六、研究歷程甘特圖

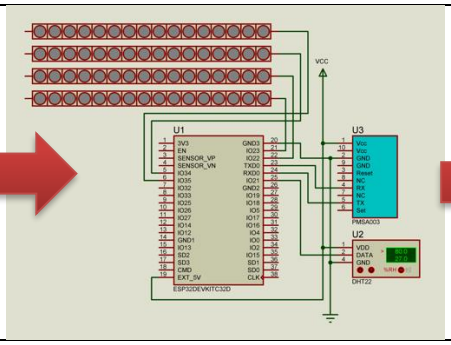
表 7.研究歷程甘特圖

日期 時段 / 工作內容	06/25 ~ 07/08	07/09 ~ 07/16	07/17 ~ 07/22	07/26 ~ 07/29	07/30 ~ 08/05	8/06 ~ 08/14	08/15 ~ 08/26	09/04 ~ 09/13	09/16 ~ 09/25	09/26 ~ 12/06	12/07 ~ 12/13	12/14 ~ 12/19	12/20 ~ 01/03	01/06 ~ 01/22	01/23 ~ 01/30	01/31 ~ 02/05	02/06 ~ 02/17
蒐集資料 尋找方法	■															■	
小組討論 擬定計劃	■	■	■	■	■	■	■	■	■	■		■	■	■		■	
手繪草圖 制定尺寸		■			■		■										
繪製各部零件 購買電子零件		■			■		■		■				■				
組合圖 及修改尺寸						■	■	■			■	■	■				
3D 列印 雷射雕刻						■	■		■			■	■				
零件組合,電機元件組裝						■	■	■			■	■	■				
學習 Arduino 編寫及修改程式										■				■			
性能測試											■		■		■		
專題作品 報告撰寫																■	■

七、作品之製作歷程(表 8)

表 8. 製作歷程

		
1.蒐集資料	2.手繪草圖	3.小組討論
		
4.元件設計	5.第一層設計	6.第二層設計
		
7.部分組合	8.完整組合	9.3D 列印
		
10.雷射切割	11.ESP32 與 PMSA003 接線	12.ESP32 與 PMSA003 測試

		
13.模型組裝	14.整體軟件接線	15.完成及測試

伍、研究結果

一、第一代測定燈塔之設計

圖 7.第一代測定燈塔

圖 8.第一代測定燈塔

表 9. 第一代測定燈塔比較表

優點	1.明顯燈號使眾人遠處就能看見
	2.利用程式設計使 LED 燈條依 PM2.5 濃度，顯示高低及顏色
缺點	1.木板容易吸水，底座無法排水，時間久了容易腐爛
	2.塔頂並無排水設計
	3.無 LED 數值面板
	4.電源供應器無墊高雨天時容易進水故障
	5.外型普通，無設計感

剛開始想出這個燈塔的設計真的很不容易，當初的設計目的不但要有亮麗的外觀還要利用明顯的燈號吸引眾人的目光來達到人們從遠處就可以馬上知道現在的空氣品質汙染程度，最後組員們與老師討論出來的以台北 101 的塔形作為本作品的造型外觀(圖 7、8)，在做測試時也發現許多缺點與不足的地方(表 9)，例如:作品在室外如果下雨底座容易吸水腐爛、沒有 LED 顯示器觀看 PM2.5 數值等等，我們也將問題記錄下來與指導老師討論與改良。

二、第二代燈塔設計及改良

第一代燈塔雖有達成顏色及燈示測定功能，但缺點也不少，故開始第二代之設計，故以保留第一代設計之優點及改善缺點為設計方向。在第二次的改良中為了不要產生干涉也重新訂定尺寸。

圖 9.第一代底座

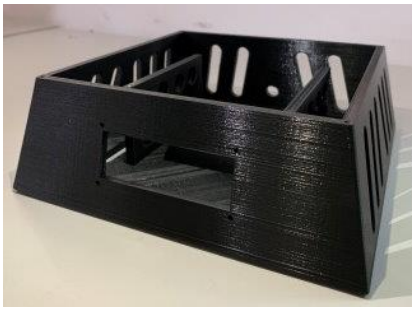


圖 10.第二代底座外觀

圖 11.第二代底座內部

(一)底座設計及修正

第二代我們改用 3D 列印製造，塑料材質是 PLA 由玉米粉製成無毒安全又環保且兼具防水功能。由於第一代的平面設計防水功能不佳(圖 9)，所以將其外殼採用斜面設計，並在底面挖兩個槽以利排水，中間的小平台是為了防止電源供應器泡水所以我們將它墊高 10mm，又設計美觀的散熱孔洞來解決電源供應器的散熱問題(圖 10、11)。

(二)測定塔造型設計及修正

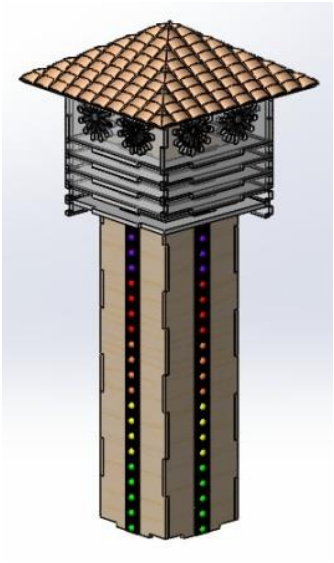


圖 12.四面燈條設計

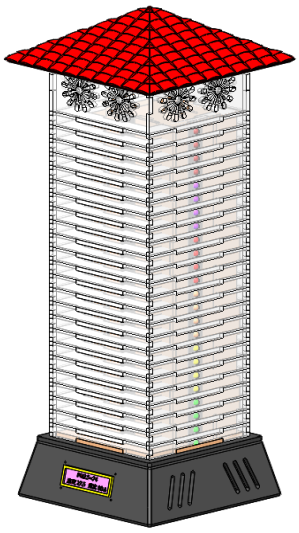


圖 13. 第二代作品整體設計

作品的上下底蓋為了不讓指示燈互相干擾，我們選用不透明的壓克力板，塔身的外殼我們使用的是半透明與透明的壓克力板黏合，我們使用半透明的原因是可以有燈光發散的效果，使整個塔身看起來亮眼且美麗，宛如裝置藝術一般。(圖 12、

13)

利用電腦繪圖軟體 SolidWorks 作設計，因此軟件精確且功能強大，組合圖狀態時可做干涉檢查，避免設計錯誤而造成無法組裝的材料浪費情況，如下圖(圖 14)所示。

我們使用創新實作的方法，利用 3D 列印可多樣造型設計，避免開模製造的大量成本，也使用壓克力採取雷射雕刻製程，速度快美觀且燈式透光度佳。

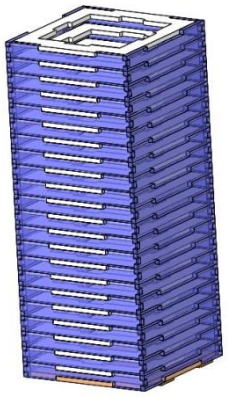


圖 14.部分塔身

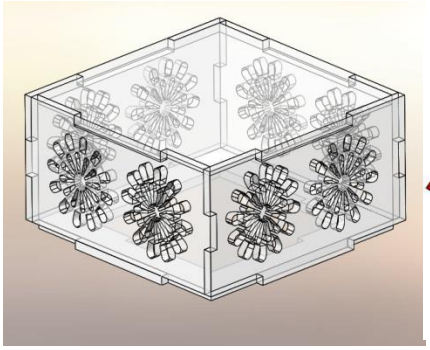


圖 15.頂抽氣及散熱設計

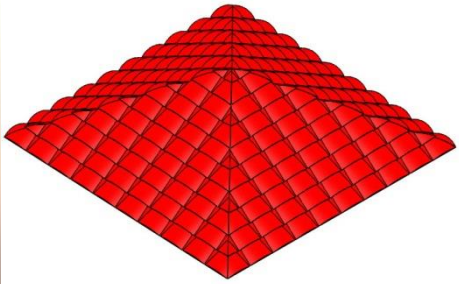


圖 16.屋頂排水設計

為了讓塔頂裡面的電子儀器散熱及抽氣，我們選用透明壓克力來製作，在四周設計孔洞讓風扇吸入空氣給測定器檢測，為了美觀設計花瓣造型的孔，更有藝術感(圖 15)。

要避免塔頂內的電子儀器被雨水打濕而故障，所以我們在塔頂上方安裝屋頂，另外在屋頂的表面增加瓦片不僅在下雨時雨水滴落屋頂時能排水也增加美觀性(圖 16)。

三、第三代燈塔設計及改良

(一)第三代底座設計

圖 17. 第三代底座

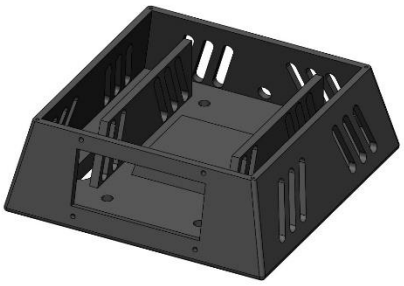


圖 18.第三代底座外觀

圖 19.第三代底座底部

第三代底座(圖 17、18、19)相較於第二代的最大差別在加大 LED 面板的安裝孔洞，將只能顯示兩排訊息的 LED 液晶面板，更換為四排大型 LED 面板，使得更多

資訊能顯示在面板上，達到一目了然的目的。

(二)第三代測定塔造型設計及修正

於二代檢測塔完成後，我們實行兩次的雨天實測，發現第一次雨天實測雨勢較小，故一切正常，而第二次雨天實測雨勢風勢皆大，測試完成後發現儀器數值雖然正常，但頂層儀器放置處因為雨勢風勢大而略有進水之狀況，且層與層接縫處也有一點點滲水情形，故必須進行改良。

組員們先設計層與層接縫彌封設計(圖 20)，如此解決了層層接縫滲水情形，再者為頂層問題，所以我們查閱書籍發現百葉魚鱗設計可達通風散熱又可防風防雨的優點，於是我們著手修正頂層設計(圖 21)，修正完成後進行組合圖的組裝(圖 22)及製作完成後的實體圖(圖 23)。

圖 20. 層與層接縫彌封設計

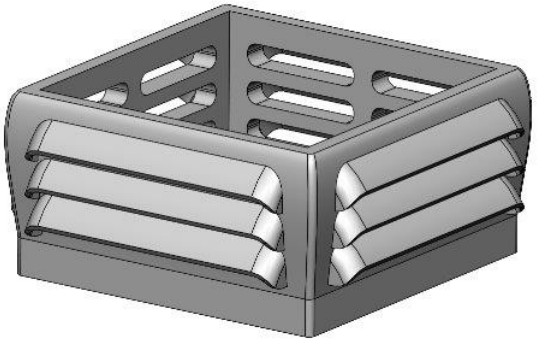


圖 21. 頂層百葉魚鱗設計

圖 22. 第三代組合圖

圖 23.第三代作品實體圖

四、Arduino 程式設計

我們利用 arduino-1.8.10 開發軟體設計控制程式，並參考多本 arduino 書籍及詢問老師，完成了以下「檢測燈塔」之 Arduino 程式，如下表所示(表 10)：

表 10.「檢測燈塔」之 Arduino 部份程式

程式內容	說明
#include <Arduino.h>	
#include <WiFi.h>	//載入 WiFi 模組
#include <HTTPClient.h>	//載入網頁應用模組
#include <TimeLib.h>	//載入時間模組
#include <PubSubClient.h>	//MQTT 模組
#include "DHT.h"	//DHT22 溫度感測器
#include <Adafruit_NeoPixel.h>	//載入 WS2812B 控制模組
#include <HardwareSerial.h>	//TX2,RX2 序列埠模組控制語音播放
#include <JQ6500_Serial.h>	//JQ6500 語音模組
#include <Wire.h>	//I2C 模組
#include <LiquidCrystal_I2C.h>	//LCD 顯示模組
LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display	
JQ6500_Serial mp3(16, 17);	//產生語音模組實體 mp3
#define DHTPIN 19	//設定感測器接腳
#define DHTTYPE DHT22	//設定感測器類型 DHT22
#define PIN 23	//燈條 1 接腳
DHT dht(DHTPIN, DHTTYPE);	//產生實體 dht
Adafruit_NeoPixel strip = Adafruit_NeoPixel(24, PIN, NEO_GRB + NEO_KHZ800);	
//-----設定 MQTT Server 各項參數----------	設定 MQTT Server 各項參數
const char* mqttServer = "iot.epa.gov.tw";	MQTT Server 網址
const int mqttPort = 1883;	
const char* mqttUser = "PK9U79PZKC57E19S4G";	
const char* mqttPassword = "PK9U79PZKC57E19S4G";	
//----------	
long nowTime = millis();	//取得模板的啟動時間
//----------	//----PMSA003 各項數據變數設定----------
long pmcf10 = 0, pmcf25 = 0, pmcf100 = 0;	
long pmat10 = 0, pmat25 = 0, pmat100 = 0;	
//----------	//------產生 MQTT 實體--------
WiFiClient espClient;	
PubSubClient client(espClient);	
//----------	
float h ;	//--濕度變數----
float t ;	//--攝氏溫度變數-----

```
float hic ;                                                                  //--體感溫度---------
byte degree[8]={B00000110, B00001001, B00001001, B00000110,
0, 0, 0, 0};
unsigned long strToLong(String p) {                                          //-----字串轉長整數函數------
  unsigned long temp = 0;
  for (int i = 0; i < p.length(); i++) {
    temp += (p.substring(i, i + 1).toInt()) * pow(10, p.length() - i - 1);
  }
  return temp;
}
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();                                                           //初始化 LCD 模組
  lcd.createChar(0, degree);
  mp3.begin(9600, SERIAL_8N1, 16, 17);
  mp3.reset();                                                               //----初始化語音模組設定
  mp3.setVolume(30);
  mp3.setLoopMode(MP3_LOOP_NONE);                                            //設定音量
  lcd.print("Hello World!");                                                 //不循環播放
  strip.begin();
  strip.show();                                                              //燈條初始化
   WiFi.mode(WIFI_AP_STA);                                                   //燈條顯示
  WiFi.beginSmartConfig();                                                   //----智能 WiFi 設定---------
  dht.begin();                                                               //----初始化設定 ESP32 為 AP
    Serial.println("Waiting for SmartConfig.");                              及 STATION 兩用-----
  while (!WiFi.smartConfigDone()) {                                          //dht 初始化
    delay(500);                                                              //----等待手機 APP 設定連線
        while (Serial.available()) {
     CopeSerialData(Serial.read());                                          //demo();//等待連接 WIFI 時,
    }                                                                        全彩 LED 燈展示
    show2LCD();
    if (millis() - nowTime > 60000) {
      speakData(pmat25);
      nowTime = millis();
    }
    Serial.println(pmat25);
    Serial.print(".");
    showLevel(pmat25);
  }
  Serial.println("");
  Serial.println("SmartConfig received.");
  //-------------------------------------
    Serial.println("Waiting for WiFi");
  while (WiFi.status() != WL_CONNECTED) {                                    //----等待 WiFi 連接 AP-------
```

```
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi Connected.");
  Serial.print("IP Address: ");                                    //-----連接 MQTT Server-------
  Serial.println(WiFi.localIP());
    client.setServer(mqttServer, mqttPort);
    HTTPClient http;
  http.begin("http://web1.foxhollow.ca/nodemgr/?unixtime");
//HTTP                                                            //--透過網頁獲取現在時刻---
  int httpCode = http.GET();
  if (httpCode > 0) {
    if (httpCode == HTTP_CODE_OK) {
      unsigned long getTimes = strToLong(http.getString()) + 28800;
      setTime(getTimes);
    }
  } else {
    Serial.printf("[HTTP]      GET...      failed,      error:      %s\n",
http.errorToString(httpCode).c_str());
  }
  http.end();
  //-------------------------------------------------
}
void loop() {
  delay(2000);                                                    //主程式
  while (Serial.available()) {
    CopeSerialData(Serial.read());
  }
  h = dht.readHumidity();
  t = dht.readTemperature();                                      //透過感測器獲得溫濕度數
  hic == dht.computeHeatIndex(t, h, false);                       值
  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;                                                       //感測器失誤時印出之訊息
  }
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(hic);
  Serial.println(F("oC "));
  if (millis() - nowTime > 60000) {
    while (!client.connected()) {                                 //每隔一分鐘執行傳送資料
      Serial.println("Connecting to MQTT...");                    至物聯網
      if (client.connect("ESP32Client", mqttUser, mqttPassword )) {
```

Serial.println("connected");
} else {
Serial.print("failed with state ");
Serial.print(client.state());
delay(2000);
}
}
	//連接 MQTT 伺服器

String pubData25 = "[{\"id\":\"pm2_5\",\"time\":\"" + String(year()) + "-" + String(month()) + "-" +String(day()) + " " + String(hour()) + ":" + String(minute()) + ":" + String(second()) + "\",\"value\":[\"" + String(pmat25) + "\"]}]";	//組合 PM2.5 數據
String pubData10 = "[{\"id\":\"pm1\",\"time\":\"" + String(year()) + "-" + String(month()) + "-" +String(day()) + " " + String(hour()) + ":" + String(minute()) + ":" + String(second()) + "\",\"value\":[\"" + String(pmat10) + "\"]}]";	//組合 PM1.0 數據
String pubData100 = "[{\"id\":\"pm10\",\"time\":\"" + String(year()) + "-" + String(month()) + "-" +String(day()) + " " + String(hour()) + ":" + String(minute()) + ":" + String(second()) + "\",\"value\":[\"" + String(pmat100) + "\"]}]";	//組合 PM10 數據
String temperature="[{\"id\":\"temperature\",\"time\":\""+tring(year()) + "-" + String(month()) + "-" +String(day()) + " " + String(hour()) + ":" + String(minute()) + ":" + String(second()) + "\",\"value\":[\"" +String(t) + "\"]}]";	//組合溫度數據
String humidity = "[{\"id\":\"humidity\",\"time\":\"" + String(year()) + "-" + String(month()) + "-" + String(day()) + " " + String(hour()) + ":" + String(minute()) + ":" + String(second()) + "\",\"value\":[\"" +String(h) + "\"]}]";	//組合濕度數據

char pub[100];
pubData25.toCharArray(pub, 100);
client.publish("/v1/publish/project/865/device/7768530482/rawdata", pub);	

pubData10.toCharArray(pub, 100); client.publish("/v1/publish/project/865/device/7768530482/rawdata", pub);	//傳送 PM2.5 數據
pubData100.toCharArray(pub, 100); client.publish("/v1/publish/project/865/device/7768530482/rawdata", pub);	//傳送 PM1.0 數據
temperature.toCharArray(pub, 100); client.publish("/v1/publish/project/865/device/7768530482/rawdata", pub);	//傳送 PM10 數據
humidity.toCharArray(pub, 100); client.publish("/v1/publish/project/865/device/7768530482/rawdata", pub);	//傳送溫度數據
//Serial.println(pubData);	//傳送濕度數據

```
    nowTime = millis();
    speakData(pmat25);
  }
}

void show2LCD() {
  h = dht.readHumidity();
  t = dht.readTemperature();                    //---LCD 顯示副程式----------
  lcd.setCursor(3, 0);
  lcd.print("PM2.5=");
  if (pmat25 < 10) {
    lcd.print("00");
  } else if (pmat25 < 100) {
    lcd.print("0");
  }
  lcd.print(pmat25);
  lcd.setCursor(1, 1);
  lcd.print(t);
  lcd.write(0);
  lcd.print("C  ");
  lcd.print(h);
  lcd.print("%");
}
//-----------------------------
void speakData(long number) {
  byte num1 = number / 100;                     //---語音系統副程式------------
  byte num2 = (number / 10) % 10;
  byte num3 = number % 10;
  mp3.playFileByIndexNumber(1);
  delay(3500);
  if (num1 != 0) {
    mp3.playFileByIndexNumber(19 + num1);
    Serial.print(num3);
    delay(1000);
  }
  if (num2 != 0) {
    mp3.playFileByIndexNumber(10 + num2);
    Serial.print(num2);
    Serial.print(",");
    delay(1000);
  }
  if (num3 != 0) {
    mp3.playFileByIndexNumber(num3 + 1);
    Serial.println(num1);
    delay(1000);
```

```
  }
  mp3.playFileByIndexNumber(22);
  delay(2000);
  int a = pmat25;
  switch (a) {
    case 0 ... 15:
      mp3.playFileByIndexNumber(23);
      break;
    case 16 ... 35:
      mp3.playFileByIndexNumber(24);
      break;
    case 36 ... 54:
      mp3.playFileByIndexNumber(25);
      break;
    case 55 ... 150:
      mp3.playFileByIndexNumber(26);
      break;
    case 151 ... 250:
      mp3.playFileByIndexNumber(27);
      break;
    default:
      mp3.playFileByIndexNumber(28);
      break;
  }
  delay(2000);
}
char CopeSerialData(unsigned char ucData) {
  static unsigned char ucRxBuffer[250];
  static unsigned char ucRxCnt = 0;
  static unsigned char toggle = 0;
  long  HCHO = 0, temp = 0, humi = 0;

  ucRxBuffer[ucRxCnt++] = ucData;

  if (ucRxBuffer[0] != 0x42 && ucRxBuffer[1] != 0x4D) {
    ucRxCnt = 0;
    return ucRxCnt;
  }
  if (ucRxCnt < 32) { // PMS7003
    return ucRxCnt;
  }
  else {
    pmcf10 = (float)ucRxBuffer[4] * 256 + (float)ucRxBuffer[5];
    pmcf25 = (float)ucRxBuffer[6] * 256 + (float)ucRxBuffer[7];
    pmcf100 = (float)ucRxBuffer[8] * 256 + (float)ucRxBuffer[9];
```

//----取得 PMSA003 各項數據函數----------

```
    pmat10 = (float)ucRxBuffer[10] * 256 + (float)ucRxBuffer[11];
    pmat25 = (float)ucRxBuffer[12] * 256 + (float)ucRxBuffer[13];
    pmat100 = (float)ucRxBuffer[14] * 256 + (float)ucRxBuffer[15];
    ucRxCnt = 0;
    return ucRxCnt;
  }
}
void showLevel(long x) {
  for (int i = 0; i < 24; i++) strip.setPixelColor(i, 0);
  delay(2);
  long color[5] = {strip.Color(0, 255, 0), strip.Color(255, 255, 0),
strip.Color(255, 128, 0),
strip.Color(255, 0, 0), strip.Color(128, 0, 128)
                  };
  if (x <= 15.4) {
    for (int i = 0; i < rank(x, 3, 6, 9, 12); i++) strip.setPixelColor(23 -
i, color[0]);
  } else if (x <= 35.4) {
    for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
    for (int i = 0; i < rank(x, 19, 23, 27, 31); i++) strip.setPixelColor(18
- i, color[1]);
  } else if (x <= 54.4) {
    for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(18 - i, color[1]);
    for (int i = 0; i < rank(x, 39, 43, 47, 51); i++) strip.setPixelColor(13
- i, color[2]);
  } else if (x <= 150.4) {
    for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(18 - i, color[1]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(13 - i, color[2]);
    for (int i = 0; i < rank(x, 73, 92, 107, 121); i++)
strip.setPixelColor(8 - i, color[3]);
  } else {
    for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(18 - i, color[1]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(13 - i, color[2]);
    for (int i = 0; i < 5; i++) strip.setPixelColor(8 - i, color[3]);
    for (int i = 0; i < rank(x, 175, 199, 223, 247); i++)
strip.setPixelColor(3 - i, color[4]);
  }
  strip.show();
  delay(2);
}……以上為部分程式之重要內容……
```

```
//---------判斷顏色----------
  //0-15.4 良好(綠色)

//15.5-35.4 普通(黃色)

//35.5-54.4-對敏感族群不健康(橙色)

//54.5-150.4 對所有族群不健康(紅色)

//150.5-250 非常不健康(紫色)
```

陸、討論

一、作品實測情形

表 11.實測數據表

編號	地點	實測圖	時間及細懸浮微粒數據						平均值
1	校門口		時間	14:28	14:29	14:30	14:31	14:32	
			PM2.5數據	76	86	80	75	77	78.8
2	化學工廠附近		時間	14:47	14:48	14:49	14:50	14:51	
			PM2.5數據	79	79	76	78	80	78.4
3	宮廟		時間	15:03	15:04	15:05	15:06	15:07	
			PM2.5數據	160	65	360	68	72	145
4	山區		時間	15:47	15:48	15:49	15:50	15:51	
			PM2.5數據	64	69	63	65	65	65.2

對健康影響	良好	普通	對敏感族群不健康	對所有族群不健康	非常不健康	危害
PM2.5 空氣汙染指標	0~15.4	15.5~35.4	35.5~54.4	54.5~150.4	150.5~250.4	250.5~500.4
狀態色塊						

實測當天室內有空調的教室，PM2.5 的平均值為 48，由上表(表 11)可知屬「對敏感族群不健康」的橘色等級，而室外皆屬於「對所有族群不健康」的紅色等級。這表示室內加上空調，將有效過濾空氣中細懸浮微粒的數量，即當空氣品質不佳時，為了自己健康著想，應該留在室內且開啟空氣清淨機或空調。

編號 4 山區的 PM2.5 的平均值為 65.2，與當天網路上公告的空氣品質 PM2.5 的平均值 63 的 數值差不多。

編號 1 校門口的 PM2.5 的平均值為 78.8，數值偏高，經過實測觀察，原因為校門口車輛進出頻繁，廢氣較多的緣故。

編號 2 化學工廠附近的 PM2.5 的平均值為 78.4，數值偏高，經過實測觀察，原因為工廠廢氣排氣，提高了細懸浮微粒的數量的緣故。

編號 3 宮廟的 PM2.5 的平均值為 145，數值超高，實測觀察為測定空氣品質指數的地點，太接近宮廟香爐，當香爐的煙飄過檢測燈塔時，數值一度飆高至 300 以上，故燒香的煙 PM2.5 細懸浮微粒的數量非常高，長期吸入人體內對健康有不良的影響。

二、作品所需成本

表 12.成本表

項目名稱	數量	單價
2mm 半透明壓克力板	1	300
3mm 透明壓克力板	1	450
5mm 木板	1	217
5mm 白色壓克力板*2	2	1500
LED 螢幕面板	1	110
Arduino EDP32 開發板	1	270
燈條*4	4	440
溫溼度感測器	1	150
空氣品質感測器	1	600
風扇	1	25
揚聲器	1	50
電源供應器	1	370
屋瓦*4	4	168
底座	1	220
塔頂	1	96
合計	4966 元	

三、作品雨天實測

圖 24.雨天測試

圖 25.雨天測試

圖 26.雨天測試

因為作品製作的方向原本就是要放置在室外，必須考慮有防水功能，所以在材質的選用上，使用 PLA 材質來製作屋瓦及底座，來達成防水效果，並在作品完成之後的下雨天，拿去戶外實測，我們將它放置在外兩個小時並每隔一段時間去觀察一次，在這段時間內作品運作完全正常，所以作品防水性測試是成功的。(圖 24、25、26)

四、操作方法

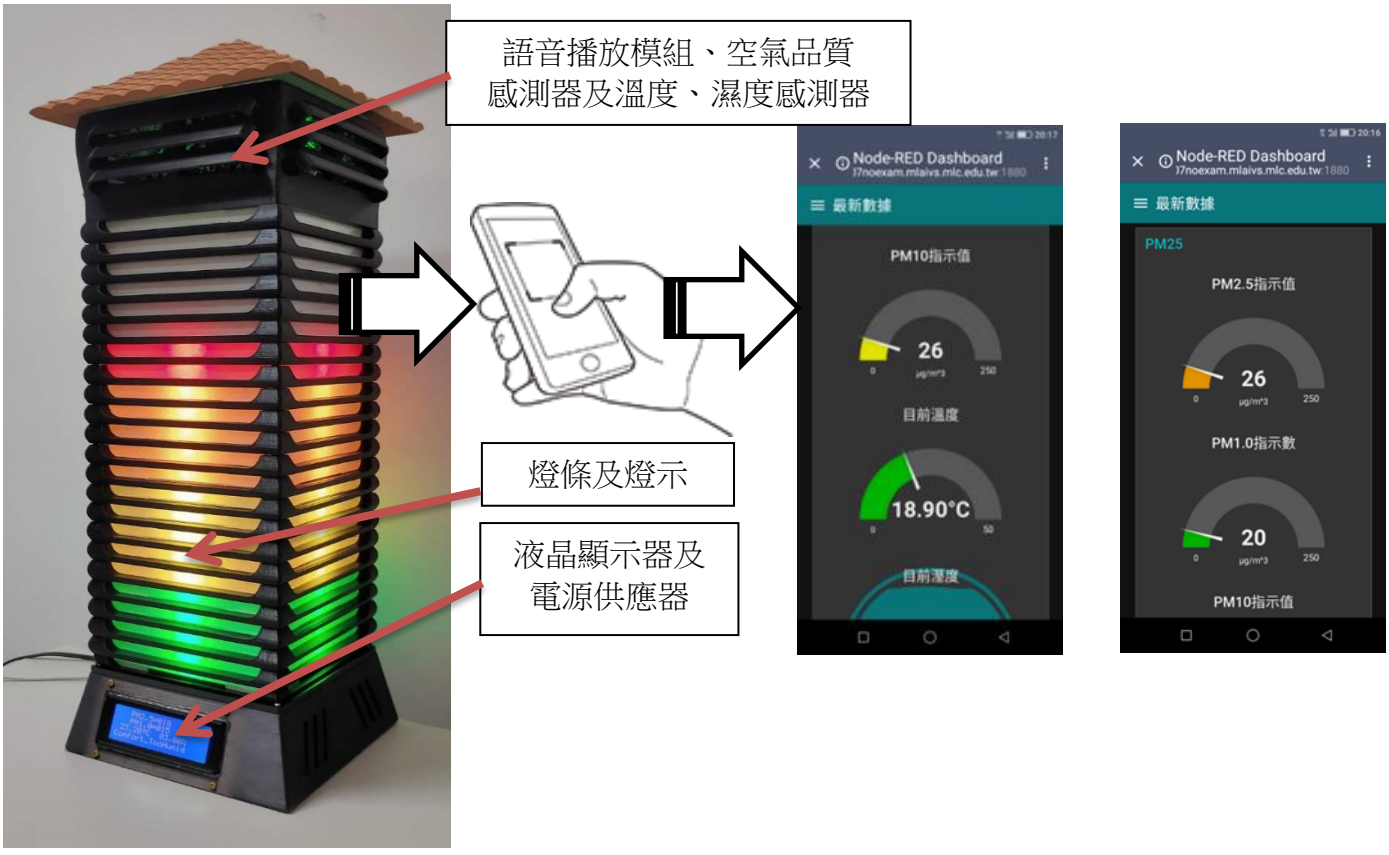


圖 27.作品實體與手機連接並進入網站及行動裝置畫面

作品操作很簡單，電源部分有 USB 插頭及 110V 電源插頭可自由選擇其一，但不可

都使用，否則會雙電流輸入損害電機模組。

將插頭插入插座，打開開關即可使用。使用手機連上網站即可看到最新上傳數據，或由作品 LED 液晶顯示器、燈條亮的顏色及作品語音就可得知目前當地之空氣品質的各項數據。(圖 27)

柒、結論

綜合以上研究及討論，可得到以下幾點結論：

一、成功利用電路與電路板的連結，達成即時測定空氣品質的目的。

二、完成用明顯的燈號、LED 液晶顯示及語音廣播讓民眾可以在遠處就能清楚了解到現在的空氣品質(AQI)之狀態。

三、利用電腦繪圖設計出簡潔俐落的外觀，讓路過的民眾、學生能在遠處或白茫茫的霧霾中看清楚燈示，宛如燈塔一般，了解空氣品質及 PM2.5 狀況，能即刻防患，使大家知道今天適不適合戶外活動，為環境保護及學生健康盡一己之力。

四、成功利用程式編寫與無線網路連接將數據回傳，可讓民眾用行動裝置及網路知道現在溫度、濕度、PM1.0、PM2.5、PM10 空氣品質之詳細數據。

經過不斷測定的數據來看，利用裝置可準確又即時的知道空氣品質指標(AQI)之狀態，本作品比機關或學校掛的空氣品質旗幟更加準確，而且美觀清楚又簡單，知道空氣品質何時安全何時紫爆，若能於學校及公務機關大量運用，放置於牌樓、學校司令台或校門口之明顯處，將比看氣象報導更準確，達成維護及提醒學生民眾身體健康的目的。

捌、參考資料及其他

- 吳權威。Solidworks2003 實務。台北市：碁峯資訊股份有限公司
- 黃建庭。輕鬆玩 Arduino 程式設計與感測器入門。台北市：碁峰資訊股份有限公司
- 行政院環境保護署-空氣品質監測網 https://taqm.epa.gov.tw/taqm/tw/b0201.aspx
- 楊明豐(2018)。Arduino 物聯網最佳入門與應用：打造智慧家庭輕鬆學。台北市：碁峰資訊股份有限公司
- 林健仁。Arduinoru 機器人與專題製作。台北市：全華圖書股份有限公司

【評語】052607

本作品運用 Arduino 程式設計及電路設計，作出能即時測定及更新目前空氣品質之燈號；以明顯的燈號、LED 液晶顯示及語音廣播讓民眾可以在遠處就能清楚了解到現在的空氣品質；作品也利用 3D 電腦繪圖軟體設計把 PM2.5 檢測顯示器轉化成裝置藝術品；並利用無線網路把裝置上所測到的數據傳送到行政院環保署環境物聯網，讓即將前往裝置所在地的人能提前知道當地的空氣品質，有助於提升一般民眾環境品質意識。但研究未能與實測數據進行比較，且未使用統計方法針對實驗結果進行分析等，均為未來研究可再加強者。

作品海報

壹.研究動機

工業發達，工廠林立，隨之而來的廢氣也危害人的健康。近年來注重環保及健康的呼聲愈來愈高漲，細懸浮微粒 PM2.5 儼然是全世界必須正視的問題，不管是本地污染源，又或者是鄰國的污染，這些都是在這片土地上的我們必須去面對的，去年至今不時的在新聞上看到「紫爆」這個詞，而我們其實都不知道現在的空氣品質的好壞，多半都是用肉眼來看。

以我們的學校為例，學校位於兩旁是農田及附近有大型化學肥料工廠的環境之下，有時空氣品質變差，但每天插的空氣品質旗號依然沒有變，不禁讓我們懷疑它的準確性及即時性。在市售的空氣 PM2.5 偵測盒子中，都是小型而且數據無法即時觀看，也沒有明顯燈號顯示，不適合學校單位使用，因此我們想做出既美觀又可以隨時知道空氣品質的「空汙報你知-守護你我檢測燈塔」

貳.研究目的

一、運用 Arduino 程式設計及電路設計，創作出能即時測定及更新目前空氣品質之燈號。

二、用明顯的燈號、LED 液晶顯示及語音廣播讓大家可以在遠處就能清楚了解到現在的空氣品質。

三、利用 3D 電腦繪圖軟體設計把 PM2.5 檢測顯示器轉化成裝置藝術品，達到簡單清楚美觀的設計。

四、利用無線網路把裝置上所測到的數據傳送到行政院環保署環境物聯網，讓即將前往裝置所在地的人能提前知道當地的空氣品質。

參.設備及方法

一、研究工具、設備及材料

研究設備及工具

名稱	數量	單位	名稱	數量	單位	名稱	數量	單位
3D 印表機	5	台	銼刀、砂紙、鏟刀	數	個	Solidworks 軟體	3	套
程控電腦	3	台	3M 膠帶	2	捲	Cura 轉文件軟體	1	套
雷射切割機	1	台	游標卡尺	2	把	SD 記憶卡	4	個
電銘鐵及焊錫	1	組	十字螺絲起子	1	把	Arduino 程式語言	3	套

研究材料

名稱	數量	單位	名稱	數量	單位	名稱	數量	單位
PLA 線材 1.75mm	多	捲	電線	1	捲	5mm 壓克力板	3	個
螺絲	多	個	電源供應器	1	個	溫溼度感測器	1	個
燈條及控制模組	2	組	2mm 壓克力板	1	片	PMSA003 粉塵感測器	1	個
LED 螢幕面板	1	個	3mm 壓克力板	1	片	Arduino ESP32 開發版	1	個

二、研究步驟

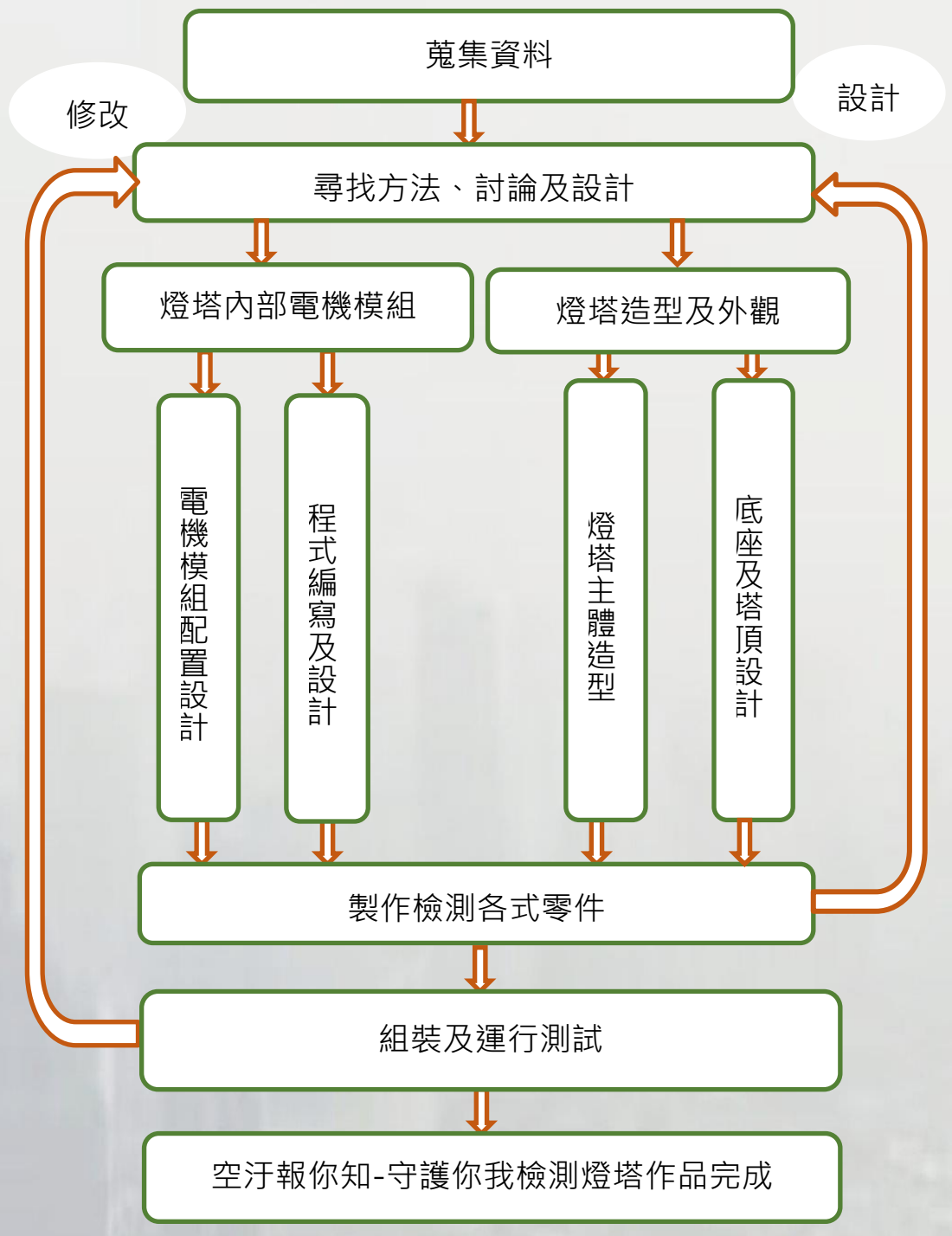
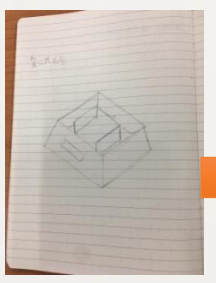


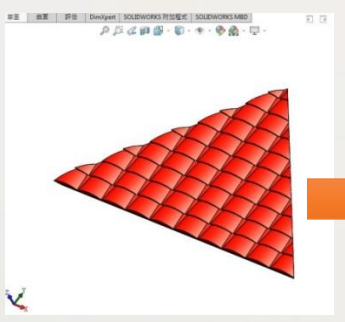
圖 1.研究過程圖

三、製作歷程

1.蒐集資料

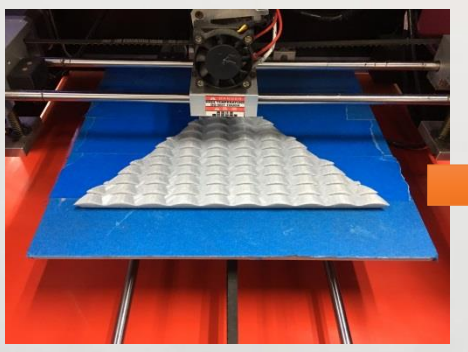


2.手繪草圖

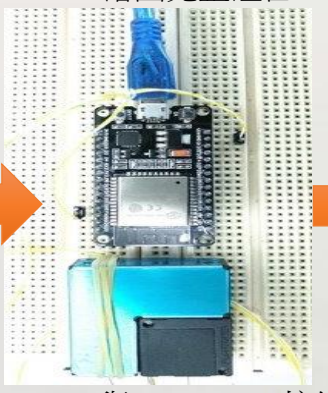


3.元件設計

4. 3D 繪圖完整組合



5. 3D 列印、雷射切割



6. ESP32 與 PMSA003 接線

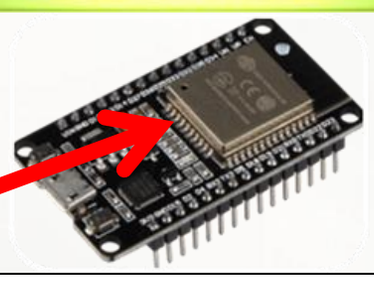
7. 模型組裝

8. 完成及測試

肆.研究結果

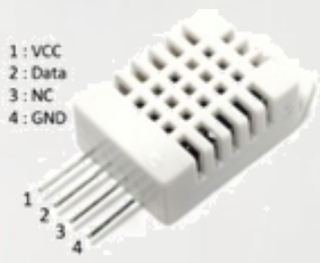
一、電機系統架構及操作方式

```
void showLevel(long x) {
   for (int i = 0; i < 24; i++) strip.setPixelColor(i, 0);
   delay(2);
   long color[5] = {strip.Color(0, 255, 0), strip.Color(255, 255, 0),
strip.Color(255, 128, 0),
strip.Color(255, 0, 0), strip.Color(128, 0, 128)
};
   if (x <= 15.4) {                          //0-15.4  良好(綠色)
      for (int i = 0; i < rank(x, 3, 6, 9, 12); i++)
strip.setPixelColor(23 - i, color[0]);
} else if (x <= 35.4) {                 //15.5-35.4 普通(黃色)
      for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
      for (int i = 0; i < rank(x, 19, 23, 27, 31); i++)
strip.setPixelColor(18 - i, color[1]);
} else if (x <= 54.4) { //35.5-54.4-對敏感族群不健康(橙色)
      for (int i = 0; i < 5; i++) strip.setPixelColor(23 - i, color[0]);
```



ESP32 開發板 Arduino 程式設計用

塔頂做成斜面設計是為了在雨天能夠快速排水之作用，材質使用環保塑料 PLA，無毒且安全，又兼具防水性。



Plantowe(G10)PMSA003 空氣品質感測器及溫濕度感測器、語音播放模組、

LED 燈條及燈示

四排訊息的 LED 液晶面板，更大且顯示更多訊息，PM 2.5、PM 1.0、溫度、濕度及舒適度。

無線網路分享器上傳數據至雲端，進入網站及行動裝置畫面

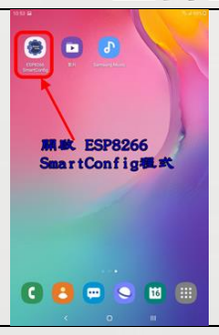
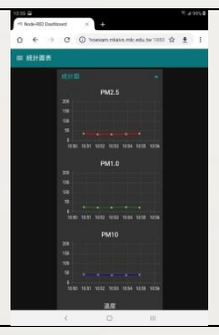
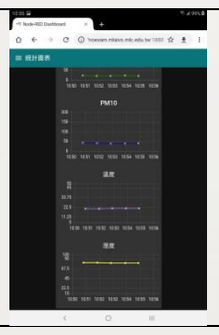
電源供應器

空氣品質檢測塔之狀態色塊

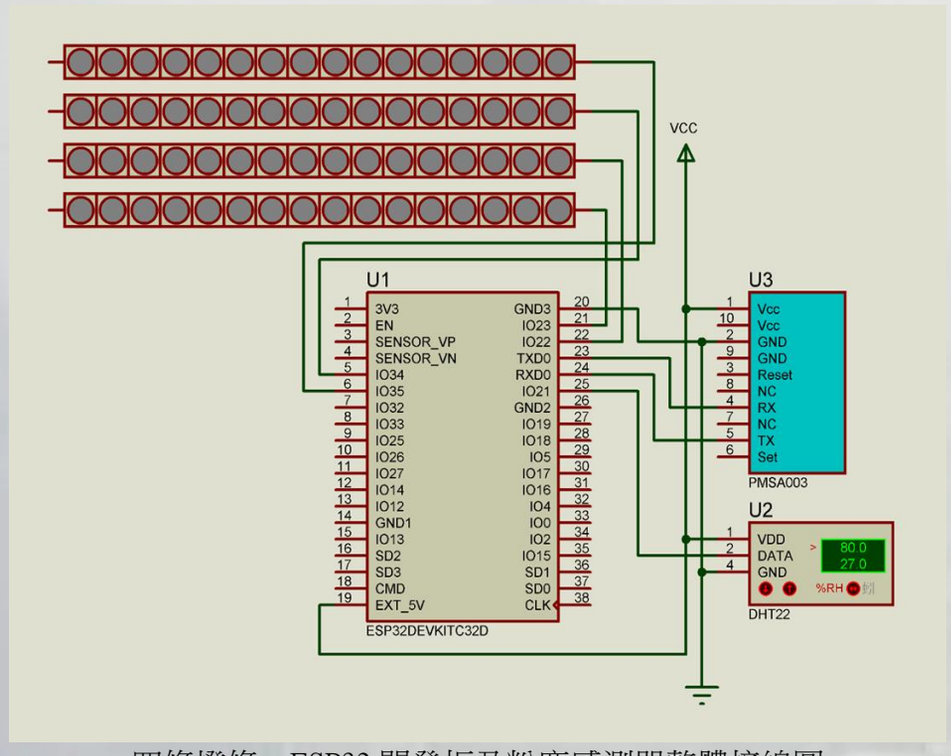
1.作品操作很簡單，電源部分有 USB 插頭及 110V 電源插頭可自由選擇其一，但不可都使用，否則會雙電流輸入損害電機模組。

2.將插頭插入插座，打開開關即可使用。使用手機連上網站即可看到最新上傳數據，或由作品 LED 液晶顯示器、燈條亮的顏色及作品語音就可得知目前當地之空氣品質的各項數據。

二、上傳至物聯網的方式

							
開啟 ESP8266 SmartConfig App	設定 Wifi 連線 IP 位址	連線中.....	成功後 則開啟網站 PM2.5, PM1.0,	觀看 PM10,溫度,溼度數值	即時統計圖表 1	即時統計圖表 2	網站 QR code

三、電路設計及製作



四條燈條、ESP32 開發板及粉塵感測器整體接線圖

ESP32 是一款 WiFi 和藍牙系統級晶片(SoC)，ESP32 完全符合 WiFi 802.11b/g/n/e/i 和藍牙 4.2 的標準，集成了 WiFi/藍牙/BLE 射頻和低功耗技術，並且支持開放性的即時操作系統，以下為 燈條、ESP32 開發板、粉塵感測器等電機元件整體接線

(1) 將燈條分別與 ESP32 開發板之 IO22(PIN22)、IO23(PIN21)、IO34(PIN5)、IO35(PIN6)連接。

(2) ESP32 開發板 EXT_5V(PIN19)、PMSA003 粉塵感測器(PIN1)、溫濕度感測器(PIN1)與電源 5V 連接。

(3) ESP32 開發板 GND3(PIN20)、PMSA003 粉塵感測器 GND(PIN3)、溫濕度感測器 GND (PIN4)與電源負連接。

(4) PMSA003 粉塵感測器與 ESP32 開發板之 RX、TX 連接。

(5) 溫濕度感測器 DATA (PIN2)與 ESP32 開發板 IO21 (PIN25) 連接。

將 ESP32 開發板燒入 Arduino 程式後，再依接線連接正確，即可測試，若是有錯誤則再檢查 Arduino 程式偵錯，直到沒有錯誤，即可與硬體安裝組合。

伍.討 論

一、 作品實測情形

實測數據表

編號	地點	實測圖	時間及細懸浮微粒數據						平均值
1	校門口		時間	14:28	14:29	14:30	14:31	14:32	
			PM2.5	76	86	80	75	77	78.8
2	化學工廠旁		時間	14:47	14:48	14:49	14:50	14:51	
			PM2.5	79	79	76	78	80	78.4
3	宮廟		時間	15:03	15:04	15:05	15:06	15:07	
			PM2.5	160	65	360	68	72	145
4	山區		時間	15:47	15:48	15:49	15:50	15:51	
			PM2.5	64	69	63	65	65	65.2

實測當天室內有空調的教室，PM2.5 的平均值為 48，由上表可知屬「對敏感族群不健康」的橘色等級，而室外 PM2.5 的平均值 63，屬於「對所有族群不健康」的紅色等級。這表示室內加上空調，將有效過濾空氣中細懸浮微粒的數量，即當空氣品質不佳時，為了自己健康著想，應該留在室內且開啟空氣清淨機或空調。

編號 4 山區的 PM2.5 的平均值為 65.2，與當天網路上公告的空氣品質 PM2.5 的平均值 63 的 數值差不多。

編號 1 校門口的 PM2.5 的平均值為 78.8，數值偏高，經過實測觀察，原因為校門口車輛進出頻繁，廢氣較多的緣故。

編號 2 化學工廠附近的 PM2.5 的平均值為 78.4，數值偏高，經過實測觀察，原因為工廠廢氣排氣，提高了細懸浮微粒的數量的緣故。

編號 3 宮廟的 PM2.5 的平均值為 145，數值超高，實測觀察為測定空氣品質指數的地點，太接近宮廟香爐，當香爐的煙飄過檢測燈塔時，數值一度飆高至 300 以上，故燒香的煙 PM2.5 細懸浮微粒的數量非常高，可能長期吸入人體內對健康有不良的影響。

分級	良好	普通	對敏感族群不健康	對所有族群不健康	非常不健康	危害
PM2.5 空氣指標	0~15.4	15.5~35.4	35.5~54.4	54.5~150.4	150.5~250.4	250.5~500.4
狀態色塊	●	■	▲	⬢	◆	✹

二、作品雨天實測

雨天測試 1

雨天測試 2

雨天測試 3

因為作品製作的方向原本就是要放置在室外，必須考慮有防水功能，所以在材質的選用上，使用 PLA 材質來製作屋瓦及底座，來達成防水效果，並在作品完成之後的下雨天，拿去戶外實測，我們將它放置在外兩個小時並每隔一段時間去觀察一次，在這段時間內作品運作完全正常，所以作品防水性測試是成功的。(如左圖)

三、作品與市售空氣品質偵測器之比較表

	空氣品質旗幟	空氣品質檢測儀 1	空氣品質檢測儀 2	空氣品質檢測儀 3	即時空污測定燈塔
圖片					
空氣品質偵測功能	無	PM2.5/甲醛/TVOC 揮發性有機物	PM2.5	PM2.5/溫度/濕度	PM2.5/ PM1.0/ PM10/溫度/濕度
外觀設計	不美觀	佳	佳	佳	優
數據更新頻率	1 天	5 分鐘	3 分鐘	1 分鐘	1 分鐘
語音功能	無	超標有警告聲	無	無	有
明顯燈號	無	無	無	無	有
傳數據至雲端	無	無	無	無	有
機體顯示 測得數據	無	有	有	有	有
室外防水性	無	室內使用	室內使用	室內使用	室內室外防水俱佳
總評	差	普通	普通	佳	最佳

陸.結 論

經由以上敘述，我們得到以下結論:

一、成功利用電路與電路板的連結，達成即時測定空氣品質的目的。

二、完成用明顯的燈號、LED 液晶顯示及語音廣播讓民眾可以在遠處就能清楚了解到現在的空氣品質(AQI)之狀態。

三、利用電腦繪圖設計出簡潔俐落的外觀，讓路過的民眾、學生能在遠處或白茫茫的霧霾中看清楚燈示，宛如燈塔一般，了解空氣品質及 PM2.5 狀況，能即刻防患，使大家知道今天適不適合戶外活動，為環境保護及學生健康盡一己之力。

四、成功利用程式編寫與無線網路連接將數據回傳，可讓民眾用行動裝置及網路知道現在溫度、濕度、PM1.0、PM2.5、PM10 空氣品質之詳細數據。