第十六屆旺宏科學獎

成果報告書

參賽編號：SA16-465

作品名稱:粒粒皆吸附－便攜式節能 PM2.5 淨化器之研究

姓名：許躍蘭

關鍵字：PM2.5、細懸浮微粒、Arduino

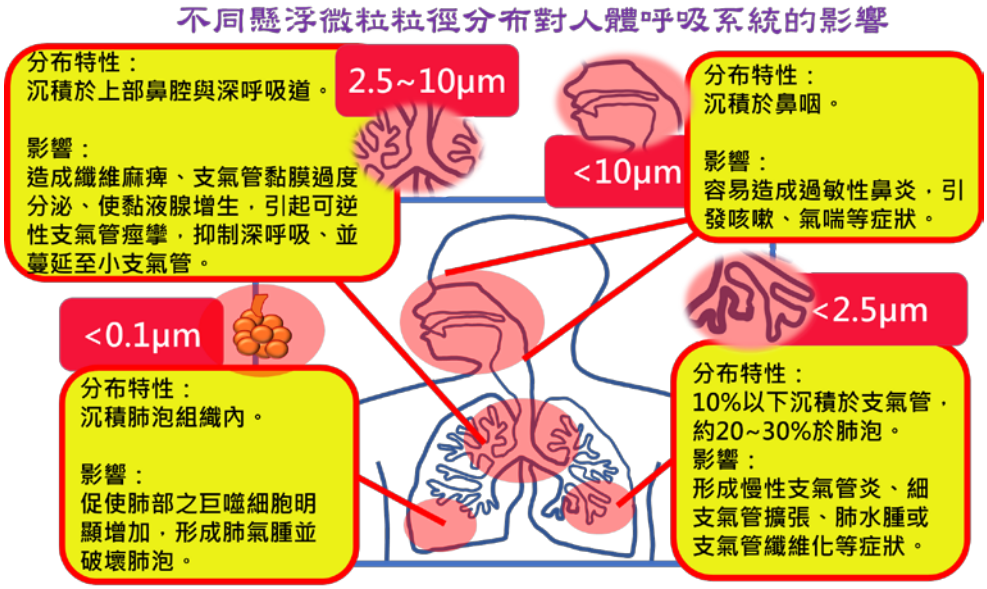
粒粒皆吸附－便攜式節能 PM2.5 淨化器之研究

摘要

本次研究中，我使用 Arduino 單晶片微控制板自製了細懸浮微粒偵測器，用來測量 PM2.5 的穿透率，以達量化懸浮微粒的效果，實驗中發現，使用摩擦起電的方式因為電場太小無法有效吸附懸浮微粒，接著使用平行電網，發現吸附能力與電壓與總表面積有關，因而改採電纖維通以高壓電的方式進行濾淨，其吸附率約達 52%。在整個實驗中我使用 Arduino 單晶片微控制板控制數據的擷取、分析，與節電系統的調控，做出了一個兼具輕便、低成本、與節電環保的懸浮微粒清淨器。

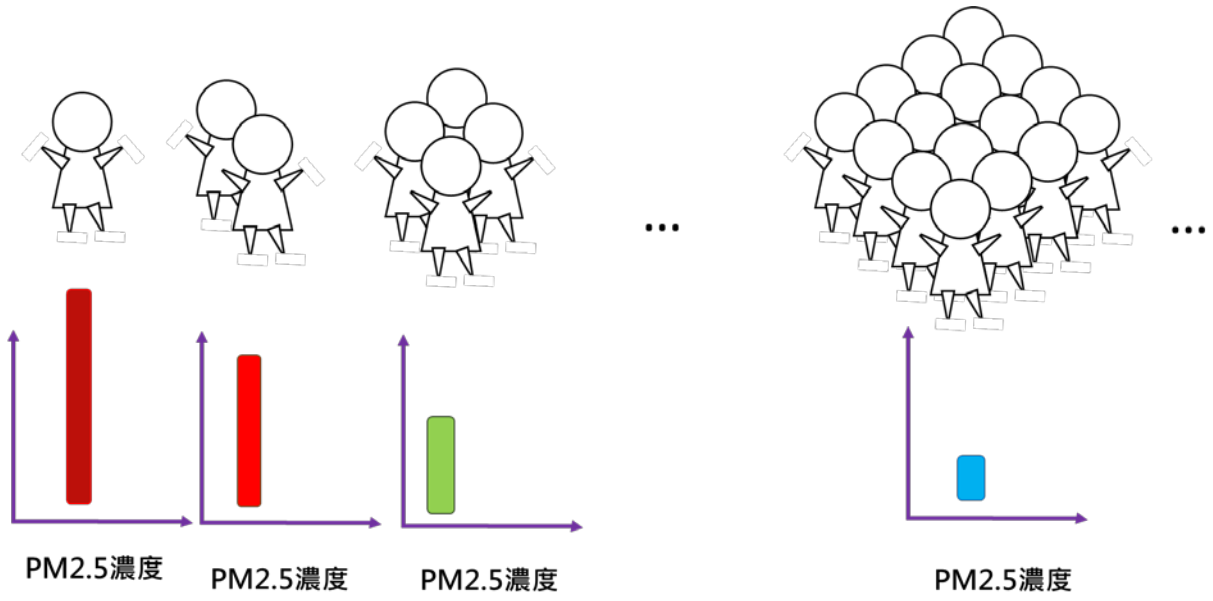
壹、研究動機

在現今高度工業化的時代，許多高科技產品的運作與製作過程會產生不可計量的氣體副產物，每天上下學的過程中我們無時無刻吸入著充滿懸浮汙染物的空氣，讓我每每感到呼吸到的不適—不管是打噴嚏或是咳嗽，都造成我們生活上許多的不便，也會對健康造成極大傷害。現在普遍使用空氣濾淨器來濾淨空氣，然而現在的空氣濾淨器不僅攜帶不便、成本高昂，還非常耗電，於是我開始思考是否可以使用其他方式來吸附懸浮微粒。我便想到以物理課本上教到的摩擦起電與介電力的原理為基礎，並試圖推廣至利用尖端放電的方法，來自己製作節能且體積小的空氣濾淨器，並找出最佳的濾淨方法，以清除這讓所有人身陷其害(如下圖所示)的嚴重汙染。



貳、研究目的

本研究將以創新的方式製造一個人人可攜帶的空氣濾淨器，並使其擁有省電及低成本的特性，期望能達到人手一台「便攜式節能 PM2.5 淨化器」的普及率，使每一個人都成為一個會走動的空氣濾淨器，不僅可以淨化自己周遭的空氣，更可以大面積的過濾空氣中的 PM2.5 懸浮微粒(示意圖如下)。



以下列出本研究的六個重要目標

一、自製 PM2.5 濃度偵測器。

二、利用靜電濾材吸附懸浮微粒。

三、使用高壓靜電金屬網吸附懸浮微粒。

四、設計負離子吸附系統。

五、回饋節電系統。

六、PM2.5 吸附器的應用。

本報告已先進行了部分的前期實驗：包含實驗器材的架設、PM2.5 濃度偵測器的初步製作、及簡易的 PM2.5 負離子吸附器等.....。(如下流程圖所示)

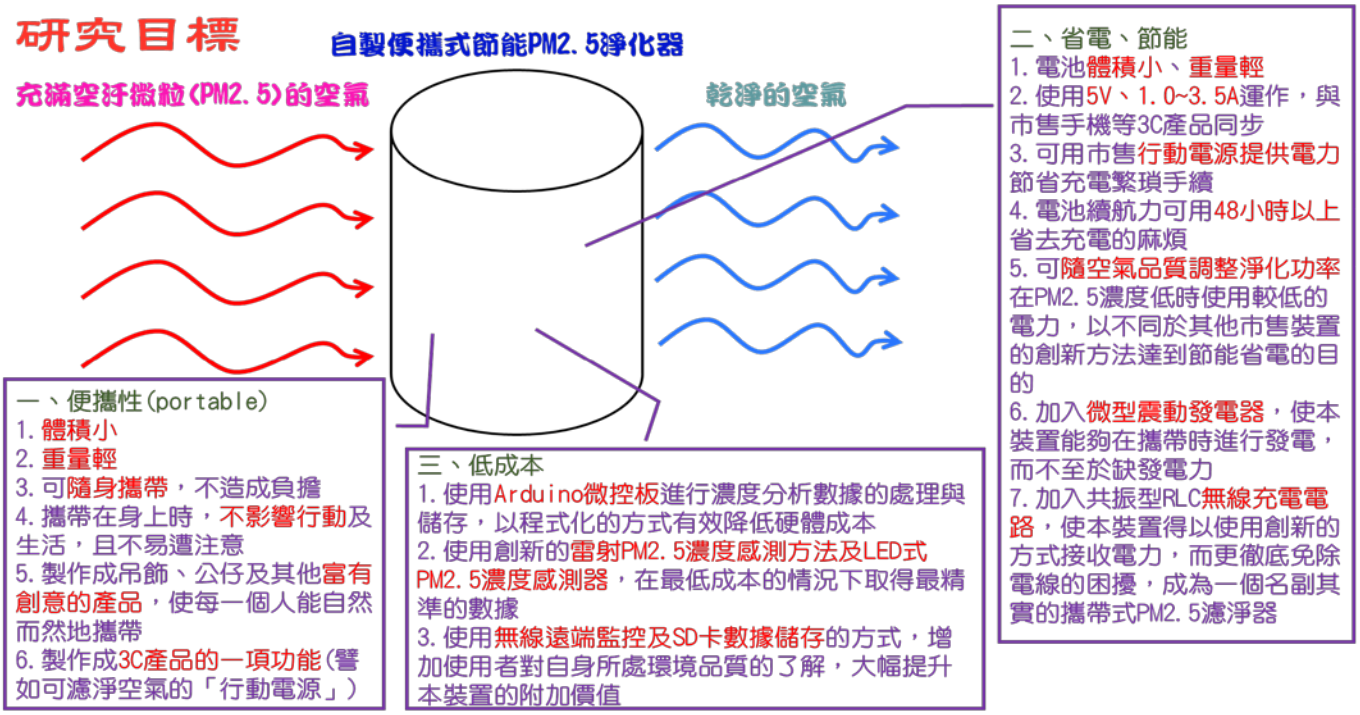
●詳細說明：

一、自製 PM2.5 濃度偵測器：以兩種便宜、輕便且富有創意的方法（雷射及 LED）來偵測 PM2.5 的濃度，在本報告中我先進行了雷射偵測 PM2.5 濃度的實驗，並使用夏普販售的 LED 式 PM2.5 偵測器來進行初期的實驗。之後我將繼續研究自製的雷射及 LED 式 PM2.5 偵測器，並提高其準確度，同時減小其體積，以創新且低成本的方式測量 PM2.5 濃度。

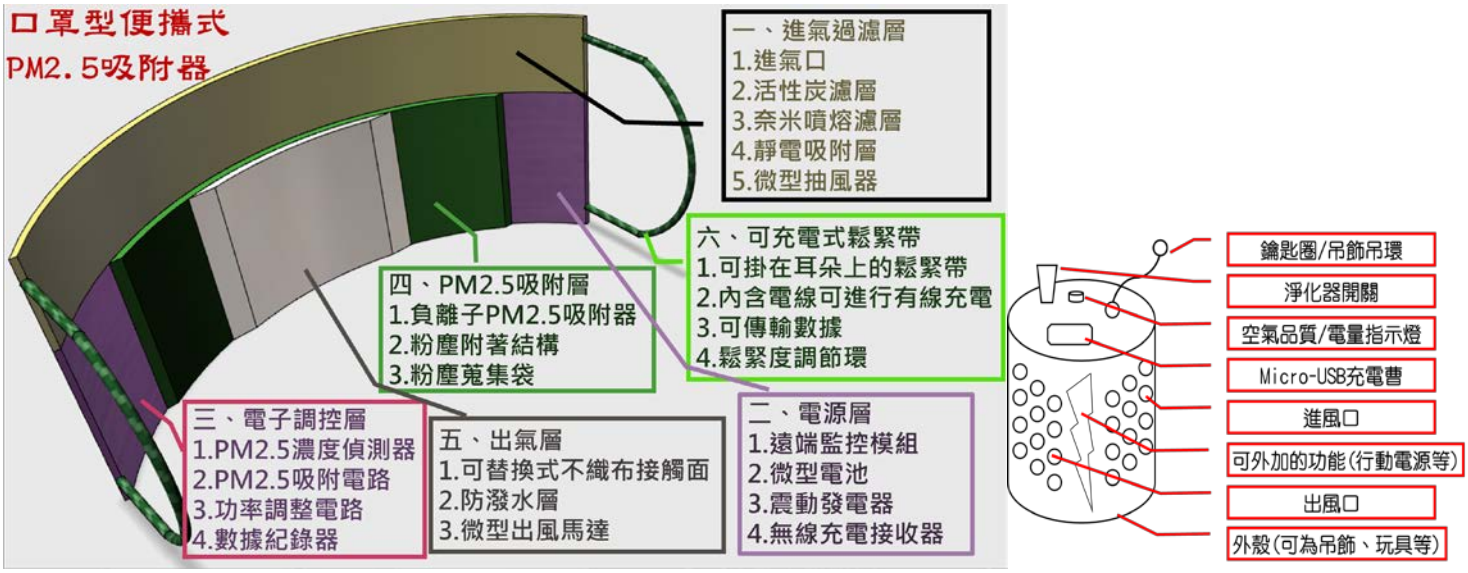
二、利用靜電濾材吸附懸浮微粒：使用可帶大量靜電荷的材料，並將其互相摩擦產生大量異性電荷，以吸附空氣中的懸浮粒子。在本報告中，我已先對部分材料(如：靜電紙...等)進行摩擦吸附測試；在往後的實驗中，我將改進摩擦帶電的裝置，並使用更多元的靜電材料進行 PM2.5 的吸附測試。

三及四、使用高壓靜電金屬網吸及負離子吸附系統附懸浮微粒：我將嘗試的各種吸附 PM2.5 的方式，包括通以高壓直流電及尖端放電等方式，並使用創新的自製電路使吸附率達到最佳。在本報告中已經先測試了幾種電路，其中包括效果頗佳的碳纖維毛刷負離子吸附法。

五及六、PM2.5 吸附器的應用：將本裝置的各項性能優化並符合便攜、低成本、節能的要點。（如下圖所示）



更進一步的，我將進行各項數據的定量分析，以近一步去了解並調整本裝置的電路及軟體設計，在最後我期望做出一個口罩型 PM2.5 吸附器(如下左圖)並把他推廣到各種用途中(如下右圖)。



三、研究方法及討論

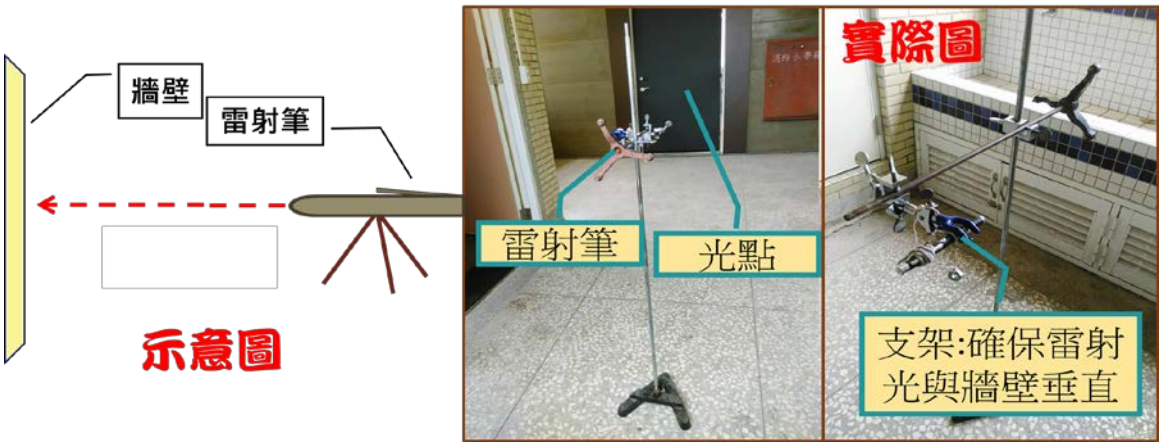
研究一、使用雷射光進行 PM2.5 的測量

探討 1、設計雷射 PM2.5 測量器

(一)目的：探討以雷射光進行不同空汙濃度測定的可能性

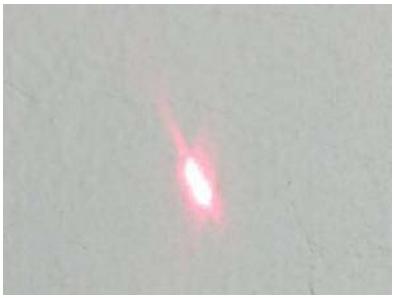
(二)步驟：1.將雷射筆垂直射向屏幕。

2.雷射筆測量裝置如下圖。



3.分別在不同 PM2.5 濃度的日子觀察雷射光投影至牆壁之情形。

(三)結果：1. 雷射光投影情形

	
平日	紫爆

(四)討論：1.在空氣懸浮微粒濃度較高時，因為光線易散射，所以投射到牆壁上的面積較大。

2.為了進行更精確的測量，我在查閱了網路資料與教科書後，決定以照度為測試的標準，而非面積。

3.因為 10m 時的光度變化最明顯，所以用 10m 為最佳距離。

探討 2、雷射測照度與 PM2.5 濃度的關係

(一)目的：直接從照度推估當下 pm2.5 濃度

(二)步驟：1.將雷射筆垂直對準照度計，距離 10 公尺。

2.利用數位攝影機拍下照度的變化。

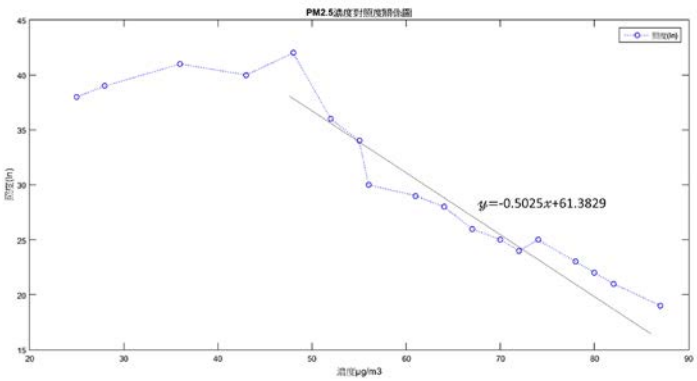
3.利用威力導演將影片分成一秒六格，並將數據進行平均

4.與氣象局之 PM2.5 數據進行對照，分析各種不同空汙濃度下的照度

5.擷取在各個不同濃度的照度值，並繪製成表

(三)結果：1. PM2.5 濃度對照度關係圖

2.濃度 45μg/m^3 以下因為濃度稀薄、散射較小、數據變化不大，故我將濃度 45μg/m^3 以上的數據進行線性回歸得到方程式



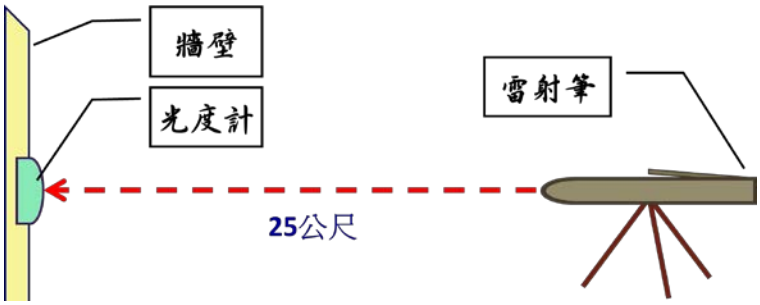
y=-0.5025x+61.3829

(四)討論：1. 45μg/m^{3}以下數據變化較小可能造成日後實驗的障礙，但我仍可以方程式 y=-0.5025x+61.3829 推論出當下的懸浮微粒濃度。

探討 3、測量距離與雷射照度的關係

(一)目的：找到運作此系統的最佳距離

(二)步驟：1.將雷射筆與光度計架設如下圖。

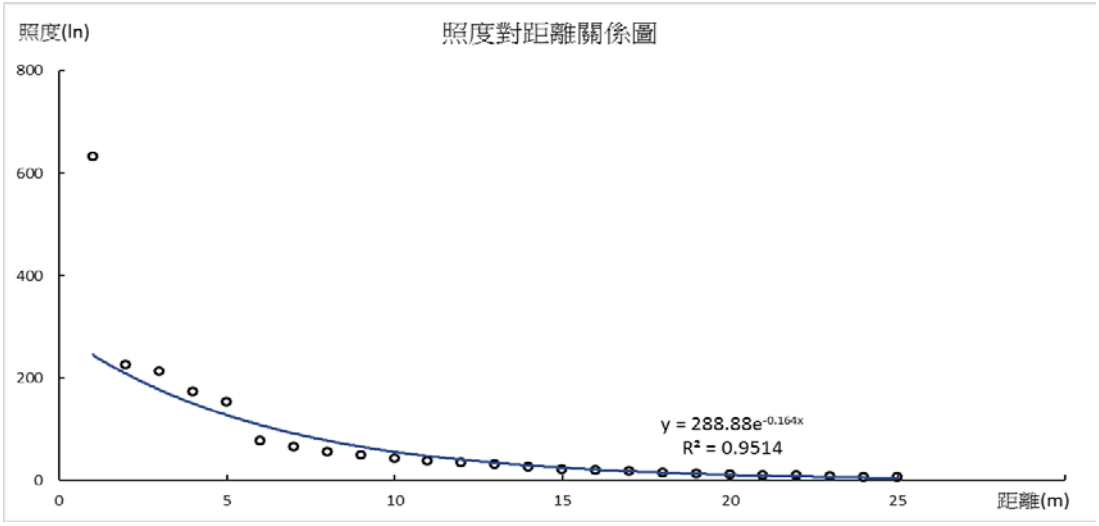


2.利用數位攝影機拍下照度的變化。

3.利用威力導演將影片分成一秒六格，並將數據進行平均

4.與氣象局之 PM2.5 數據進行對照，分析各種不同空汙濃度下的照度

(三)結果：1. 照度對距離關係如下圖



(四)討論：1.經由文獻探討了解當距離增為 2 倍時，照度會變為 1/4，為平方反比關係。

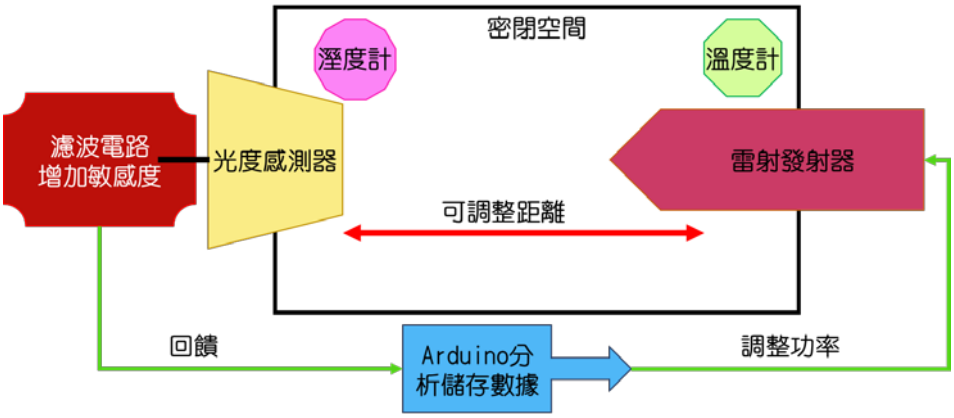
2.在本實驗中距離 1~15 公尺範圍呈現明顯的平方反比關係，和文獻上的理論相符，但 15 公尺後數值無明顯變化，我們推測因為照度計靈敏度不足，故無明顯變化。

3.經討論後我發現會引起散射的懸浮微粒除了 PM2.5 粒子之外更有其他懸浮微粒影響，所以我便想要設計了一個機制來做更精確的測試，並提供一個回饋的機制來調整雷射發射器的功率，以避免測值過高，並加裝濾波電路與濕、溫度計以探討其他因素的影響，並期許做出一個具獨特創新的雷射式 PM2.5 偵測器。

探討 4、自製雷射型 PM2.5 偵測器

(一)目的：使用雷射與感光元件偵測 PM2.5 微粒濃度

(二)步驟：1.將雷射筆與光度計架設於一方型塑膠盒內(示意圖如下)。



2.摩擦內部使塑膠盒內部材質不反光

3.利用程式調整雷射發射功率

4.紀錄光度感測器的數值

(三)結果：

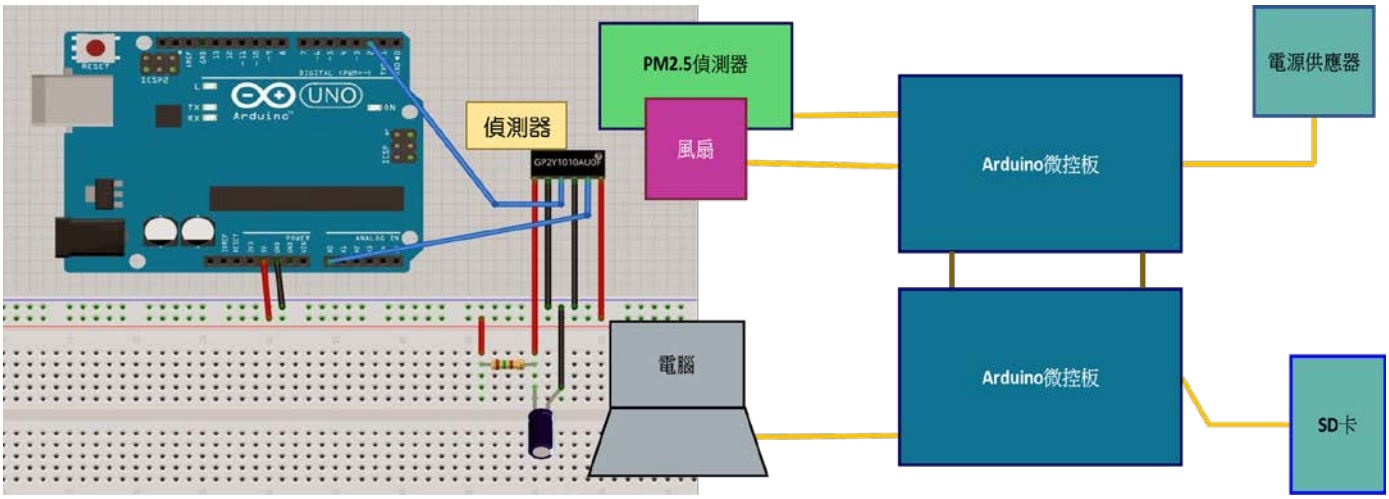
(四)討論：

研究二、利用 Arduino 單晶片微控制板自製 PM2.5 濃度偵測器

探討 1、LED 式 PM2.5 偵測器的數據擷取

(一)目的：擷取並計算 PM2.5 偵測器測得之數據

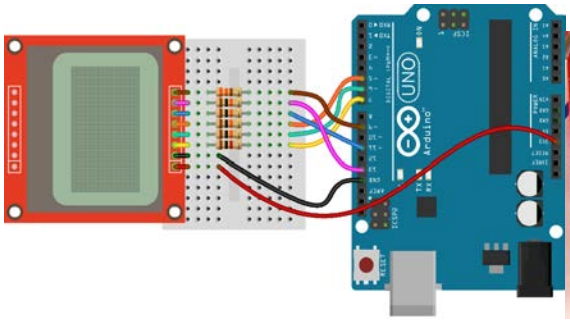
(二)步驟：1.使用夏普 PM2.5 偵測器(型號：GP2Y1010AU0F)並連接如下圖。

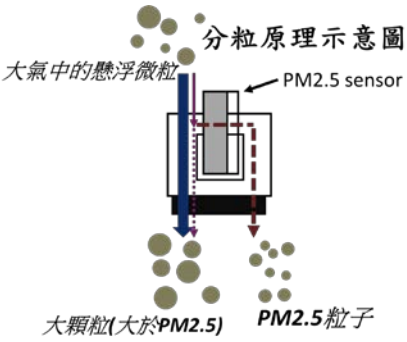


2.使用類比腳位量取 PM2.5 偵測器的電壓，並經由以下程式碼轉換為濃度。

```
calcVoltage = voMeasured * (5.0 / 1024.0);//先算出電壓的值
  dustDensity = 0.17 * calcVoltage - 0.1;//再轉換為 PM2.5 濃度
```

3.將 LCD 面板以下左一圖的方式連接到數位腳位，並使用附錄中的程式碼使其顯示即時的數值。





(三)結果：1.可以在 LCD 面板看見即時的空氣懸浮微粒數據，如上右一、右二圖所示。

2.這個值的單位是常見的 μg/m^3，測量間隔為 1 秒鐘。

(四)討論：1. 在這個研究中，我發現濃度數值時常有浮動不定的情況發生，經檢查後發現使用麵包版容易產生干擾與雜訊，我經指導老師建議將 PM2.5 偵測器焊接於 Arduino 版上。

2.針對夏普 LED PM2.5 偵測器結構與技術報告仔細的分析後，我發現這個偵測器使用了 LED 燈取代了雷射，同樣使用照度計來測量空氣懸浮微粒的光透度；另外此感測器內有小風扇，能定時定量抽取外界環境空氣，並利用慣性原理分離 PM2.5 和較大微粒。較大微粒會通過上結構圖(上右)中左方實線箭頭通道回到外界，而 PM2.5 會經過右方虛線箭頭，內有 LED 與光感測器的通道(LED 光被 PM2.5 散射，經由光感測器感測)。一開始感測器會先檢測 100 次(1 秒/次)未開風扇、氣流穩定之外界空氣的 PM2.5 濃度(空氣中粒子會沉澱不飄動)。測完 100 次後，內部小風扇開始運轉，感測器每 1 秒吸入定量氣體，測出散射的 LED 受光面積，帶入上一個實驗(探討一)的公式即可輸出 PM2.5 濃度。

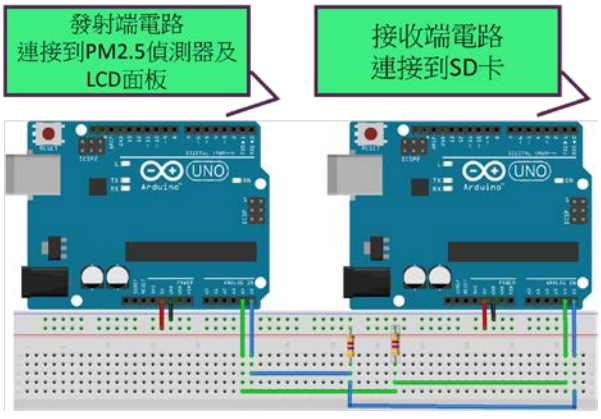
3.我除了可以從 LCD 螢幕知道即時的懸浮微粒濃度外，也可以將 Arduino 版連接上電腦並由 Arduino 編譯器之監控視窗得知數據，但此兩種方法皆不易記錄數據，同時使用電腦進行監控更造成了不方便攜帶的困擾，我便想要將 Arduino 單晶片微控制板連接上記憶卡以方便記錄數據。

探討 2、數據的分析與儲存

(一)目的：將經處理後的數據直接存取於記憶卡中，便可進行長時間監控

(二)步驟：1.將兩塊 Arduino 單晶片微控制板如下圖連接(使用類比腳位進行兩塊 Arduino 單晶片微控制板的通訊)

2.將下方程式碼(SD 卡寫入函式)寫入上圖右側(連接到 SD 卡)的 Arduino 單晶片微控制板




```
void receiveEvent(int howMany)
{
  while (Wire.available())
  {
```

取得所需記憶體空間

讀取 I2C 回傳值

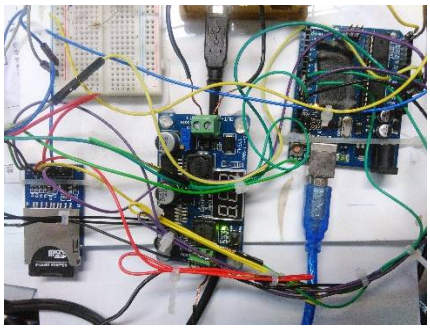
```
    incomingByte = Wire.read();
  datafile=SD.open("pm25data.txt", FILE_WRITE);
    if (datafile){;
    }else{
Serial.println("open error");
    }
if (datafile){
   int trans=incomingByte;
  datafile.println(trans);
     Serial.print(trans);
}else{
      Serial.println("FILEWRITEERROR");
     }
 datafile.close();
}
}
```

開啟 SD 卡檔案

確認可寫入

寫入

關閉檔案



(三)結果：1.連接完成後如右圖

2.可以從 SD 卡擷取長時間的數據

3.發現在長時擷取數據的過程中時常有傳輸不完全的地方或雜訊

(四)討論：1. 在進行本實驗時原本預定使用一個 Arduino 版同時達到數據擷取、分析、儲存、以及 LCD 顯示，但是發現時常無法正確寫入，經程式除錯發現：Arduino 單晶片微控制板之動態記憶體僅 2kb，但讀寫 SD 卡之檔案需其總記憶空間之 54%，故常發生記憶體不足現象。

2.為了解決上述問題，我使用了第二塊 Arduino 單晶片微控制板來進行 SD 卡寫入，並且使用其內建的 I^2C 通信機制、透過類比腳位來傳輸數據。

研究三、校正 PM2.5 濃度偵測系統

探討 1、使用線香產生人工懸浮微粒

(一)目的：使用線香生成不同濃度的懸浮微粒以利定量分析

(二)步驟：1.將線香放置於電扇與測試箱中間。

2.改變線香的數目(調控進入測試通道粒子濃度)。

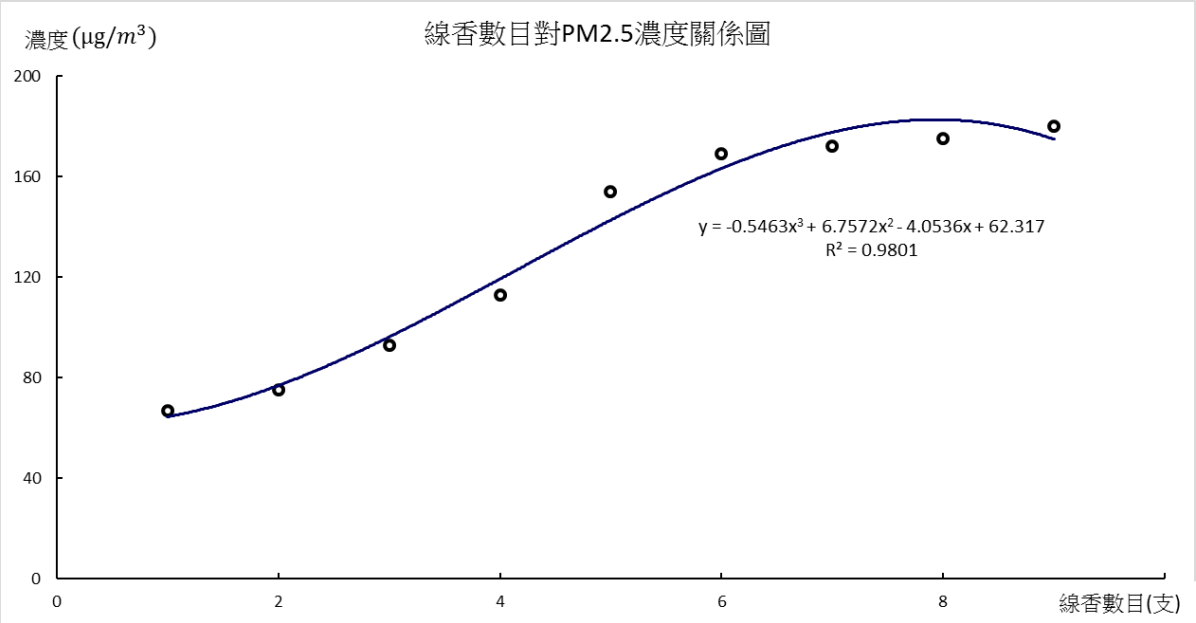
3.使用 PM2.5 偵測器在蒐集數據一小時

4. 分析吸附程度

(三)結果：1.微粒濃度隨線香數增加而遞增，但最終濃度將趨於一定值，而不受線香數目之影響(當線香數極大時)

2.可透過控制線香數目達到人工控制懸浮微粒濃度之效果(但線香數不可太多)

3.數據如下圖：



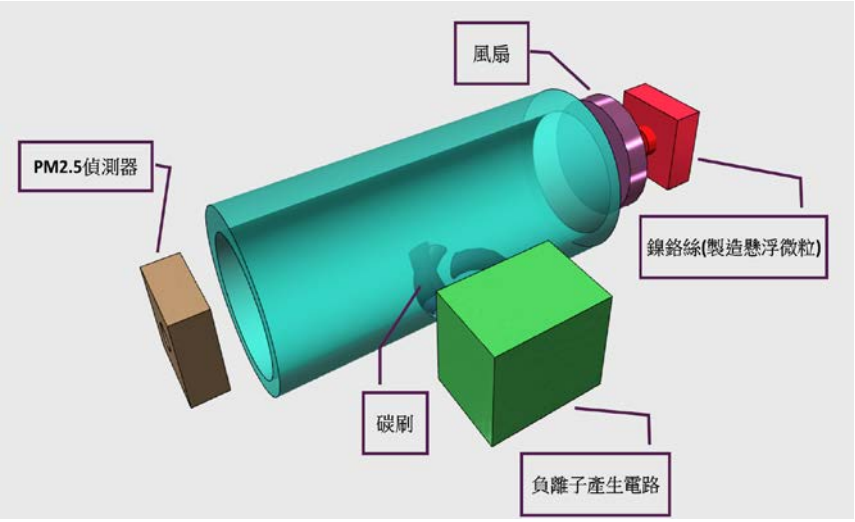
(四)討論：1.線香數目愈高，PM2.5 濃度愈高，因此可在一定範圍內增加 PM2.5 濃度，以利實驗控制變因

2.雖線香可產生 PM2.5 但由於不能精確控制，我們便想採用其他的懸浮粒子產生方法。

探討 2、使用鎳鉻絲產生人工懸浮微粒

(一)目的：利用鎳鉻絲生成不同濃度的懸浮微粒以利定量分析

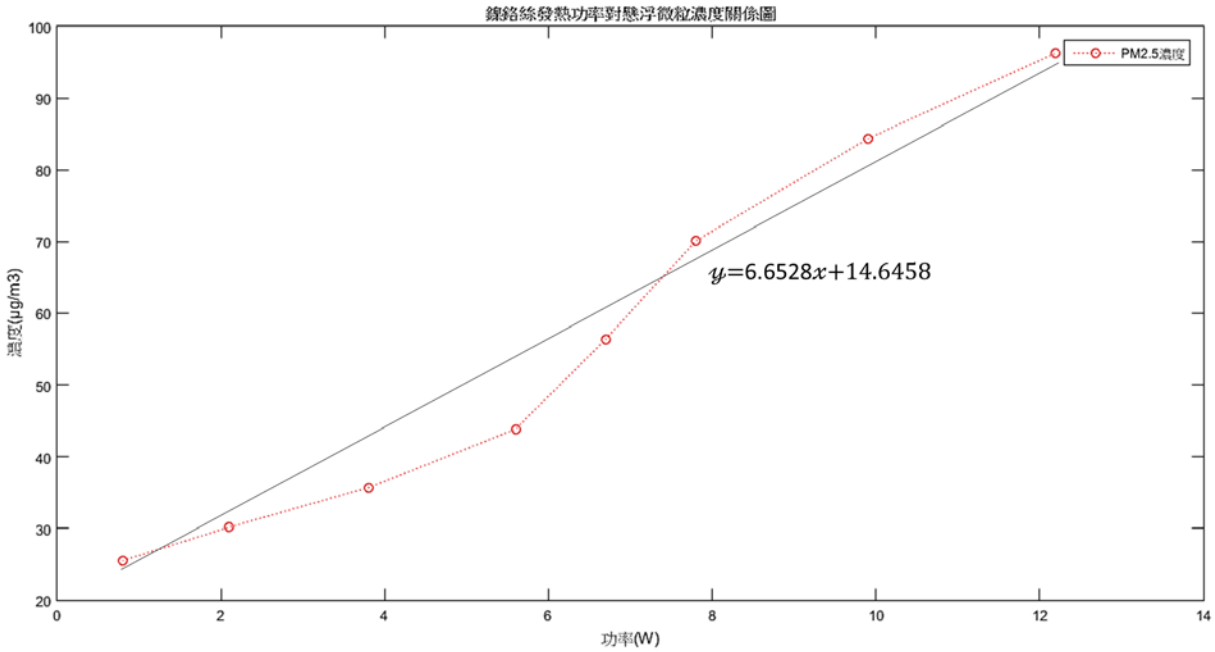
(二)步驟：1.將點菸器尾端的鎳鉻絲拔出，如下右圖



2.通以不同電流使其發煙，以產生懸浮微粒。

2.並以 PM2.5 偵測器測量

(三)結果：1.鎳鉻絲發熱功率對懸浮微粒濃度關係圖如下



(四)討論：1.發熱功率越高，懸浮微粒的濃度也會隨之提升，並呈現一線性變化。

2.因此可以使用鎳鉻絲來進行懸浮微粒濃度的定量實驗。

探討 2、風速對 PM2.5 偵測系統的誤差

(一)目的：探討風速對 PM2.5 濃度偵測器產生的誤差

(二)步驟：1.將數個 PM2.5 偵測器放置同一密閉空間中

2.使用一煙霧產生器產生微量懸浮微粒

3.調整每一偵測器之風扇並測量其風速

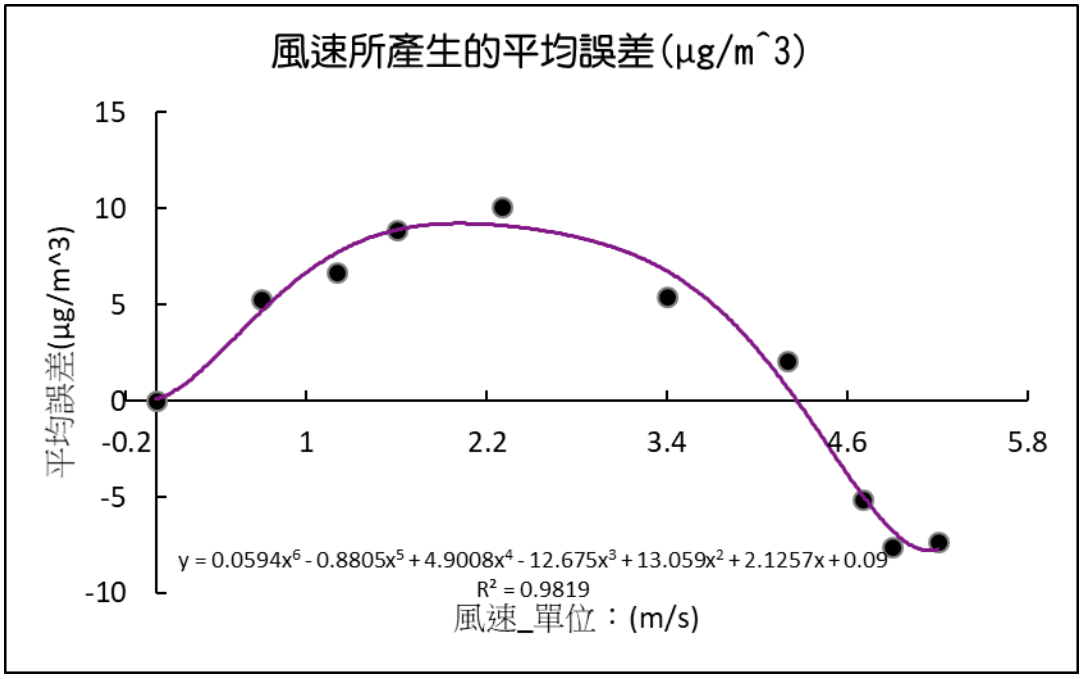
4.測量並記錄其偵測誤差

(三)結果：1.統計在每一風速狀況下之誤差值

2.將測量值進行平均並取得平均值

3. 我們使用風速為零的 PM 2.5 濃度作為基準值來算平均值的誤差

4.繪製圖表如下



(四)討論：1.我發現到風速比較小的時候，隨著風速的上升誤差值會隨之上升，也就是說當風速沒有很大的時候 PM 2.5 的濃度會隨著風速的上升而提高，但是當風速達到一定程度的時候我們可以發現到，誤差值非常迅速地下降，甚至下降到風速「0」的基準值之下。

2.經過一連串的分析，我推測這樣的結果是因為，當風速很小時，懸浮微粒無法均勻的分佈在測量的空間中，而導置偵測值並沒有很高，我們在這個時候通以一些流動的空氣，可以讓懸浮微粒均勻的分佈，造成偵測值變得更準確，當我們繼續增高風速，因為懸浮微粒通過的速度太快了，導致 PM 2.5 濃度偵測器無法準確偵測裡面的懸浮物質，而導致濃度的下降。

3.另外我們也發現到當風速中等時，由於風速的增加使得慣性濃度分粒機制失效，導致有可能偵測到大量的 pm10 粒子。

探討 3、探討風扇功率與風速的關係

(一)目的：為達耗電量最小化，探討在最低用電量達到最佳準確濃度偵測數值的方法

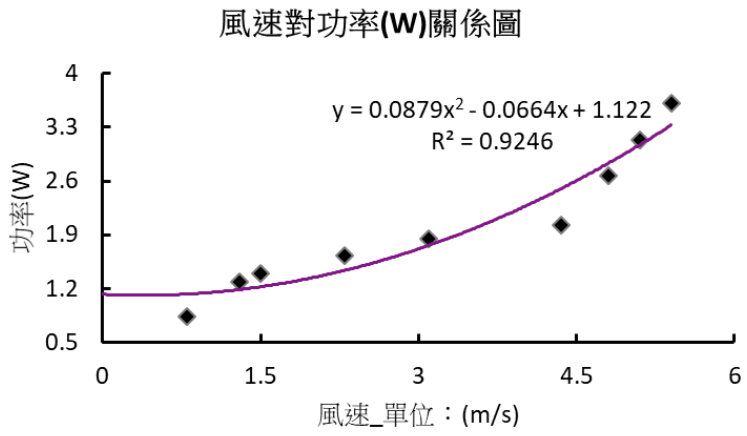
(二)步驟：1.開啟風扇馬達進行抽風

2.量測其產生之風速

3.以示波器測量其電壓與電流

4.計算風速下之功率並繪製圖表

(三)結果：1.風速對功率關係圖大致成二次曲線之趨勢



2.在風速 2.5~3.5 m/s 之間耗電功率曲線斜率極小

(四)討論：1.中等風速時耗電量較風速大時小的許多，並且與上一實驗之許可誤差範圍吻合

2. 為達耗電量最小化，擬取風速約 3m/s 當作之後實驗的抽氣風速

3.需著手測量風速對吸附效果之鷹想方可確定風速之最佳值，於是我進行下面的實驗

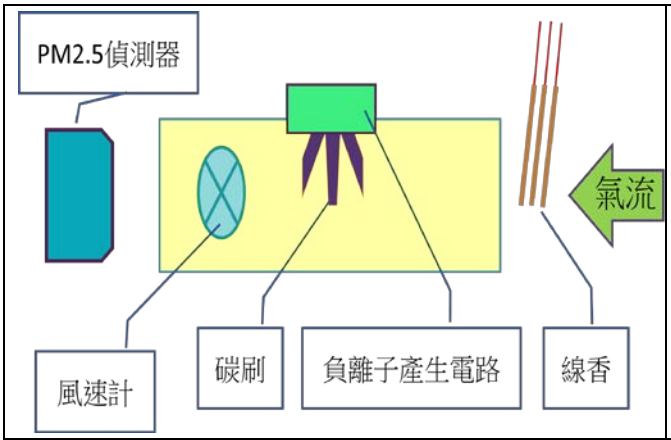
探討 2、風速對吸附率的影響

(一)目的：改變風速，探討風速對於吸附率的影響

(二)步驟：1.開啟偵測系統和電路

2. 改變風扇的風速

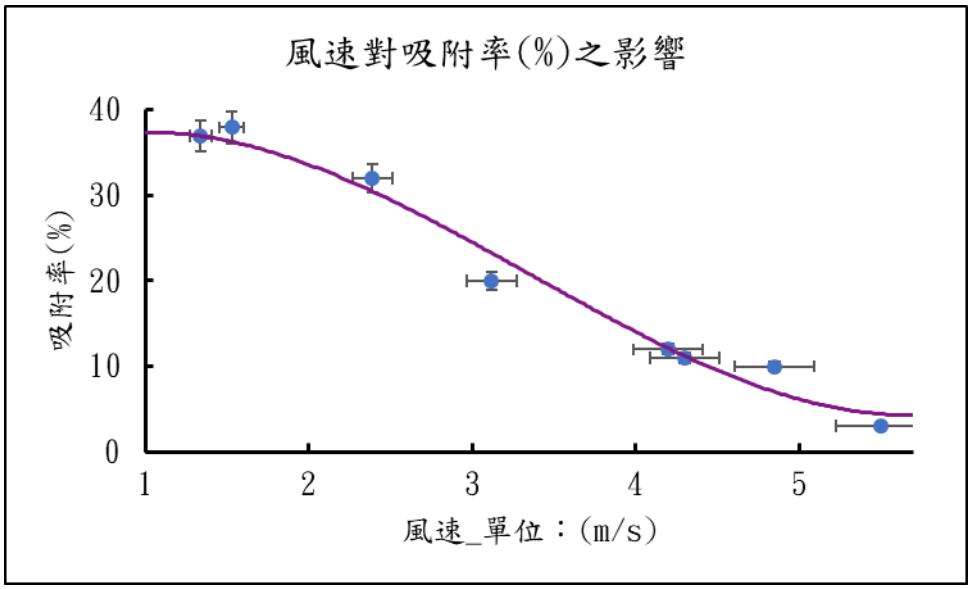
3.量測 pm2.5 濃度變化



圖解

實際圖

(三)結果：1.發現隨著風速增強，微粒之運動速率也隨之增加，使得靜電不易吸附、效果大幅降低(如下圖表所示)



2.偵測器測得之數據在無汙染物干擾(正常情況)下，淨化前後數值無明顯改變；在污染物大量的情況下，淨化前後數值差異亦無明顯改變。

(四)討論：1.風速上升，則吸附率下降。

2.汙染物隨著風速提升而與金屬網接觸時間減小，故吸附效果減少，穿透率提升。

探討 3、校正自製 PM2.5 濃度偵測器

(一)目的：將背景值以及其餘外界干擾因素去除，以得到更精確的數據

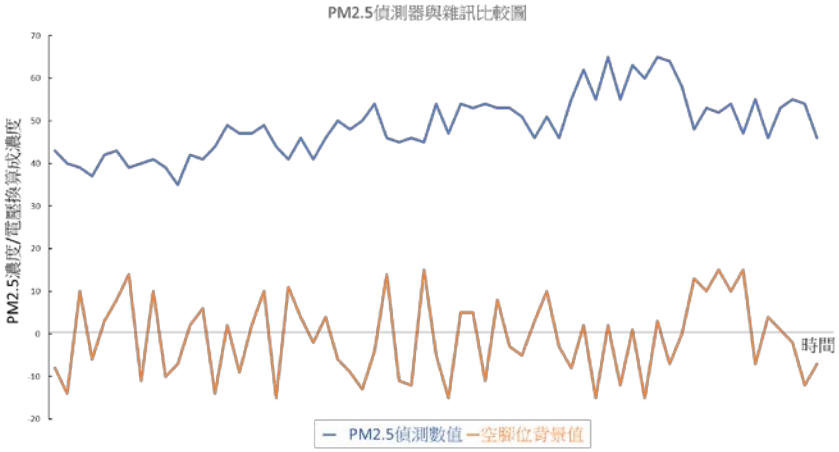
(二)步驟：1.確定 1 號類比腳位及 2 號類比腳位為空腳位。

2.直接偵測空腳位背景值(電壓)換算成 PM2.5 濃度並與偵測值比較製表

3.將數據擷取的 Arduino 單晶片微控制板中，其程式加上以下這段程式碼，將原本的偵測值(電壓)減去空腳位(1 號及 2 號)的平均電壓類比腳位所讀取到的背景值(平均電壓)。

```
voMeasured =
analogRead(measurePin)-((analogRead(blankping1)+analogRead(blankping2)+analogRead(blankping3))/3); // read the dust value
```

(三)結果：1.PM2.5 偵測器與雜訊比較圖



(四)討論：1.經過校正後，測值的變動雖幅度大幅降低，但仍有許多誤差值得我探討，我打算再向風速、濕度、溫度等方向進行探討，並研究這些變因對空汙濃度的影響，故在此我設計了一些實驗(如下表)。

1.風速	1a.風速對 PM2.5「測值」的影響	1b.實驗並理論計算風速對 PM2.5「實際濃度」的影響
2.濕度	2a.濕度對 PM2.5「測值」的影響	2b.實驗並理論計算風速對 PM2.5「實際濃度」的影響
3.溫度	3a.溫度對 PM2.5「測值」的影響	3b.實驗並理論計算風速對 PM2.5「實際濃度」的影響

探討 4、分析自製偵測器數據與環保局測值之誤差

(一)目的：了解自製偵測器之精確度並加以檢討缺失

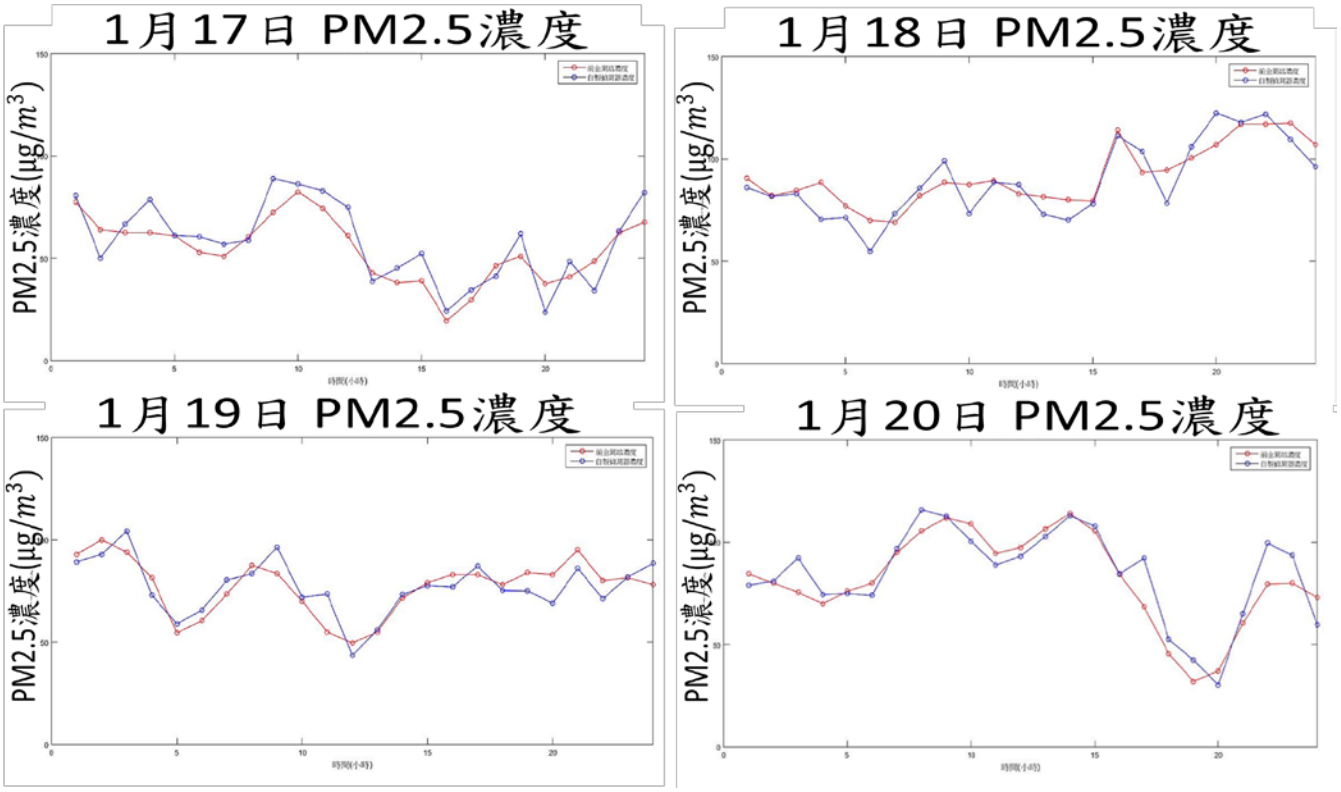
(二)步驟：1.將所蒐集到一周內數據和氣象局數據進行比對

2.繪製濃度關係圖以更深入了解誤差

(三)結果：1.PM2.5 濃度和前金測站比較曲線(如下表所示，僅取 4 天為例)

2.所取得數據和環保局資料相當吻合，但數值偏高且波動幅度較大。

紅色的數據點為前金空汙測站所測得的數據，藍色的數據點為我所測得的數據



(四)討論：1.自製測站附近較為市中心空汙情況較前金站嚴重，故和環保局測站之數值有少許誤差，並偏高。

2.我將自行製作一個 LED 式的 PM2.5 偵測器取代市售的，並校正其誤差，使本裝置更為完備。

研究三、利用靜電濾材吸附細懸浮微粒

探討 1、靜電產生（摩擦起電)裝置的設計

(一)目的：以摩擦起電的方式提供濾材靜電，以達到吸附微粒之效果

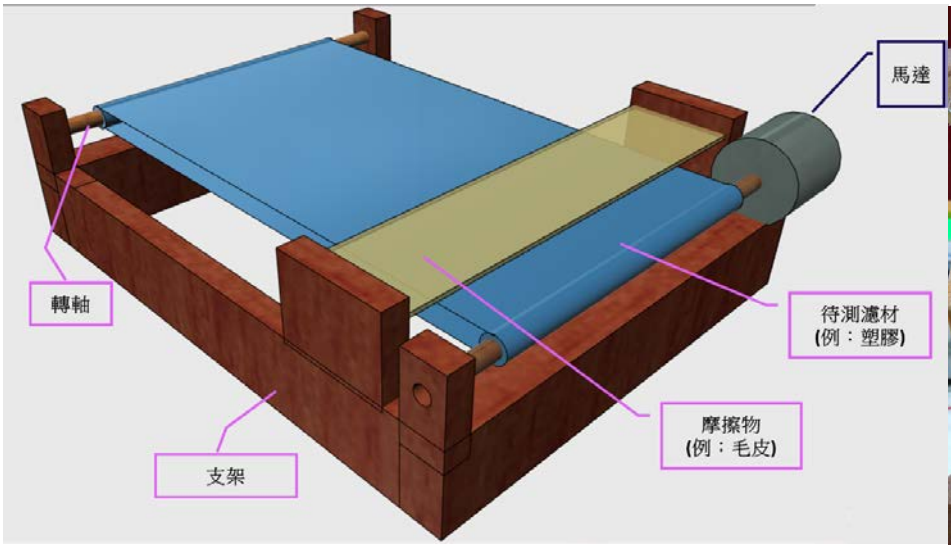
(二)步驟：1.製作支架並將馬達組入裝置上方使其轉動

2.在馬達軸上裝入待測濾材，並以一塊包覆摩擦物的塑膠塊放置其上

3.啟動馬達使其產生靜電並測試其流暢性

4.使用驗電瓶確認裝置是否產生靜電

5.利用此裝置運轉後，測其 PM2.5 值



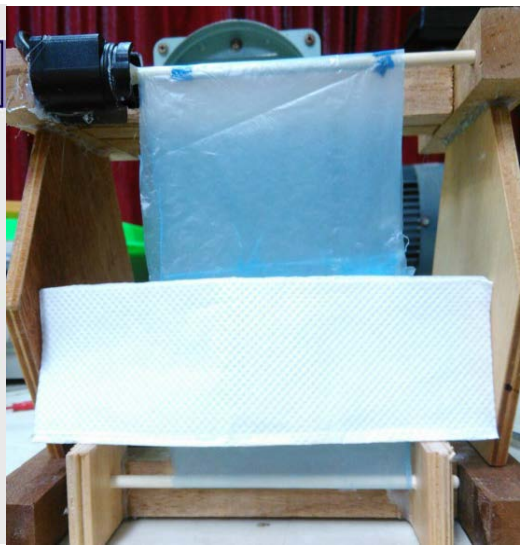


示意圖　　實際圖

(三)結果：1. 馬達在啟動後順利運作

2. 使用驗電瓶檢測其表面帶電量，發現確實帶有大量靜電

(四)討論：1. 在轉動開始時會受到塑膠布接合處影響而暫時性轉動困難，透過調整轉軸可以解決此問題

2.必須使用膠布增加轉軸摩擦係數，以增加塑膠布轉動之摩擦力以避免空轉。

3.此方法效率不彰，所以我開始討論哪一種濾材比較好。

探討 2、找出最佳靜電濾材

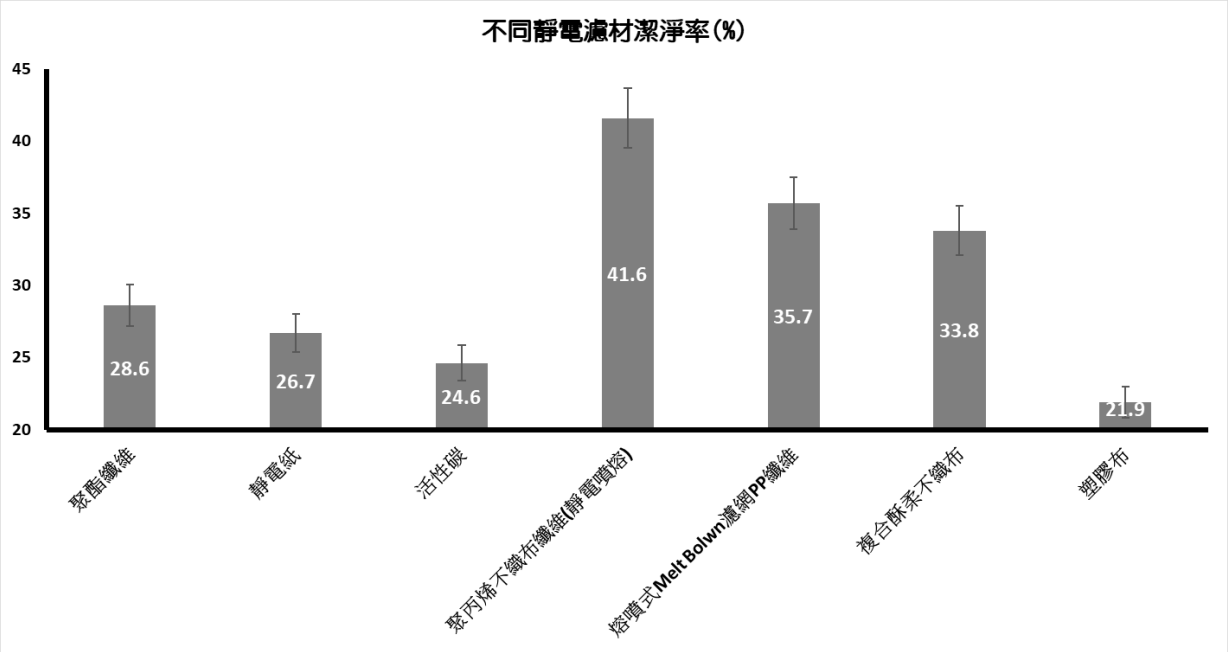
(一)目的：討論不同靜電濾材對吸附效率的影響

(二)步驟：1.分別更換待測濾材的材質(例如：聚酯纖維、靜電紙、活性碳、聚丙烯不織布纖維、熔噴式Melt Bolwn濾網PP纖維、複合酥柔不織布、塑膠布)

2.啟動裝置並測量其吸附率。

3.分析並找出最佳材質

(三)結果：1.比較圖如下



(四)討論：1.整體而言以靜電紙為最佳靜電濾材，可達 24%濾淨效果，但仍不敷使用，故我著手思考其他可行的替代方案。

研究四、使用高壓靜電金屬網吸附懸浮微粒

探討 1、直流高壓金屬網的架設

(一)目的：使用高壓金屬網作為帶電濾材

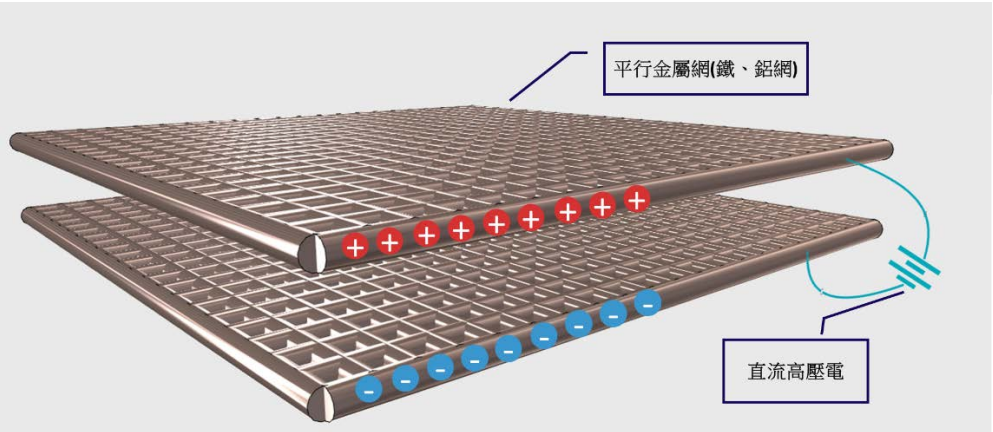
(二)步驟：1. 將兩金屬板相隔短距離放置(如下圖)

2.透過高壓直流電源供應器對金屬板充能，累積靜電荷

3.將金屬網置於量測箱入口

4.開啟風洞，使汙染源流過金屬板

5.紀錄並量測數據。



(三)結果：1.測得高壓靜電金屬板確實對於懸浮微粒有吸附效果

(四)討論：1.充能時間必須夠常以累積足夠量之電荷

2.以 12 分鐘充電時較果為佳，故此後實驗均以充電 12 分鐘為基準

探討 2、輸入電壓對吸附效率的影響

(一)目的：改變電壓，以找出最佳吸附效果時之電壓值

(二)步驟：1. 改變輸入電時之電壓

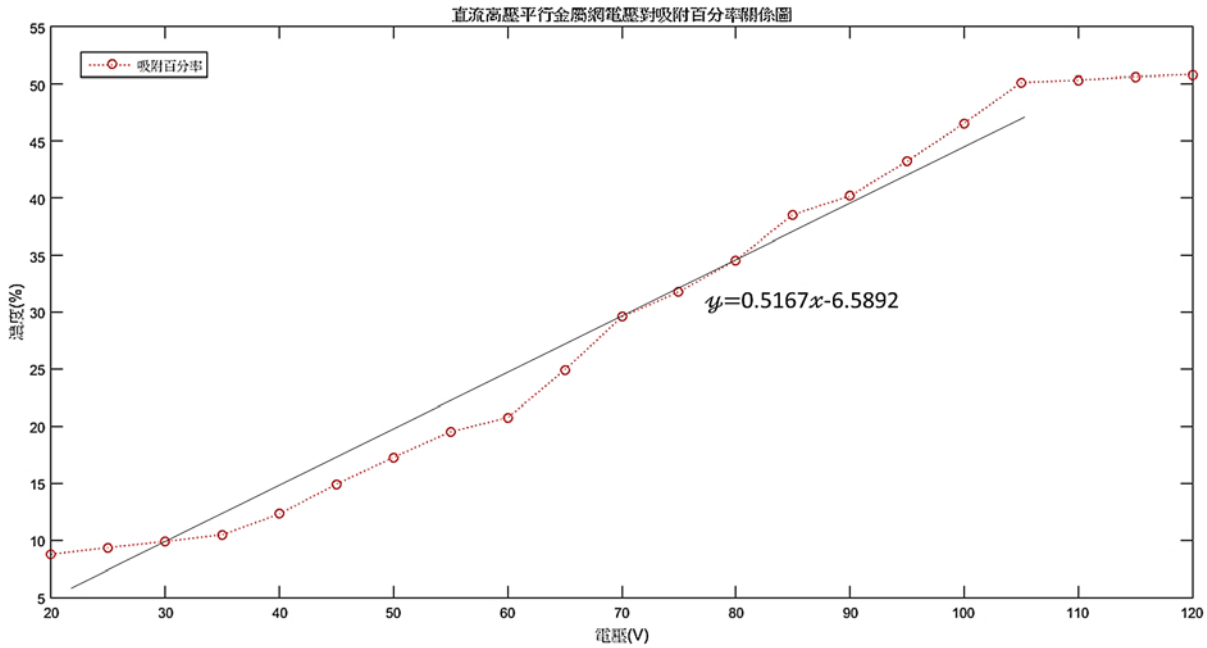
2.使氣流通過金屬網

3.量測通過淨化裝置後之 PM2.5 濃度

4.換算成吸附百分比並作圖

(三)結果：1.吸附效率隨電壓之提升而增加，顯示表面電荷也隨之增加(如下圖)

2.電壓到達某一程度則吸附效率趨近於穩定



(四)討論：1.電壓太高時對於提升淨化效果影響不顯著，原因為表面電荷以達最大密度，故無法提升效能，因此有一最佳化之電壓值(約 104.2V)

探討 3、網距對吸附效率的影響

(一)目的：改變網距，測試庫倫力衰減程度對 PM2.5 之吸附的影響

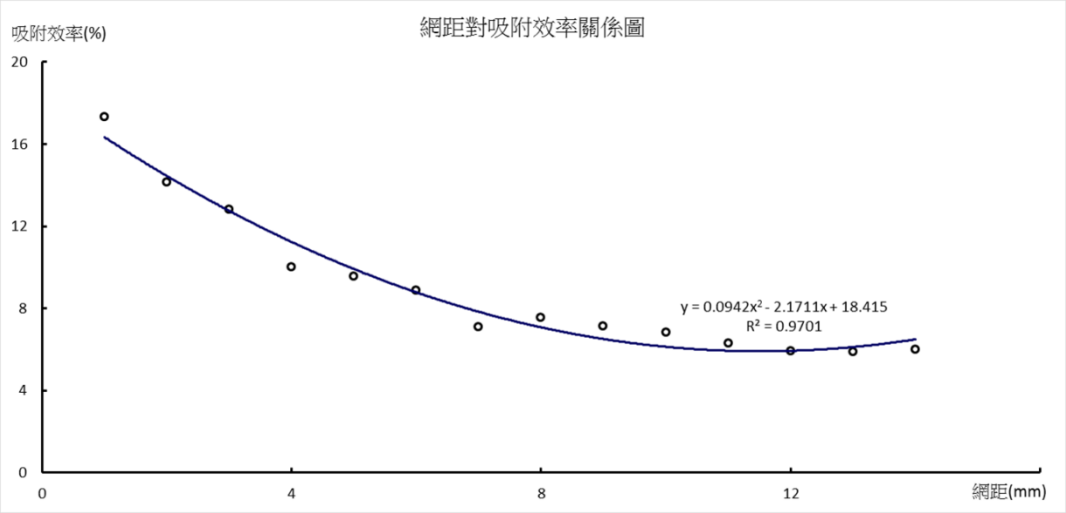
(二)步驟：1. 將兩帶電金屬板以不同距離間隔放置

2、使空氣通過金屬板夾層

3.量測通過淨化裝置後之 PM2.5 濃度

4.換算成吸附百分比並作圖

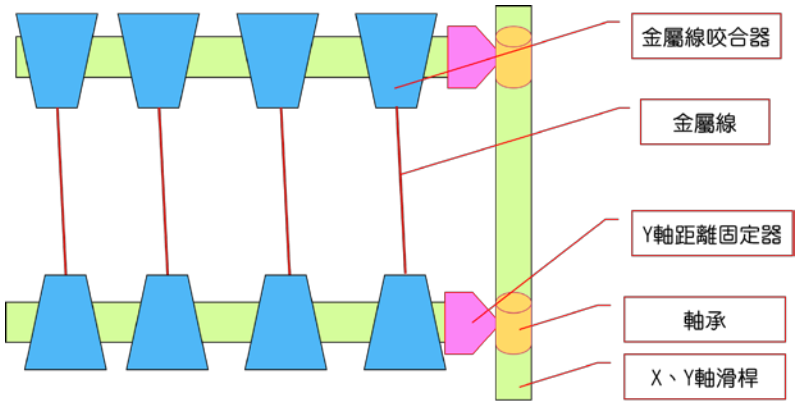
(三)結果：1.網距越近則吸附效能越佳(如下圖)



(四)討論：1.吸附效能與距離平方成反比，與文獻吻合。

2.由於電子能量的耗損，使得吸附效能和距離平方成反比。

3.由於金屬網不夠緊密，我著手設計了一個可自行編織網子的機械結構(如下圖)，期望能有更好的帶電效果。

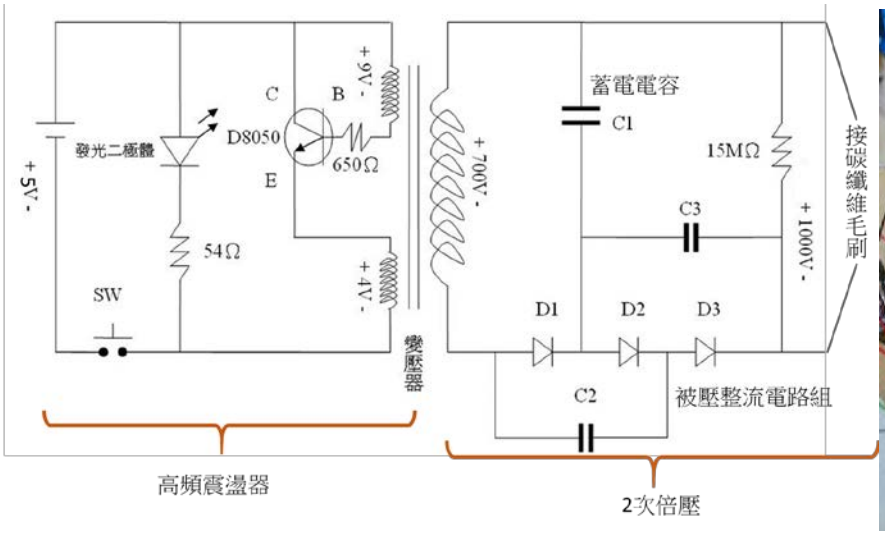


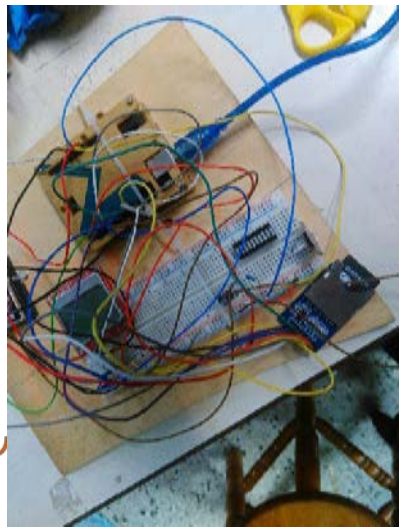
研究五、負離子吸附系統的設計與操作

探討 1、負離子產生器的設計與製作

(一)目的：自製負離子產生器，吸附 PM2.5 粒子

(二)步驟：1. 將電路以下圖方式連接





2.在電路圖放電端(紅色處)安裝碳化纖維刷毛

(三)結果：1.此電路可將直流電加壓以增加尖端放電現象

2.此裝置可產生強大的靜電場，使驗電瓶有極大的反應

(四)討論：1.我可以利用這個裝置進行吸附 PM2.5 的實驗

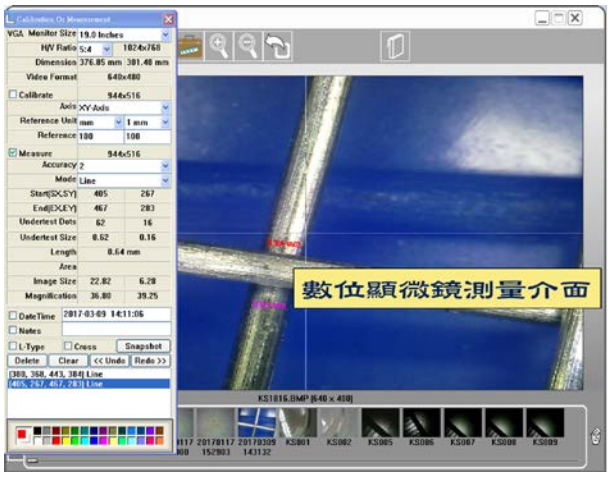
2. 原理：將輸入的直流通過脈衝式電路、過壓限流線路升為交流高壓，濾波得到直流負高壓，將直流負高壓連接到金屬或碳元素製作的釋放尖端，利用尖端直流高壓產生高電暈，高速地放出大量的電子，而電子無法長久存在於空氣中，立刻會被空氣中的氧分子捕捉，從而生成空氣負離子。

探討 2、碳刷總數量對吸附效率的影響

(一)目的：改變碳刷總表面積，了解碳刷總表面積對吸附效能之影響

(二)步驟：1、使用數位顯微鏡移除部分碳刷以改變碳刷總表面積(如下圖數位顯微照片)

規格：【碳刷刷毛：0.3mm】



2、開啟淨化裝置以量測數據

3.測得 PM2.5 濃度並轉換成吸附效率

(三)討論：影響碳纖維毛刷的要素有以下幾點：

1.長度　2.密度　3.粗細　4.數量

我將針對以上四點進行更深入的探討。

探討 3、碳刷粗細對吸附效率的影響

(一)目的：改變碳刷總表面積，了解碳刷總表面積對吸附效能之影響

(二)步驟：1、1、使用數位顯微鏡移除部分碳刷以改變碳刷總表面積(如下圖數位顯微照片)

規格：碳刷刷毛：0.3mm

2、開啟淨化裝置以量測數據

3.測得 PM2.5 濃度並轉換成吸附效率

(三)討論：影響碳纖維毛刷的要素有以下幾點：

1.長度　2.密度　3.粗細　4.數量

我將針對以上四點進行更深入的探討。

探討 4、碳刷粗細對吸附效率的影響

(一)目的：改變碳刷總表面積，了解碳刷總表面積對吸附效能之影響

(二)步驟：1、1、使用數位顯微鏡移除部分碳刷以改變碳刷總表面積(如下圖數位顯微照片)

規格：碳刷刷毛：0.3mm

2、開啟淨化裝置以量測數據

3.測得 PM2.5 濃度並轉換成吸附效率

(三)討論：影響碳纖維毛刷的要素有以下幾點：

1.長度　2.密度　3.粗細　4.數量

我將針對以上四點進行更深入的探討。

探討 5、碳刷粗細對吸附效率的影響

(一)目的：改變碳刷總表面積，了解碳刷總表面積對吸附效能之影響

(二)步驟：1、1、使用數位顯微鏡移除部分碳刷以改變碳刷總表面積(如下圖數位顯微照片)

規格：碳刷刷毛：0.3mm

2、開啟淨化裝置以量測數據

3.測得 PM2.5 濃度並轉換成吸附效率

(三)討論：影響碳纖維毛刷的要素有以下幾點：

1.長度　2.密度　3.粗細　4.數量

我將針對以上四點進行更深入的探討。

六、節能與智慧型判斷系統的製作

探討 1、負離子系統的吸附效果

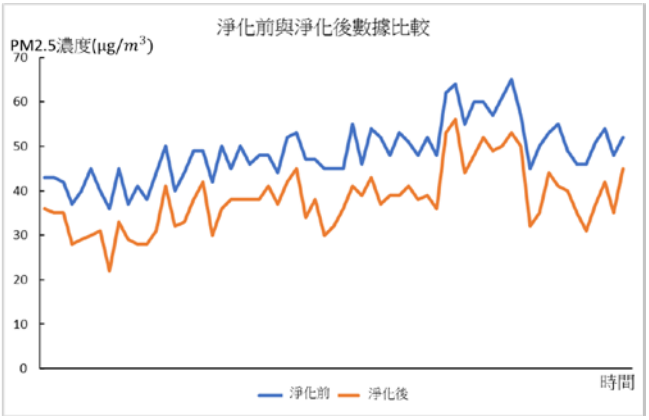
(一)目的：直接利用自製負離子系統淨化空氣，了解自製淨化器在 AQI 指標 101~200 的情況下之沉降效果。

(二)步驟：1.量測兩天內空氣懸浮微粒濃度

2.同時透過淨化裝置淨化空氣

3.比對淨化前後 PM2.5 濃度之差異

(三)結果：1.兩天內之 PM2.5 濃度如下圖：



2.計算後得知平均淨化率約為 52.6%

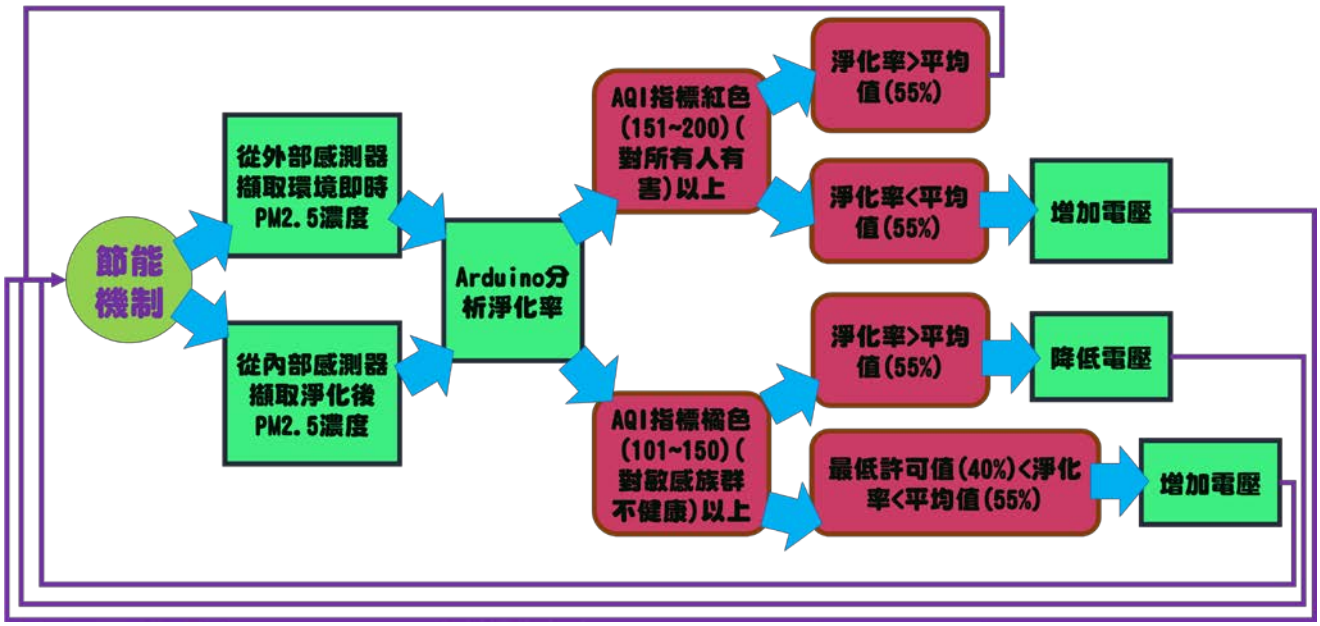
(四)討論：1. 我經過長時間觀察並實驗後發現這種淨化方式非常有效，可以降低一半的空氣懸浮微粒，且在污染嚴重時有更好的吸附效果。

探討 2、本系統之節能設計

(一)目的：使用程式控制負離子產生電路輸出電功率以達節能之效

(二)步驟：1. 使用以下邏輯程式控制電壓的輸出

更詳細的說，如果懸浮微粒濃度高，且淨化率不足，我就給予較高的電壓，並再次測試，若如果懸浮微粒濃度高，且淨化率過高，我就給予較低的電壓，並再次測試，但以淨化率不低於 40%為限。(如下圖)



2. 設定 Arduino 單晶片微控制板上當背景濃度<45 μ g/m^{3}時降低一半電壓

3. 監測節能前與節能後的吸附效率

(三)結果：1.監測 2 天後分析濃度

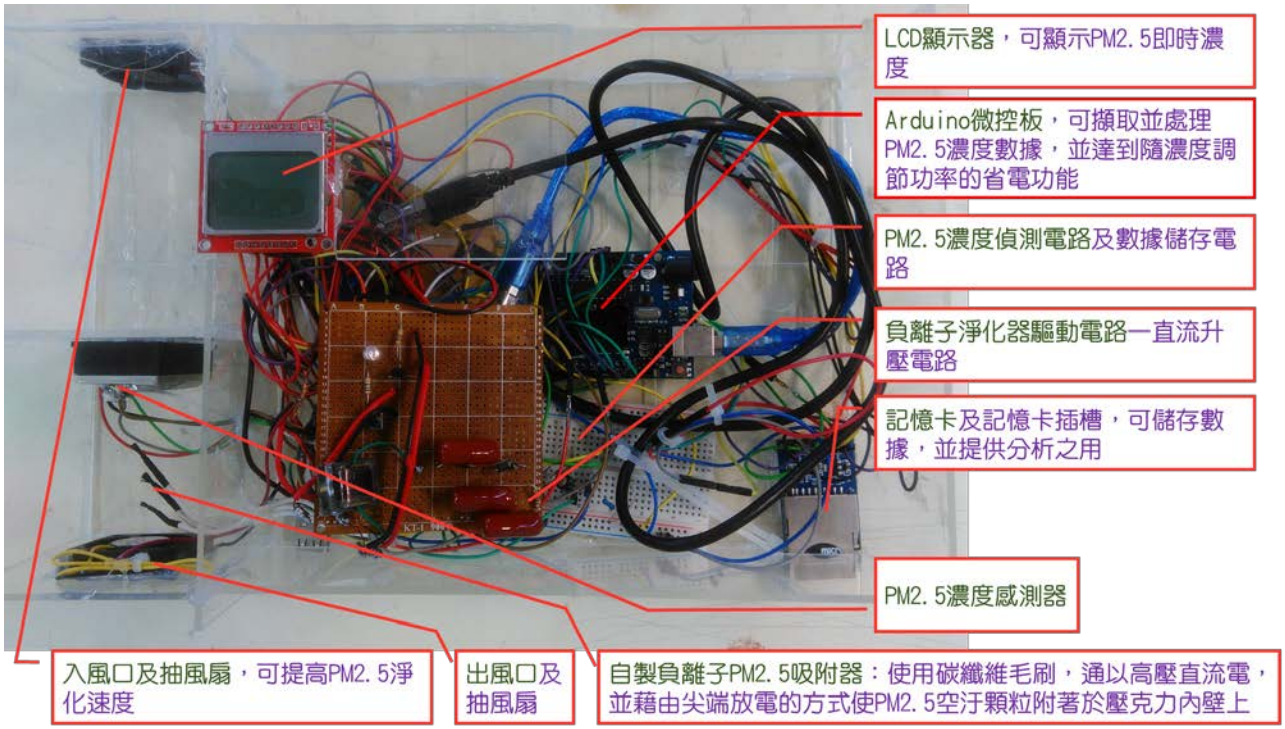
懸浮微粒背景濃度($\mu g/m^3$)	電壓	吸附率(%)
40	高電壓	51.7
	低電壓	51.3
60	高電壓	63.6
	低電壓	54.8

(四)討論：1.我發現監控效果仍維持在 50%左右，不受電壓影響

2.在濃度<45 $\mu g/m^3$時，降低至一半的電壓仍可維持一定的吸附效果

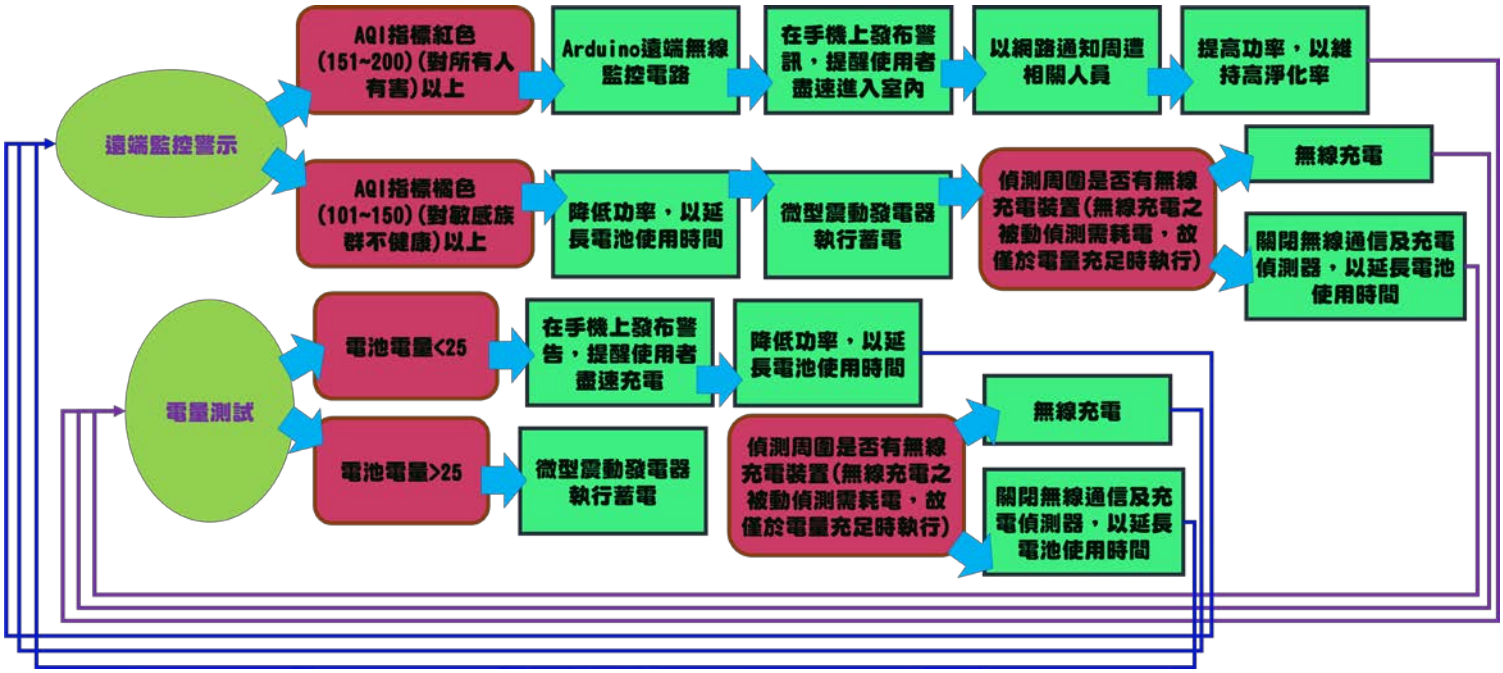
探討三、實際應用

1.半成品展示圖



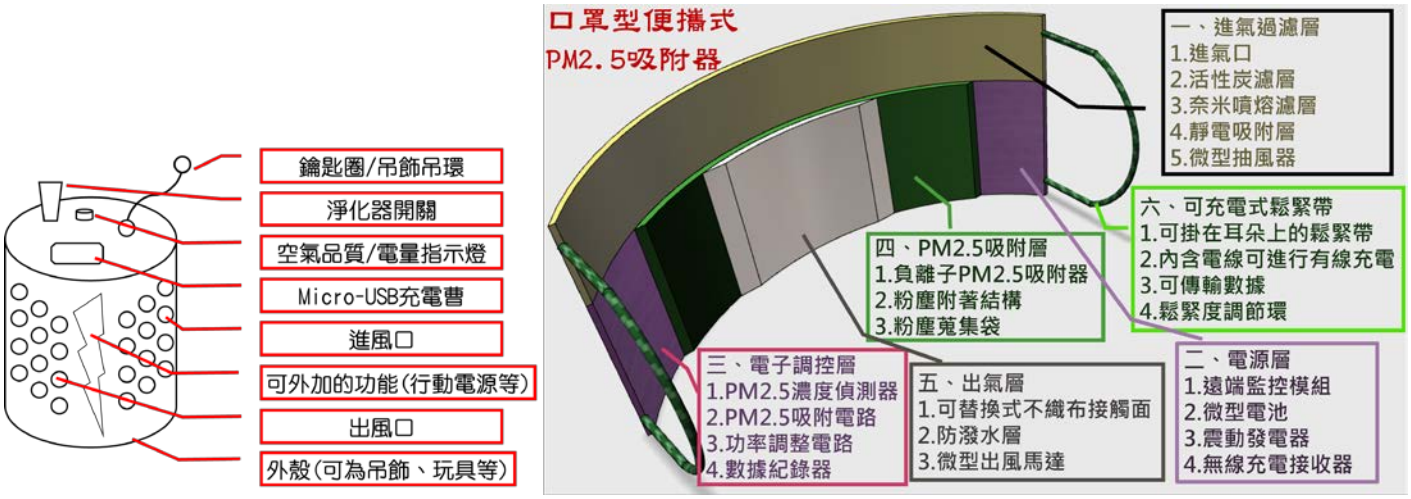
2.電路邏輯實作

下一個實驗我將會把外部的電路完成，使其更完備(如下圖)



3.應用設計圖

在本研究後期將會把成品進行實用化的製作，並落實低成本、高實用、高創新的理念。(如下圖)



研究七、現行空氣清淨機效能比較與長時可靠性分析

探討1、本系統與現行空氣清淨機效能比較

(一)目的：使用不同市售空氣清淨機 進行空氣清淨效率的比較

(二)步驟：1. 我們使用了以下的市售空氣清淨機與自製的空氣清淨系統進行效率與速度的比較：3M 個人隨身型空氣清淨機、Hanlin 車用家用迷你空氣清淨機、小米二代智能空氣清淨機、SHARP 夏普日本原裝空氣清淨機 hypersonic HP2306 臭氧負離子空氣清淨機、日本 IONION LX 小型空氣清淨機

2.同時透過淨化裝置淨化空氣

3.比對淨化前後 PM2.5 濃度之差異

4. 我們首先將各種空氣清淨機放在一個密閉環境中，並通少許的懸浮微粒並且同

時開機測試，測量其濾淨前後濃度

5.分測量其將環境中的 PM 2.5 濃度從一個很高的定值降到一個較低的標準值所需花的時間

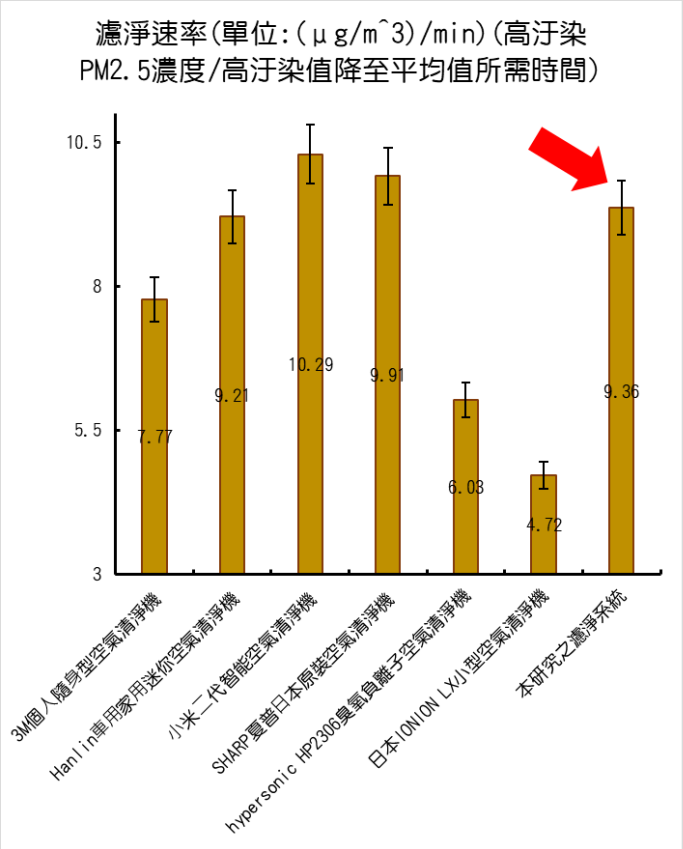
6.計算瞬間過濾效果(%)；算法:【((濾淨前 PM2.5 濃度-濾淨後 PM2.5 濃度)/濾淨前 PM2.5 濃度)*%】

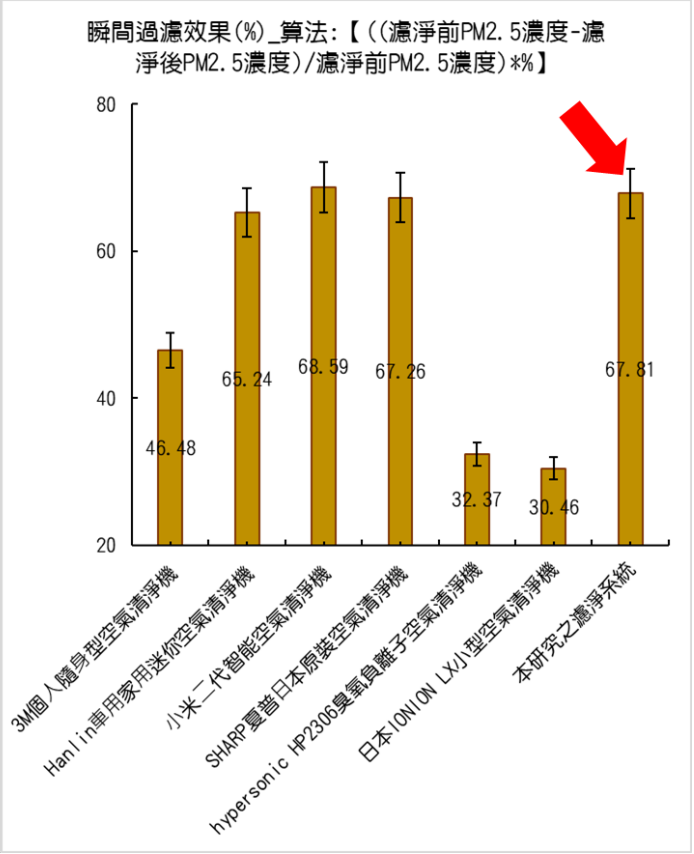
7. 計算濾淨速率(單位:(μ g/m^3)/min)(高汙染 PM2.5 濃度/高汙染)

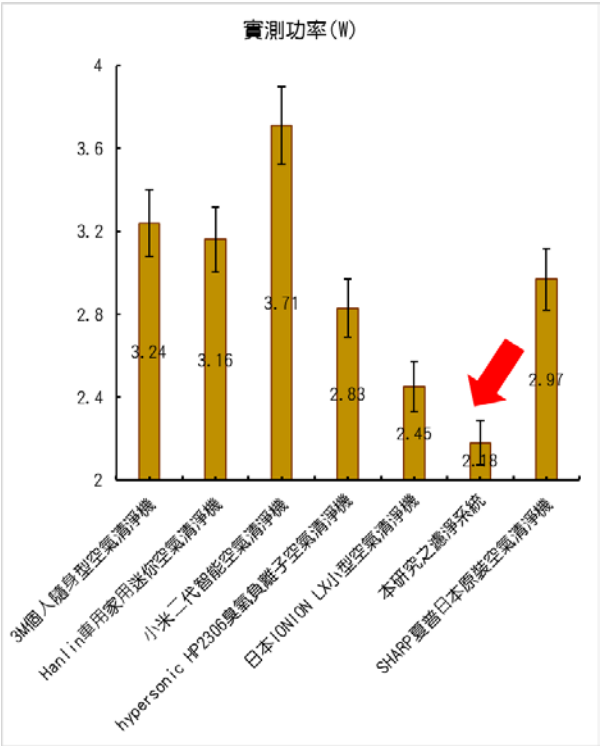
(三)結果：1. 我們首先分析各個品牌所使用的技術，其中最主要的技術有噴融濾網負離子吸附的技術

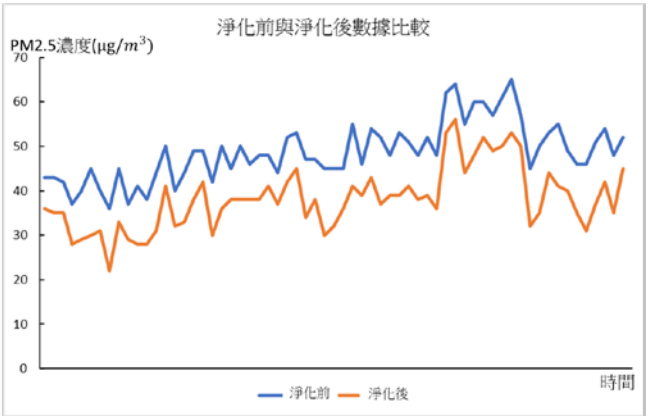
品牌	空氣濾淨模式
3M 個人隨身型空氣清淨機	濾網
Hanlin車用家用迷你空氣清淨機	濾網+負離子
小米二代智能空氣清淨機	濾網+負離子
SHARP 夏普日本原裝空氣清淨機	濾網+負離子
hypersonic HP2306 臭氧負離子空氣清淨機	負離子
日本 IONION LX 小型空氣清淨機	負離子
本研究之濾淨系統	濾網+負離子

2. 接著我們將各個品牌的濾淨效率與濾淨速度繪製成長條圖(如下圖)，在這裡我們可以很明顯的看到，自製的濾淨系統相較於市售的濾淨系統之功率與速率維持在一個很高的品質，並且我們可以發現好使用濾網加上負離子濾淨功能的濾淨系統會有較好品質。


濾淨速率(單位:(μg/m^3)/min)(高汙染PM2.5濃度/高汙染值降至平均值所需時間)
10.5
8
5.5
3
7.77
9.21
10.29
9.91
6.08
4.72
9.36
3M個人隨身型空氣清淨機
Hanlin專用家用迷你空氣清淨機
小米二代智能空氣清淨機
SHARP夏普日本原裝空氣清淨機
hypersonic HP2306臭氧負離子空氣清淨機
日本IONION LX小型空氣清淨機
本研究之濾淨系統


瞬間過濾效果(%)_算法:【((濾淨前PM2.5濃度-濾淨後PM2.5濃度)/濾淨前PM2.5濃度)*%】
80
60
40
20
46.48
65.24
68.59
67.26
32.37
30.46
67.81
3M個人隨身型空氣清淨機
Hanlin專用家用迷你空氣清淨機
小米二代智能空氣清淨機
SHARP夏普日本原裝空氣清淨機
hypersonic HP2306臭氧負離子空氣清淨機
日本IONION LX小型空氣清淨機
本研究之濾淨系統


實測功率(W)
4
3.6
3.2
2.8
2.4
2
3.24
3.16
3.71
2.83
2.45
2.18
2.97
3M個人隨身型空氣清淨機
Hanlin專用家用迷你空氣清淨機
小米二代智能空氣清淨機
hypersonic HP2306臭氧負離子空氣清淨機
日本IONION LX小型空氣清淨機
本研究之濾淨系統
SHARP夏普日本原裝空氣清淨機



2.計算後得知平均淨化率約為 52.6%

(四)討論：1. 我經過長時間觀察並實驗後發現這種淨化方式非常有效，可以降低一半的空氣懸浮微粒，且在污染嚴重時有更好的吸附效果。

四、未來展望

本次的研究成功地利用 Arduino 單晶片微控制板進行 PM2.5 即時監測，此一攜帶式監測系統未來可放置在室內，隨時得知 PM2.5 濃度，並提醒開啟空氣淨化裝置，或帶上負離子淨化口罩。同時也成功開發直流高壓負離子裝置進行懸浮粒子的吸附與淨化，不僅可免去體積龐大的空氣濾淨器的麻煩，也藉由數位控制進行裝置省電，未來期望能應用在攜帶式的空氣濾淨口罩上，可更加便利的去除 PM2.5 懸浮微粒，使我可以對自己呼吸的空氣有所監控。

伍、參考資料

一、翰林出版社。高中物理第一冊。電與磁。

二、翰林出版社。自然與生活科技第五冊。電。

三、環保署＼空氣品質監測網＼細懸浮微粒指標:http://taqm.epa.gov.tw/taqm/tw/Pm25Index.aspx

四、PM2.5 全台即時概況 http：//env.childgrowth.info/air/

五、高壓電 https：//zh.wikipedia.org/wiki/%E9%AB%98%E5%A3%93%E9%9B%BB

六、懸浮粒子 https：//zh.wikipedia.org/wiki/%E6%87%B8%E6%B5%AE%E7%B2%92%E5%AD%90

七、碳纖維 https：//zh.wikipedia.org/wiki/%E7%A2%B3%E7%BA%96%E7%B6%AD

八、楊明豐(2014)。**Arduino 最加入門與應用—打造互動設計輕鬆學**。台北：碁峰。

九、賴柏廷(2016)。環境及淨化設備於細懸浮微粒子影響之研究。國立彰化師範大學機電工程學系所碩士論文。**彰化師範大學博碩士論文系統**，G00A0351002。

十、廖述良等. (1991). **一般室內空氣污染物與污染源現況之調查研究**, EPA-80-F101-09-24, 行政院環保署。

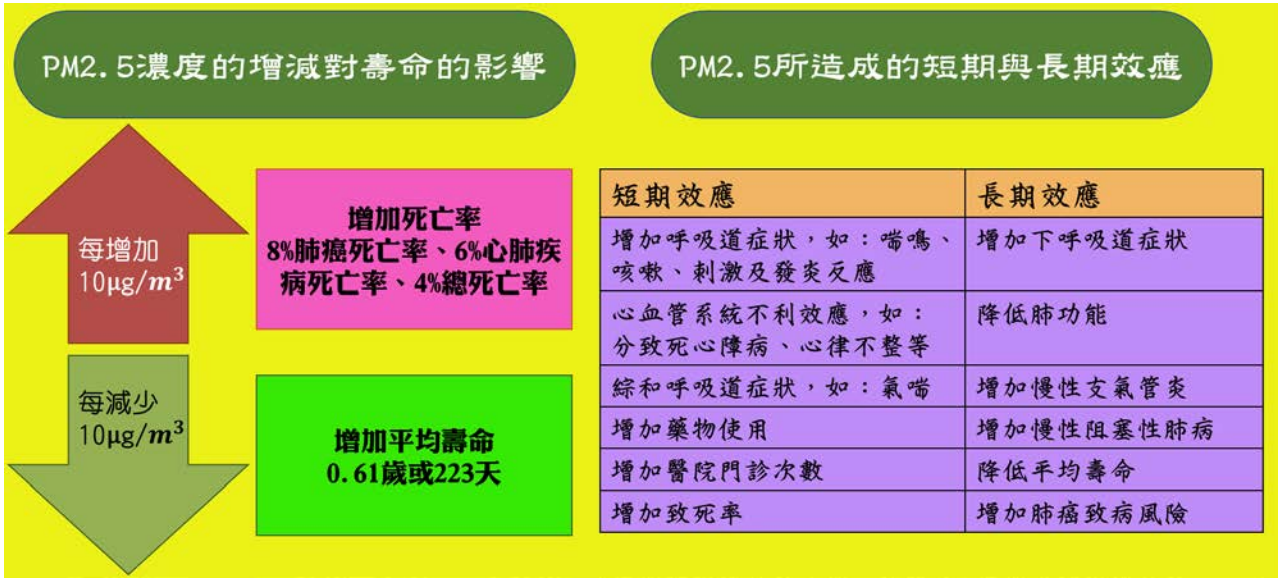
十一、王秋森. (1993). **氣膠技術學**, 國立台灣大學醫學院出版委員會出版, 台北。

十二、張順雄、張忠誠、李榮乾(譯)(1999)。電子元件與電路理論。台北市：東華 書局。

陸、附錄

A、PM2.5懸浮微粒對人體的影響

一、PM2.5 微粒造成的長短期影響：空氣中存在許多污染物，其中漂浮在空氣中類似灰塵的粒狀物稱為懸浮微粒(particulate matter, PM)，PM 粒徑大小有別，小於或等於 2.5 微米(μm)的粒子，就稱為 PM2.5，通稱細懸浮微粒，單位以微克/立方公尺(μg/m3)表示之，它的直徑還不到人的頭髮絲粗細的 1/28，非常微細可穿透肺部氣泡，並直接進入血管中隨著血液循環全身，故對人體及生態所造成之影響是不容忽視的。如下表所示（下表/圖為自行整理）。



短期效應	長期效應
增加呼吸道症狀，如：喘鳴、咳嗽、刺激及發炎反應	增加下呼吸道症狀
心血管系統不利效應，如：分致死心障病、心律不整等	降低肺功能
綜和呼吸道症狀，如：氣喘	增加慢性支氣管炎
增加藥物使用	增加慢性阻塞性肺病
增加醫院門診次數	降低平均壽命
增加致死率	增加肺癌致病風險

二、空汙程度分級指標：空氣汙染物依照粒徑大小分為幾種不同的層級，並將其製作成空氣品質指標(AQI)。空氣品質指標為依據監測資料將當日空氣中臭氧(O3)、細懸浮微粒(PM2.5)、懸浮微粒(PM10)、一氧化碳(CO)、二氧化硫(SO2)及二氧化氮(NO2)濃度等數值，以其對人體健康的影響程度，分別換算出不同污染物之副指標值，再以當日各副指標之最大值為該測站當日之空氣品質指標值(AQI)。在此研究中，我只討論 PM2.5 所對照到的 AQI，如下頁表所示。

AQI指標	普通51~100	對敏感族群不健康 101~150	對所有族群不健康 151~200	非常不健康201~300
PM2.5 （μg/m3 ） 24小時平均值	15.5 - 35.4	35.5 - 54.4	54.5 - 150.4	150.5 - 250.4
狀態色塊	黃	橘	紅	紫
人體健康影響	空氣品質普通；但對非常少數之極敏感族群產生輕微影響。	空氣污染物可能會對敏感族群的健康造成影響，但是對一般大眾的影響不明顯。	對所有人的健康開始產生影響，對於敏感族群可能產生較嚴重的健康影響。	健康警報：所有人都可能產生較嚴重的健康影響。

B、程式碼

(一)Arduino 單晶片微控制板程式碼-PM2.5 偵測器數據擷取部分

```
#include <Wire.h>  //I2C 通訊標頭檔
```

```
const int SLAVE_ADDRESS = 1;
char incomingByte = 0;
char incomingByte2 = 0;
int i1=0;
float i2=0;
int i3=0;
const int measurePin = 0;
const int ledPower = 2;
const int blankping1=1;
const int blankping2=2;
const int blankping3=3;
int blanksignal=0;
int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
unsigned long timeshow;
unsigned long minshow;
unsigned long secshow;
char dustchar[7]={'0','0','0','0','0','0','\0'};
char dustchardot[7]={'0','0','0','0','0','0','\0'};
char Dust_Density[14]={"Dust Density："};
int dustint =0;
#define PIN_SCE   7 //Pin 3 on LCD
#define PIN_RESET 6 //Pin 4 on LCD
#define PIN_DC    5 //Pin 5 on LCD
#define PIN_SDIN  8 //Pin 8 on LCD
#define PIN_SCLK  3 //Pin 7 on LCD
#define LCD_COMMAND 0
#define LCD_DATA  1
#define LCD_X     84
#define LCD_Y     48

static const byte ASCII[][5] = {
  {0x00, 0x00, 0x00, 0x00, 0x00} // 20
  ,{0x00, 0x00, 0x5f, 0x00, 0x00} // 21 !
  ,{0x00, 0x07, 0x00, 0x07, 0x00} // 22 "
```

設定 LCD 輸出字元

```
,{0x14, 0x7f, 0x14, 0x7f, 0x14} // 23 #
,{0x24, 0x2a, 0x7f, 0x2a, 0x12} // 24 $
,{0x23, 0x13, 0x08, 0x64, 0x62} // 25 %
,{0x36, 0x49, 0x55, 0x22, 0x50} // 26 &
,{0x00, 0x05, 0x03, 0x00, 0x00} // 27 '
,{0x00, 0x1c, 0x22, 0x41, 0x00} // 28 (
,{0x00, 0x41, 0x22, 0x1c, 0x00} // 29 )
,{0x14, 0x08, 0x3e, 0x08, 0x14} // 2a *
,{0x08, 0x08, 0x3e, 0x08, 0x08} // 2b +
,{0x00, 0x50, 0x30, 0x00, 0x00} // 2c ,
,{0x08, 0x08, 0x08, 0x08, 0x08} // 2d -
,{0x00, 0x60, 0x60, 0x00, 0x00} // 2e .
,{0x20, 0x10, 0x08, 0x04, 0x02} // 2f /
,{0x3e, 0x51, 0x49, 0x45, 0x3e} // 30 0
,{0x00, 0x42, 0x7f, 0x40, 0x00} // 31 1
,{0x42, 0x61, 0x51, 0x49, 0x46} // 32 2
,{0x21, 0x41, 0x45, 0x4b, 0x31} // 33 3
,{0x18, 0x14, 0x12, 0x7f, 0x10} // 34 4
,{0x27, 0x45, 0x45, 0x45, 0x39} // 35 5
,{0x3c, 0x4a, 0x49, 0x49, 0x30} // 36 6
,{0x01, 0x71, 0x09, 0x05, 0x03} // 37 7
,{0x36, 0x49, 0x49, 0x49, 0x36} // 38 8
,{0x06, 0x49, 0x49, 0x29, 0x1e} // 39 9
,{0x00, 0x36, 0x36, 0x00, 0x00} // 3a :
,{0x00, 0x56, 0x36, 0x00, 0x00} // 3b ;
,{0x08, 0x14, 0x22, 0x41, 0x00} // 3c <
,{0x14, 0x14, 0x14, 0x14, 0x14} // 3d =
,{0x00, 0x41, 0x22, 0x14, 0x08} // 3e >
,{0x02, 0x01, 0x51, 0x09, 0x06} // 3f ?
,{0x32, 0x49, 0x79, 0x41, 0x3e} // 40 @
,{0x7e, 0x11, 0x11, 0x11, 0x7e} // 41 A
,{0x7f, 0x49, 0x49, 0x49, 0x36} // 42 B
,{0x3e, 0x41, 0x41, 0x41, 0x22} // 43 C
,{0x7f, 0x41, 0x41, 0x22, 0x1c} // 44 D
,{0x7f, 0x49, 0x49, 0x49, 0x41} // 45 E
,{0x7f, 0x09, 0x09, 0x09, 0x01} // 46 F
,{0x3e, 0x41, 0x49, 0x49, 0x7a} // 47 G
,{0x7f, 0x08, 0x08, 0x08, 0x7f} // 48 H
,{0x00, 0x41, 0x7f, 0x41, 0x00} // 49 I
```

```
,{0x20, 0x40, 0x41, 0x3f, 0x01} // 4a J
,{0x7f, 0x08, 0x14, 0x22, 0x41} // 4b K
,{0x7f, 0x40, 0x40, 0x40, 0x40} // 4c L
,{0x7f, 0x02, 0x0c, 0x02, 0x7f} // 4d M
,{0x7f, 0x04, 0x08, 0x10, 0x7f} // 4e N
,{0x3e, 0x41, 0x41, 0x41, 0x3e} // 4f O
,{0x7f, 0x09, 0x09, 0x09, 0x06} // 50 P
,{0x3e, 0x41, 0x51, 0x21, 0x5e} // 51 Q
,{0x7f, 0x09, 0x19, 0x29, 0x46} // 52 R
,{0x46, 0x49, 0x49, 0x49, 0x31} // 53 S
,{0x01, 0x01, 0x7f, 0x01, 0x01} // 54 T
,{0x3f, 0x40, 0x40, 0x40, 0x3f} // 55 U
,{0x1f, 0x20, 0x40, 0x20, 0x1f} // 56 V
,{0x3f, 0x40, 0x38, 0x40, 0x3f} // 57 W
,{0x63, 0x14, 0x08, 0x14, 0x63} // 58 X
,{0x07, 0x08, 0x70, 0x08, 0x07} // 59 Y
,{0x61, 0x51, 0x49, 0x45, 0x43} // 5a Z
,{0x00, 0x7f, 0x41, 0x41, 0x00} // 5b [
,{0x02, 0x04, 0x08, 0x10, 0x20} // 5c ^
,{0x00, 0x41, 0x41, 0x7f, 0x00} // 5d _
,{0x04, 0x02, 0x01, 0x02, 0x04} // 5e ^
,{0x40, 0x40, 0x40, 0x40, 0x40} // 5f _
,{0x00, 0x01, 0x02, 0x04, 0x00} //a
,{0x20, 0x54, 0x54, 0x54, 0x78} // 61 b
,{0x7f, 0x48, 0x44, 0x44, 0x38} // 62 b
,{0x38, 0x44, 0x44, 0x44, 0x20} // 63 c
,{0x38, 0x44, 0x44, 0x48, 0x7f} // 64 d
,{0x38, 0x54, 0x54, 0x54, 0x18} // 65 e
,{0x08, 0x7e, 0x09, 0x01, 0x02} // 66 f
,{0x0c, 0x52, 0x52, 0x52, 0x3e} // 67 g
,{0x7f, 0x08, 0x04, 0x04, 0x78} // 68 h
,{0x00, 0x44, 0x7d, 0x40, 0x00} // 69 i
,{0x20, 0x40, 0x44, 0x3d, 0x00} // 6a j
,{0x7f, 0x10, 0x28, 0x44, 0x00} // 6b k
,{0x00, 0x41, 0x7f, 0x40, 0x00} // 6c l
,{0x7c, 0x04, 0x18, 0x04, 0x78} // 6d m
,{0x7c, 0x08, 0x04, 0x04, 0x78} // 6e n
,{0x38, 0x44, 0x44, 0x44, 0x38} // 6f o
,{0x7c, 0x14, 0x14, 0x14, 0x08} // 70 p
```

```
  ,{0x08, 0x14, 0x14, 0x18, 0x7c} // 71 q
  ,{0x7c, 0x08, 0x04, 0x04, 0x08} // 72 r
  ,{0x48, 0x54, 0x54, 0x54, 0x20} // 73 s
  ,{0x04, 0x3f, 0x44, 0x40, 0x20} // 74 t
  ,{0x3c, 0x40, 0x40, 0x20, 0x7c} // 75 u
  ,{0x1c, 0x20, 0x40, 0x20, 0x1c} // 76 v
  ,{0x3c, 0x40, 0x30, 0x40, 0x3c} // 77 w
  ,{0x44, 0x28, 0x10, 0x28, 0x44} // 78 x
  ,{0x0c, 0x50, 0x50, 0x50, 0x3c} // 79 y
  ,{0x44, 0x64, 0x54, 0x4c, 0x44} // 7a z
  ,{0x00, 0x08, 0x36, 0x41, 0x00} // 7b {
  ,{0x00, 0x00, 0x7f, 0x00, 0x00} // 7c |
  ,{0x00, 0x41, 0x36, 0x08, 0x00} // 7d }
  ,{0x10, 0x08, 0x08, 0x10, 0x08} // 7e ~
  ,{0x78, 0x46, 0x41, 0x46, 0x78} // 7f DEL
};

void setup(void) {
 Wire.begin();
 LCDInit(); //Init the LCD
  Serial.begin(9600);
 pinMode(ledPower,OUTPUT);
}

void loop(void) {
 digitalWrite(ledPower,LOW); // power on the LED
 delayMicroseconds(samplingTime);
 voMeasured =
analogRead(measurePin)-((analogRead(blankping1)+analogRead(blankping2)+analogRe
ad(blankping3))/3); // read the dust value
  delayMicroseconds(deltaTime);
  digitalWrite(ledPower,HIGH); // turn the LED off
  delayMicroseconds(sleepTime);
  calcVoltage = voMeasured * (5.0 / 1024.0);
  dustDensity = 0.17 * calcVoltage - 0.1;
  timeshow=millis();
  minshow=(timeshow/1000)/60;
  secshow=(timeshow/1000)%60;
```

程式起始設定

擷取數據

去雜訊

數據轉換

時間紀錄

```
 dustint=dustDensity * 100000;
sprintf (dustchar, "%d", dustint);
if (dustint<=10000){
dustchardot[0]=dustchar[0];
dustchardot[1]=dustchar[1];
dustchardot[2]='.';
dustchardot[3]=dustchar[2];
dustchardot[4]=dustchar[3];
dustchardot[5]=' ';
}
else {
dustchardot[0]=dustchar[0];
dustchardot[1]=dustchar[1];
dustchardot[2]=dustchar[2];
dustchardot[3]='.';
dustchardot[4]=dustchar[3];
dustchardot[5]=dustchar[4];
}

  LCDClear();
  LCDString(Dust_Density);
      LCDString(dustchardot);
    Serial.println(dustDensity * 1000); // 這裡將數值呈現改成較常用的單位( ug/m3 )
     i1 = (int)(dustDensity * 1000);
 i2=(dustDensity * 1000)-i1;
 i3=i2*100;
incomingByte=(i1);
incomingByte2=(i3);
     Wire.beginTransmission(SLAVE_ADDRESS);
  Wire.write(incomingByte);
      Wire.endTransmission();
       Wire.beginTransmission(SLAVE_ADDRESS);
    Wire.write(incomingByte2);
Wire.endTransmission();
    Serial.println("PASS");
      Serial.println(freeRam());
  delay(1000);
}
```

LCD 輸出

I2C 通訊

LCD 函式

```
void gotoXY(int x, int y) {
  LCDWrite(0, 0x80 | x);
  LCDWrite(0, 0x40 | y);
}


void LCDBitmap(char my_array[]){
  for (int index = 0 ; index < (LCD_X * LCD_Y / 8) ; index++)
    LCDWrite(LCD_DATA, my_array[index]);
}

void LCDCharacter(char character) {
  LCDWrite(LCD_DATA, 0x00);

  for (int index = 0 ; index < 5 ; index++)
    LCDWrite(LCD_DATA, ASCII[character - 0x20][index]);
  LCDWrite(LCD_DATA, 0x00); //Blank vertical line padding
}

void LCDString(char *characters) {
  while (*characters)
    LCDCharacter(*characters++);
}


void LCDClear(void) {
  for (int index = 0 ; index < (LCD_X * LCD_Y / 8) ; index++)
    LCDWrite(LCD_DATA, 0x00);

  gotoXY(0, 0);
}

void LCDInit(void) {

  //Configure control pins
  pinMode(PIN_SCE, OUTPUT);
  pinMode(PIN_RESET, OUTPUT);
  pinMode(PIN_DC, OUTPUT);
  pinMode(PIN_SDIN, OUTPUT);
```

```
  pinMode(PIN_SCLK, OUTPUT);
  digitalWrite(PIN_RESET, LOW);
  digitalWrite(PIN_RESET, HIGH);

  LCDWrite(LCD_COMMAND, 0x21);
  LCDWrite(LCD_COMMAND, 0xB0);
  LCDWrite(LCD_COMMAND, 0x04);
  LCDWrite(LCD_COMMAND, 0x14);
  LCDWrite(LCD_COMMAND, 0x20);
  LCDWrite(LCD_COMMAND, 0x0C);
}

void LCDWrite(byte data_or_command, byte data) {
  digitalWrite(PIN_DC, data_or_command);
  digitalWrite(PIN_SCE, LOW);
  shiftOut(PIN_SDIN, PIN_SCLK, MSBFIRST, data);
  digitalWrite(PIN_SCE, HIGH);
}

int freeRam ()
{
  extern int __heap_start, *__brkval;
  int v;
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}
```

動態記憶體剩餘測試

(二)Arduino 單晶片微控制板程式碼-數據儲存部分

```
#include <Wire.h>
const int SLAVE_ADDRESS = 1;
char incomingByte = 0;
char incomingByte2 = 0;
#include <SD.h>
File datafile;
Sd2Card card;
const int chipSelect = 10;

void setup() {
Wire.begin(SLAVE_ADDRESS);
  Serial.begin(9600);
```

SD 卡讀取與測試

```
if (!SD.begin(chipSelect)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
}

void loop() {
 Wire.onReceive(receiveEvent);
 Wire.onReceive(receiveEvent2);
delay(1000);

}

void receiveEvent(int howMany)
{
  while (Wire.available())
  {
    incomingByte = Wire.read();
  datafile=SD.open("pm25data.txt", FILE_WRITE);
    if (datafile){;
    }else{
Serial.println("open error");
    }

if (datafile){
   int trans=incomingByte;
  datafile.println(trans);
     Serial.print(trans);
}else{
      Serial.println("FILEWRITEERROR");
     }
 datafile.close();
}
}

void receiveEvent2(int howMany)
{
  while (Wire.available())
```

接收端收信觸發點

SD 卡寫入函式

```
  {
    incomingByte = Wire.read();
  datafile=SD.open("pm25data.txt", FILE_WRITE);
    if (datafile){;
    }else{
Serial.println("open error");
    }

if (datafile){
  datafile.println(trans);
    Serial.println(trans);
}else{
     Serial.println("FILEWRITEERROR");
  }
     datafile.close();
}
}
```