



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系

PM2.5 手機 APP 網頁即時 監測與控制

指導教授： 黃光宇 教授
 陳健忠 助理教授
組員名單： 陳冠綸 A58C090
 葉丞恩 A58C043
 馮柏翔 A18C075

中華民國一〇九年十二月

嶺東科技大學

資訊管理系

PM2.5 手機 APP 頁即時監測與控制

中華民國一〇九年十二月



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

PM2.5 手機 APP 網頁即時 監測與控制

指導教授：黃光宇 教授

陳健忠 助理教授

組員名單：陳冠綸 A58C090

葉丞恩 A58C043

馮柏翔 A18C075

指導教授：_____

口試委員：_____

謝 誌

本專題報告得以順利完成，先感謝我們兩位指導教授黃光宇老師與陳健忠老師細心的教導與耐心地幫助我們解決問題，遇到困難時給予我們專業的指導，以及耐心地點評我們的缺點，讓本專題得以完成。

口試報告期間，感謝陳建興老師與黃江富老師的細心審閱，給予我們寶貴的建議與點評，讓我們看見專題的缺陷並加以修正使我們的專題更加完善，感謝兩位老師。

陳冠綸 葉丞恩 馮柏翔謹誌
中華民國109年十二月於嶺東

摘 要

目前國內對於空氣品質的議題越來越盛行，空氣汙染的原因為數眾多，在室外與室內都可能有空氣污染物的產生與沉積，而本專題針對室內空氣污染物做各項數據之研究。

本專題研究目的是針對一般室內空間環境中空氣中污染物含量之評估，透過 Arduino UNO 上之感測裝置蒐集室內中 PM2.5、PM10、CO₂、CO、CH₄ 的濃度並在經過檢測，將蒐集到的資料傳至網頁與手機，同一時間將數據儲存在資料庫中，當空氣品質超標時手機與電腦皆會跳出警告通知，進而讓使用者了解哪一些時段的空氣品質不佳，使用者可以使用手機 app 以及網頁自由查看當前空氣數據並可在 app 介面中選擇是否開啟空氣清淨機。

本專題也於 PM2.5 與 PM10 有做連結空氣清淨機之研究，如室內 PM2.5、PM10 出現異常時，將可透過網頁或手機 app 來進行開啟/關閉空氣清淨機之動作，達到遠端監控的功能。

本系統達到以下功能：

1. 接收 Arduino 模組數據:透過 Arduino UNO 接收大氣中空氣之數據，當數值超過標準時跳出警告訊息。
2. 控制空氣清淨機:透過 Arduino UNO 控制空氣清淨機的運轉。
3. 網頁與手機介面顯示:利用網路連接電腦的伺服器，使用者可以透過電腦網頁與手機 app 即時的知道室內空氣狀況。

關鍵詞：PM2.5、PM10、CO₂、CO、CH₄ Arduino、空氣清淨機、空汙

目錄

謝 誌	III
摘 要	IV
目 錄	V
表目錄	VII
圖目錄	VIII
第壹章緒論	1
1.1 研究背景	1
1.2 研究動機	1
1.3 研究目的	1
第貳章文獻回顧與探討	2
2.1 硬體設備	2
2.1.1 Arduino Uno 控制板	2
2.1.2 Arduino 1路繼電器模組	3
2.1.3 PMS3003 (G3) 粉塵感測器	3
2.1.4 MH-Z16 二氧化碳感測器	4
2.1.5 MQ-7 一氧化碳 CO 氣體感測器	5
2.1.6 MQ-5 天然氣氣體感測器	6
2.1.7 空氣清淨機	6
2.2 軟體	7
2.2.1 Arduino IDE	7
2.2.2 Node.js	7
2.2.3 C3.js	7
2.2.4 MongoDB	7
2.2.5 HTML5	7
2.2.6 JavaScript	7
2.2.7 IFTTT	8
2.3 檢測物	8
2.2.8 PM2.5	8
2.2.9 PM10	8
2.2.10 CO ₂	8
2.2.11 CO	8
2.2.12 CH ₄	8
第參章研究方法	9
3.1 研究架構	9
3.2 「PM2.5手機app網頁即時監測與控制架構圖」	10
3.3 「PM2.5手機app網頁即時監測與控制架構圖」 流程圖	11
3.4 「數據蒐集」子系統	12
3.6 「數據蒐集」子系統流程圖	13
3.7 「視覺化即時監控」子系統	14
3.9 「視覺化即時監控」子系統系統流程圖	15
3.10 「異常訊號監控」子系統	16
3.12 「異常訊號監控」子系統流程圖	17

第肆章部分實體畫面	18
4.1系統實作畫面.....	18
第伍章結論.....	21
附錄	錯誤! 尚未定義書籤。
6.1Arduino程式碼.....	22
6.2 C3.js程式碼	25
6.3判斷異常數據並傳送警示訊息之流程	27
參考文獻	28

表目錄

表2.1 Arduino UNO控制板規格	2
表2.2 Arduino 1路繼電器模組 5V規格	3
表2.3 PMS3003 (G3) 空氣品質感測器規格	3
表2.4 MH-Z16 二氧化碳感測器規格	4
表2.5 MQ-7 一氧化碳 CO 氣體感測器規格	5
表2.6 MQ-5 氣體感測器規格	6

圖目錄

圖2.1 Arduino UNO控制板.....	2
圖2.2 Arduino 1路繼電器模組 5V.....	3
圖 2.3 PMS3003 (G3) 空氣品質 感測器	4
圖2.4 MH-Z16 二氧化碳感測器	5
圖2.5 MQ-7 一氧化碳 CO 氣體感測器.....	6
圖2.6MQ-5 一氧化碳 CO 氣體感測器	6
圖2.7空氣清淨機.....	7
圖 3.1研究流程圖	9
圖3.2主系統架構圖	10
圖3.3系統流程圖	11
圖3.4數據蒐集子系統架構圖.....	12
圖3.5氣體感測器之程式碼	22
圖3.6 PMS3003氣體感測器與繼電器控制之程式碼.....	23
圖3.7 MongoDB資料庫儲存之數據畫面	24
圖3.8數據蒐集子系統流程圖.....	13
圖3.9視覺化即時監控子系統架構圖	14
圖3.10 C3.js圖表建置程式碼	25
圖3.11 C3.js圖表建置程式碼	25
圖3.11 C3.js視覺化數據圖表	26
圖3.12視覺化即時監控子系統流程圖	15
圖3.13異常訊號監控子系統架構圖	16
圖3.14 伺服器判斷異常數值程式碼.....	27
圖3.15 將訊息傳送至使用者line之程式碼	27
圖3.16 Line 警示訊息畫面	27
圖3.14異常訊號監控子系統流程圖.....	17
圖4.1查詢數據實作畫面圖	18
圖4.2視覺化圖表.....	19
圖4.3傳送異常數據簡訊圖	20

第壹章緒論

1.1 研究背景

近年來大數據以及物聯網的發展越來越多元，許多智慧家電都具備物聯網的功能，能夠蒐集許多的數據傳送至網路上，因此我們想結合大數據以及物聯網來將空氣清淨機之數據整合起來，讓數據不會流失掉，以便用於近一步的分析與討論。

1.2 研究動機

近年民眾對於健康的議題逐漸重視，空氣污染的議題更是其中一大家關注的重要議題，造成空氣污染的原因有許多，而空氣污染也會對人體造成許多的危害，新聞常常說到減少外出、加戴口罩等等，但是待在室內其實也沒有比室外來的好，室內一樣會有許多的空氣污染物，目前民眾大多會購買室內空氣清淨機來使用，幫助維持室內的空氣品質，但大多不知道什麼時候是空氣品質不好的，什麼時候是空氣品質好的，市面空氣清淨機大多只有清潔但不具備遠端監控的功能，於是我們想結合空氣清淨機與手機遠端來製作空氣檢測之 APP。

本系統使用 Arduino UNO 板結合空氣感測器連接至空氣清淨機放置室內檢測空氣數據，透過偵測的數據將其顯示至動態網頁圖表，使用者可由網頁端與 APP 端來即時觀看當前室內各項空氣數據，發生異常時將會發送警示訊息，使用者可自行透過網頁端或 APP 來開/關空氣清淨機，達到遠端監控之功能。

1.3 研究目的

1. 利用物聯網蒐集數據。
偵測之數據透過物聯網技術儲存，可方便用於日後推測空氣數據知憑據。
2. 將偵測數據儲存至資料庫。
解決空氣清淨機無法遠端控制問題偵測之數據經過整理後儲存至資料庫，方便未來做更進一步之探討。
3. 將偵測至數據呈現給使用者方便觀看。
目前市面上空氣清淨機只具備清潔的功能，並無法得知數據情況，因此本研究想將偵測的數據呈現出來讓使用者觀看
4. 當偵測的數據異常透過伺服器傳送警示訊息至手機。
使用者在外無法及時了解家中狀況，因此本研究想在數據異常時告知使用者。
5. 解決空氣清淨機無法遠端控制問題。
目前市面上空氣清淨機只能手動在家開啟，因此本研究想做到讓空氣清淨機具備遠端開啟之功能。

第貳章文獻回顧與探討

2.1 硬體設備

2.1.1 Arduino Uno 控制板

Arduino Uno 的運作直流電壓為 5V[1]，他可透過 3 種方式提供電源，USB 連接埠、電源輸入插座、Vin 腳位，本專題採取 USB 供電。

表 2.1 Arduino UNO 控制板規格

名稱	技術規格
工作電壓	5 伏特
輸入電壓	7 到 20 伏特
數字輸入/輸出(I/O)引腳	14
模擬引腳	6
數字輸入/輸出(I/O)引腳的直流電流	20 安培
3.3 伏引腳的直流電流	50 安培
快閃記憶體	32KB
隨機存取記憶體	2KB
EEPROM	1KB
時脈	16 赫茲
尺寸	68.6 毫米 x 53.4 毫米
重量	25 克

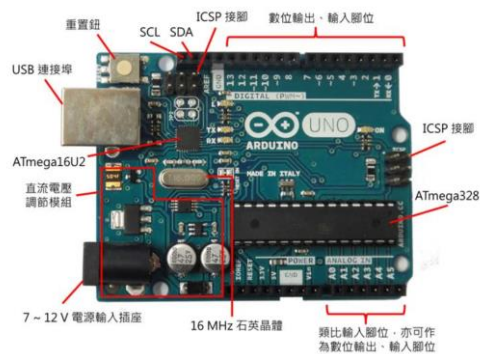


圖 2.1 Arduino UNO 控制板

2.1.2 Arduino 1 路繼電器模組

常開接口最大負載：交流 250V/10A，直流 30V/10A，採用貼片光耦隔離，驅動能力強，性能穩定；觸發電流 5mA，模塊工作電壓有 5V，模塊可以通過跳線設置高電平或低電平觸發。

表 2.2 Arduino 1 路繼電器模組 5V 規格

名稱	技術規格
工作電流	65Ma
觸發電壓	0-2V
觸發電流	2mA
電壓版本	5V
靜態電流	4mA

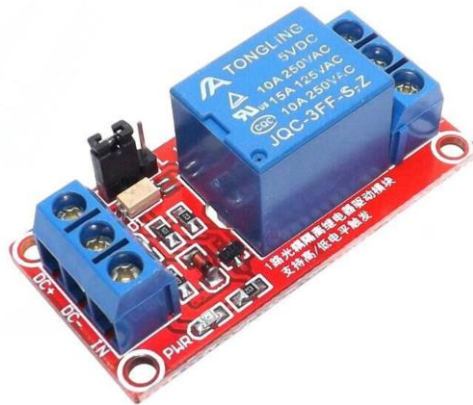


圖 2.2 Arduino 1 路繼電器模組 5V

2.1.3 PMS3003 (G3) 粉塵感測器

PMS3003 粉塵感測器使用積光散射的原理，將光打在空氣中的粉塵上來判斷空氣中粉塵大小，進而得到空氣中 0.3 ~ 10 微米懸浮顆粒物濃度，並內置微處理機處理該訊號，所以收集到的數據穩定可靠[2]。

表 2.3 PMS3003 (G3) 空氣品質感測器規格

名稱	技術規格
測量範圍	0.3~1.0 ; 1.0~2.5 ; 2.5~10 微米
計數倍率	50%@0.3um 98%@>=0.5 um
稱準體積	0.1 升
響應時間	≤10 秒
直流供電電壓	5.0 伏特
最大工作電流	120mA
待機電流	≤200μA
數據接口電平	L <0.8 @3.3 H >2.7@3.3 伏特
工作溫度範圍	-20~+50°C

工作濕度範圍	0~99%
平均無故障時間	≥3 年
最大尺寸	65×42×23 毫米



圖 2.3 PMS3003 (G3) 空氣品質感測器

2.1.4 MH-Z16 二氧化碳感測器

MH-Z16 二氧化碳氣體感測器是小型感測器，它利用非色散紅外原理來對空氣中存在的 CO₂ 進行探測，具有很好的選擇性，無氧氣依賴性，且壽命長。內置溫度補償；同時具有數位輸出與波形輸出，方便使用。該紅外感測器是將成熟的紅外吸收氣體減測技術與精密光路設計、精良電路設計緊密結合而製作出的高性能感測器。

表 2.4 MH-Z16 二氧化碳感測器規格

名稱	技術規格
工作電壓	4.5~5.5 V DC
平均電流	< 85 mA
接口電平	3.3 V
測量範圍	0~5% Vol
預熱時間	3 min
響應時間	T90 < 30 s
工作溫度	0~50 °C
工作濕度	0~95% RH
外型尺寸	97 mm×20 mm×17 mm
重量	21 g
壽命	> 5 年



圖 2.4 MH-Z16 二氧化碳感測器

2.1.5 MQ-7 一氧化碳 CO 氣體感測器

MQ-7 一氧化碳 CO 氣體感測器 MQ-7 氣體感測器所使用的氣敏材料是在清潔空氣中電導率較低的二氧化錫(SnO_2)。採用高低溫循環檢測方式，低溫檢測一氧化碳，感測器的電導率隨空氣中一氧化碳氣體濃度增加而增大，高溫清洗低溫時吸附的雜散氣體。MQ-7 氣體感測器對一氧化碳的靈敏度高，這種感測器可檢測多種含一氧化碳的氣體，是一款適合多種應用的感測器。

表 2.5 MQ-7 一氧化碳 CO 氣體感測器規格

名稱	技術規格
加熱電壓	$5 \pm 0.2\text{V}$ (AC·DC)
工作電流	140mA
回路電壓	10V (最大 DC 15V)
負載電阻	10K
檢測濃度範圍	10-1000ppm
清潔空氣中電壓	$\leq 1.5\text{V}$
回應時間	$\leq 1\text{S}$
元件功耗	$\leq 0.7\text{W}$
工作溫度	$-10 \sim 50^\circ\text{C}$
重量	6g
壽命	> 5 年



圖 2.5 MQ-7 一氧化碳 CO 氣體感測器

2.1.6 MQ-5 天然氣氣體感測器

MQ-5 氣體感測器可用於氣體洩漏檢測，可檢測液化石油氣，天然氣，城鎮煤氣等。基於其快速響應時間，測量可以盡快進行。且其靈敏度也可以通過電位器來進行調節。

表 2.5 MQ-5 氣體感測器規格

名稱	技術規格
工作電壓	4.9V~5.1V
熱能耗	0.5mW~800mW
監測濃度	200ppm~100000ppm



圖 2.6MQ-5 氣體感測器

2.1.7 空氣清淨機

從淨化原理來看，市面上空氣清淨機的淨化技術主要有四種：靜電集塵型、過濾阻隔型、光觸媒型和負離子型。負離子型空氣清淨機利用強靜電場使空氣中的分子電離出自由電子，能與空氣中的氧分子結合成負氧離子（負離子），同時與空氣中的病菌細胞結合使其內部結構改變、死亡，並聚集包覆甲苯、甲醛等顆粒物質，自然沉降後達到淨化目的。[3]

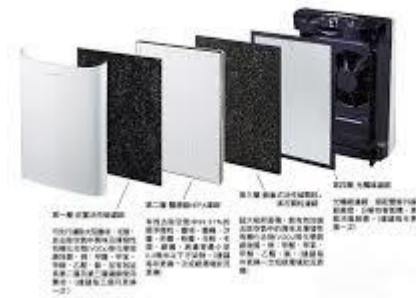


圖 2.7 空氣清淨機

2.2 軟體

2.2.1 Arduino IDE

IDE 的英文全名是 Integrated Development Environment，中文譯為：整合開發環境[4]。它功能眾多，在一個畫面中既可以編輯程式碼文字，也可以編譯、連結、除錯等，因為把原本各自散落的程式開發工具統整在一起，也因為都在同一個程式操作畫面，什麼需要的工具都齊備了，彷彿置身於一個開發「環境」內，所以稱為 IDE。

2.2.2 Node.js

Node.js 是以 JavaScript 語言為基礎，是一個開放的原始碼的應用程式框架[5]，它不需要經過太多複雜的調校、效能調整及程式修改，就能滿足網路服務在不同發展階段對效能的要求，在 Node.js 中使用者可以自己寫數 Web Server，並且擁有高效能、低耗能、支援第三方模組等功能。

2.2.3 C3.js

C3.js 是 D3.js 中的其中一個延伸，C3.js 的使用非常的簡單，而且互動性很強大[6]。它可以讓使用者將數據以圖表化的方式呈現出來，讓觀看人可以方便的解讀。

2.2.4 MongoDB

MongoDB 是 noSQL 的一種，特性是分散、欄位自由、資料格式的一致，使用 BSON 的格式來存取資料，良好的支援，open source 全部的程式都開放給你下載支援分散式資料庫-你可以把資料庫放在不同主機[7]。

2.2.5 HTML5

HTML5 是 HTML 最新的修訂版本，由全球資訊網協會於 2014 年 10 月完成標準制定。目標是取代 1999 年所制定的 HTML 4.01 和 XHTML 1.0 標準，以期能在網際網路應用迅速發展的時候，使網路標準達到符合當代的網路需求[8]。

2.2.6 JavaScript

JavaScript，通常縮寫為 JS，是一種進階的，解釋執行的程式語言。JavaScript 是一門基於原型、函式先行的語言，是一門多範式的語言，它支

援物件導向編程，指令式程式設計，以及函數語言程式設計。[9]

2.2.7 IFTTT

它是一個新生的網路服務平台，可以透過其他不同的平台的條件來決定是否執行下一條命令，口號為「if this then that」[10]，在本專題執行的條件觸發為若數值過高時。

2.3 檢測物

2.2.8 PM2.5

空氣懸浮粒子，大小約為 PM10 的 1/4 倍[11]，它無法被鼻腔阻擋，如果長期吸入會沉積在人體中造成支氣管發炎。

2.2.9 PM10

空氣懸浮粒子，大小約為人體頭髮的直徑，會被阻擋在鼻腔造成過敏性鼻炎，引發咳嗽、氣喘等症狀。[12]

2.2.10 CO₂

空氣化合物，二氧化碳，人體呼出之氣體，當二氧化碳濃度過高時，除了會刺造成呼吸費力或困難等感覺，亦會產生頭痛、嗜睡、反射減退、倦怠等症狀。

2.2.11 CO

一氧化碳為無色、無臭、無味的無機化合物氣體，主要來自於吸煙或是室外廢氣進入，長期暴露於一定程度的一氧化碳濃度中，會呈現頭暈、頭痛、呼吸困難、頻尿、口渴、體重減輕、失眠、易怒等慢性中毒症狀。

2.2.12 CH₄

甲烷它在常溫常壓下是一種無色無味的氣體，家中使用的天然氣為了方便查看洩漏而添加了人工氣味，甲烷它非常容易爆炸，因此本專題將其列為檢測之物質之一。

第參章研究方法

3.1 研究架構

1.規劃:

確定主題:空氣議題日漸增加，本專題使用空氣感測器結合空氣清淨機，將偵測之資料收集成數據，加以分析，使用物聯網之技術，使其使用者方便即時監控當前室內之空氣數據。

2.分析:

蒐集資訊:查詢如何使用 HTML 語言架設網頁，並將檢測之空氣數據透過網路傳送至網頁，並持續更新所偵測之值，讓使用者可以及時掌控室內空氣品質。

資料分析:將所蒐集之資料詳細研讀，逐步測試至程式。

3.設計:

系統撰寫:使用 C3.js 將偵測之數據以圖表之方式呈現於網頁中，使用折線圖來呈現所偵測知數據，並會持續變動，顯示即時數據，並使用 CSS 來排版美化。

系統測試:反覆測試畫面是否能完整檢視，如有異常將持續進行修改。

4.建置:

系統撰寫:前端以 HTML 語言與 JavaScript 所撰寫，使用 socket.io 之程式碼將偵測之值傳送至前端圖表，並視覺化數據，讓數據顯示至網頁圖表，方便查看。

系統測試:測試偵測之數據與傳送至網頁之圖表是否顯示異常，是否顯示所有偵測之數據，如有異常持續修改。

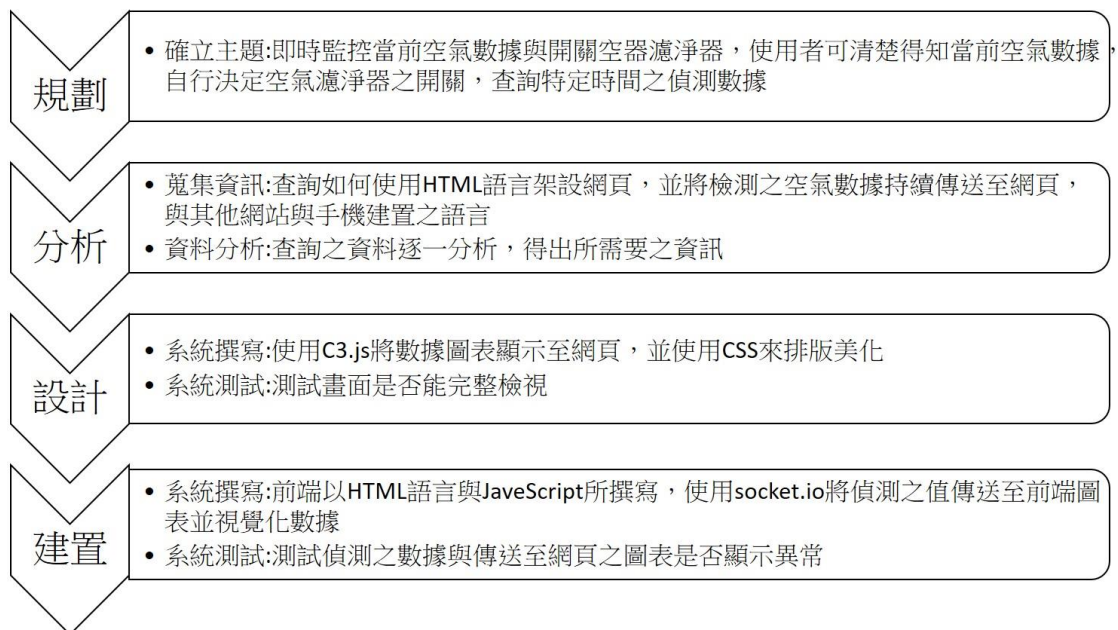


圖 3.1 研究流程圖

3.2 「PM2.5 手機 app 網頁即時監測與控制架構圖」

本系統主架構如圖 3.2 所示，分為三個子系統，（1）「數據蒐集」子系統；（2）「視覺化即時監視」子系統；（3）「異常訊號監控」子系統。

主系統架構圖

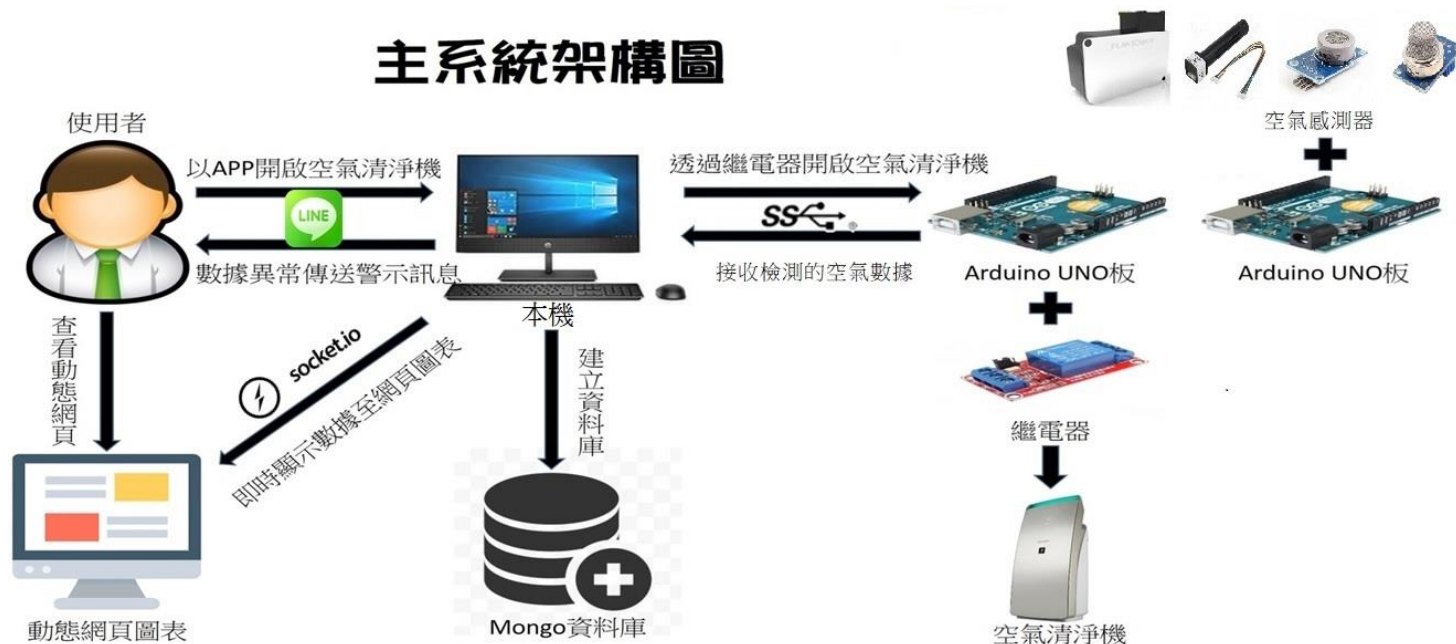


圖 3.2 「PM2.5 手機 app 網頁即時監測與控制架構圖」架構圖

3.3 「PM2.5 手機 app 網頁即時監測與控制架構圖」 流程圖

本系統主系統流程圖操作如下，將程式撰寫至 Arduino，分別連接空氣感測器與繼電器，繼電器端連接至空氣清淨機，可以隨時控制空氣清淨機之開啟，感測器端，Uno 板透過 USB 連接至本機，本機會接收感測器上所偵測的空氣數據，判斷是否成功接收後同時純存到資料庫與傳送至動態網頁，再由本機判斷數據是否異常，如異常將連接至空氣清淨機來決定是否開關空氣清淨機。

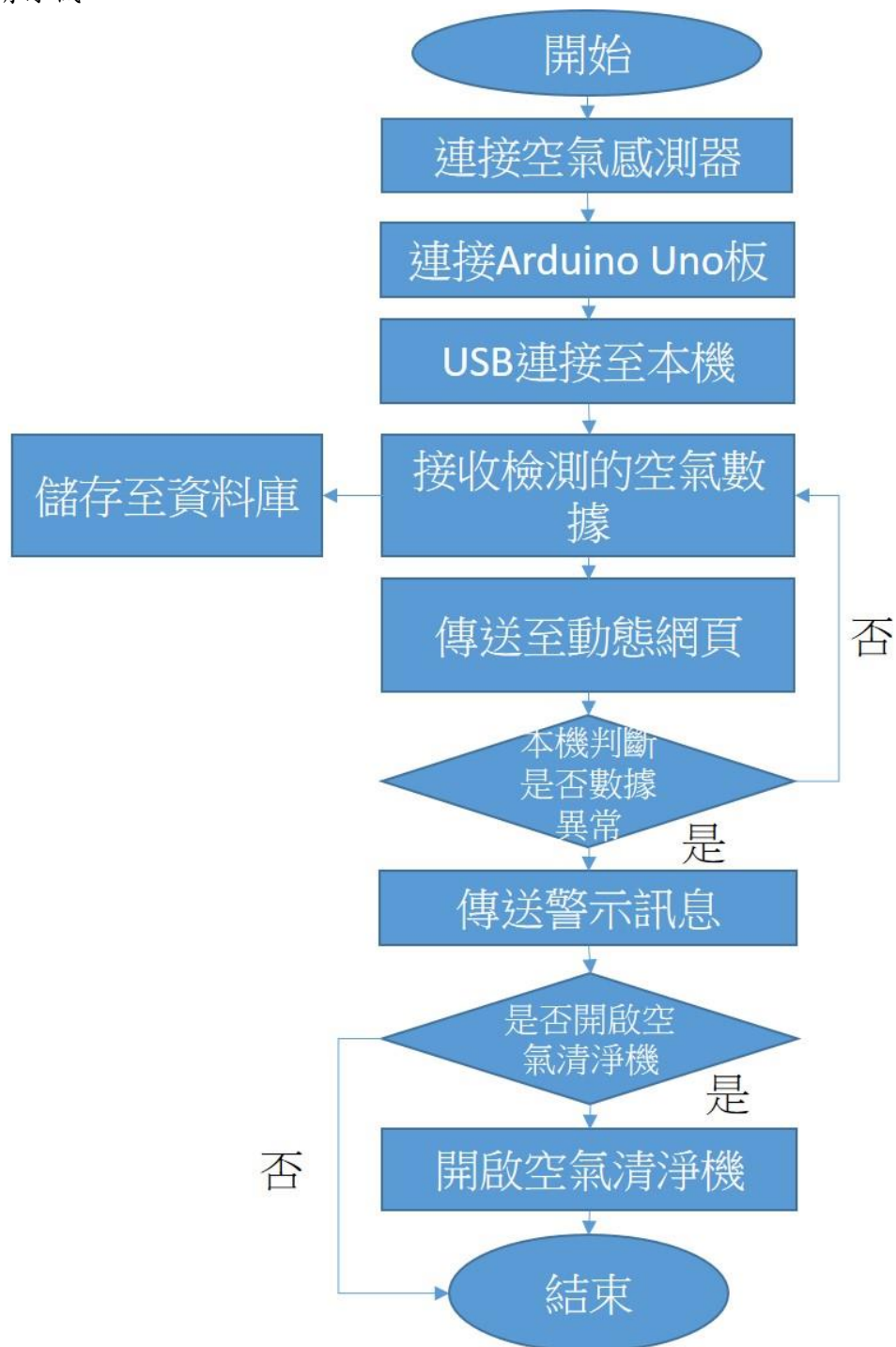


圖 3.3 「PM2.5 手機 app 網頁即時監測與控制架構圖」 流程圖

3.4 「數據蒐集」子系統

數據蒐集子系統使用電腦當作系統伺服器，透過 USB 線接上 ArduinoUNO 板，使用板子上的感知元件偵測環境中 PM2.5、PM10、CO、CO₂、CH₄的濃度含量，並將蒐集到的數據傳送到電腦建置的資料庫中



圖 3.4 「數據蒐集」子系統架構圖

3.6 「數據蒐集」子系統流程圖

本系統之數據蒐集子系統操作流程如下，UNO 板連接空氣感測器，檢測完數據透過 USB 連接至本機，本機接收檢測之數據將檢測之值儲存至資料庫之中。



圖 3.8 「數據蒐集」子系統流程圖

3.7 「視覺化即時監控」子系統

視覺化即時監視子系統利用電腦當作系統伺服器，本機透過 USB 連接 ArduinoUNO 板，接收偵測之數據，並將偵測到的數據即時傳送至前端網頁繪製成動態圖。



圖 3.9 「視覺化即時監控」子系統架構圖

3.9 「視覺化即時監控」子系統系統流程圖

本系統之視覺化即時監控子系統操作流程如下，UNO 板連接空氣感測器，檢測當前空氣數據，透過 USB 連接至本機，本機接收偵測之數據後以 socket.io 將資料傳輸至網頁端，再使用 C3.js 將資料繪製成動態網頁圖表，使用者可以在網頁上觀看動態圖表之資訊。



圖 3.12 「視覺化即時監控」子系統流程圖

3.10 「異常訊號監控」子系統

異常訊號監控子系統透過本機判斷生產數據是否達到標準值，根據環境空氣數據發生異常數據時，以 line 傳送警示訊息通知使用者，使用者可透過遠端開啟空氣濾淨器。



圖 3.13 「異常訊號監控」子系統架構圖

3.12 「異常訊號監控」子系統流程圖

本系統之異常訊號監控子系統操作流程如下，以 Arduino 分別連接空氣感測器與繼電器，本機抓取檢測之值判斷檢測之值是否達標準值，如發生異常會傳送簡訊給管理者，管理者可決定是否使用 APP、網頁端遠端控制空氣清淨機開啟。

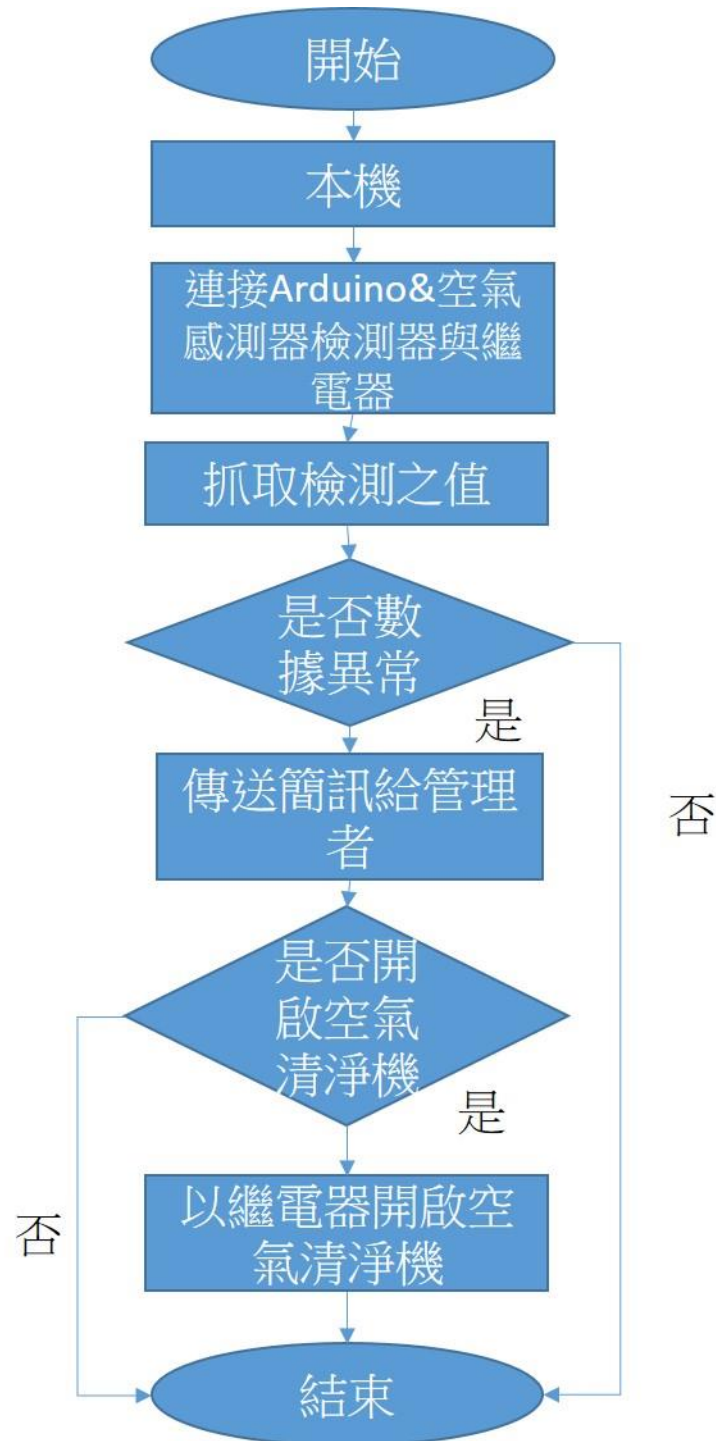


圖 3.17 「異常訊號監控」子系統流程圖

第肆章部分實體畫面

4.1 系統實作畫面

4.1.1 查詢數據實作畫面

查詢資料數據					
所有資料數據 ▾		確認			
pm2.5	pm10	CO ₂	CO	CH ₄	時間
18μg/m3	21μg/m3	1027ppm	212ppm	53ppm	2019年11月01日13時35分37秒
0μg/m3	90μg/m3	1027ppm	212ppm	53ppm	2019年11月01日13時35分36秒
0μg/m3	0μg/m3	1027ppm	212ppm	53ppm	2019年11月01日13時35分35秒
1μg/m3	0μg/m3	1027ppm	212ppm	53ppm	2019年10月30日19時15分29秒

圖 4.1 查詢數據實作畫面圖

此圖為本系統之查詢數據功能實作圖，可在此表格下拉式選單選擇想調閱隻數據進行查看，了解個時間點的數據變化。

4.1.2 視覺化圖表畫面



4.1.3 傳送異常數據簡訊畫面



圖 4.3 傳送異常數據簡訊圖

此圖為本系統之傳送異常數據簡訊圖，如果數據有超出所設定之值，將透過 IFTTT 發送一則 LINE 簡訊給使用者的 LINE，讓使用者可以前往監控畫面並決定是否開關空氣清淨機。

第伍章結論

1. 以手機APP、網頁端隨時遠端開關空氣清淨機

在網頁以及手機 app 中，使用者可以隨時將室內之空氣清淨機作開啟/關閉之動作。

2. 檢測數據異常傳送警示訊息至使用者Line

本機如檢測出當前空氣數據為超過設定之數值，將透過寫入之程式透過 IFTTT 將超標之訊息傳送給使用者，使用者可以及時得知當前室內空氣數據為超標數值。

3. 將偵測的數據儲存至資料庫

偵測之數據將逐一寫入 mongo 資料庫，如有需要時方便調閱。

4. 增加偵測CO、CO₂、CH₄

加購感測器增加了偵測的數據，分別為 CO、CO₂、CH₄

5. 加入搜尋功能

網頁端與手機 app 端使用者可以使用選單選取想觀看之資料，調閱偵測之空氣數據資料，來查看過往數據

附錄

6.1 Arduino 程式碼

圖 6.1 中引用 CO2 感測器函式庫，設定感測器接收訊號源腳位及串列埠鮑率，將感測器接收到之信號源套用感測器公式轉換成對應之單位。

```
1  #include <NDIRZ16.h>
2  #include <SoftwareSerial.h>
3  #define analogMQ7 A0
4  SoftwareSerial mySerial(9,8); //(Co2Rx, Co2Tx)-->Arduino
5  NDIRZ16 myco2 = NDIRZ16(&mySerial);
6  int sensor=A1;
7  int gas_value;
8  int CO = 0;
9  void setup()
10 {
11     Serial.begin(9600);
12     mySerial.begin(9600);
13 };
14 void loop() {
15     float sensor_volt;
16     float RS_gas;
17     int CH4;
18     //----CO
19     analogWrite(analogMQ7, HIGH);
20     CO = analogRead(analogMQ7);
21     //----
22     //----CH4
23     int sensorValue = analogRead(A2);
24     sensor_volt=(float)sensorValue/1024*5.0;
25     RS_gas = (5.0-sensor_volt)/sensor_volt; // omit *RL
26     CH4 = RS_gas/0.02; // ratio = RS/R0
27     //-----
28     if (myco2.measure()) {
29         if(myco2.ppm<1000){
30             Serial.print("1");
31             Serial.print("0");
32             Serial.print("0");
33             Serial.print(myco2.ppm);
34         }
35         else if(myco2.ppm>=10000){
36             Serial.print("1");
37             Serial.print(myco2.ppm);
38         }
39     }
40     else{
41         Serial.print("1");
42         Serial.print("0");
43         Serial.print(myco2.ppm);
44     }
45 }
46 if(CO<10){
47     Serial.print("0");
48     Serial.print("0");
49     Serial.print("0");
50     Serial.print(CO);
51 }
52 else if(CO<100&&CO>=10){
53     Serial.print("0");
54     Serial.print("0");
55     Serial.print(CO);
56 }
57 else if(CO<1000&&CO>100){
58     Serial.print("0");
59     Serial.print(CO);
60 }
61 else if(CO>1000){
62     Serial.print(CO);
63 }
64 if(CH4<10){
65     Serial.print("0");
66     Serial.print("0");
67     Serial.println(CH4);
68 }
69 else if(CH4>=10&&CH4<100){
70     Serial.print("0");
71     Serial.println(CH4);
72 }
73 else if(CH4>100){
74     Serial.println(CH4);
75 }
76 delay(500);
}
```

圖 6.1 氣體感測器之程式碼

圖 6.2 中引用 PMS3003 空氣懸浮粒子感測器函式庫，設定感測器訊號源腳位及，串列埠鮑率，資料位元組則從第三位元組到第二十二位元組，根據感測器傳輸資料格式得知我們需要第十三及第十五位元組，最後設計當 Arduino 接收到零時，第八腳位高電位，否則低電位，控制繼電器進而控制空氣濾淨器。

```

1  #include <SoftwareSerial.h>
2  SoftwareSerial Serial1(6,7);
3  int RELAYpin = 8;
4  long pmat10=0;
5  long pmat25=0;
6  long pmat100=0;
7  char buf[50];
8  void setup() {
9      Serial.begin(9600);
10     Serial1.begin(9600);
11     pinMode(RELAYpin, OUTPUT);
12     digitalWrite(RELAYpin, LOW);
13 }
14 void loop() {
15     int count = 0;
16     unsigned char c;
17     unsigned char high;
18     while (Serial1.available()) {
19         c = Serial1.read();
20         if((count==0 && c!=0x42) || (count==1 && c!=0x4d)){
21             break;
22         }
23         else if(count == 4 || count == 6 || count == 8 ||
24             count == 10 || count == 12 || count == 14){
25             high = c;
26         }
27         else if(count == 11){
28         }
29         else if(count == 13){
30             pmat25 = 256*high + c;
31             if(pmat25<10){
32                 Serial.print("0");
33                 Serial.print("0");
34                 Serial.print(pmat25);
35             }
36             else{
37                 Serial.print("0");
38                 Serial.print(pmat25);
39             }
40         }
41         else if(count == 15){
42             pmat100 = 256*high + c;
43             if(pmat100<10){
44                 Serial.print("0");
45                 Serial.print(pmat100);
46             }
47             else{
48                 Serial.print(pmat100);
49             }
50         }
51         if (Serial.available()){
52             if (char(Serial.read()) == '0'){
53                 digitalWrite(RELAYpin, HIGH);
54             }else{
55                 digitalWrite(RELAYpin, LOW);
56             }
57         }
58         count++;
59     }
60     while(Serial1.available())Serial1.read();
61     Serial.println();
62     delay(500);
63 }

```

圖 6.2 PMS3003 氣體感測器與繼電器控制之程式碼


```
系統管理員: C:\Windows\SysWOW64\cmd.exe - mongo
>
<
  "_id" : ObjectId("5de0aa4fbe9c5b04bc5302b2"),
  "pm25" : 9,
  "pm10" : 12,
  "CO2" : 417,
  "CO" : 12,
  "CH4" : 9,
  "Time" : "2019年11月29日13時19分11秒"
>
<
  "_id" : ObjectId("5de0aa50be9c5b04bc5302b3"),
  "pm25" : 9,
  "pm10" : 11,
  "CO2" : 417,
  "CO" : 12,
  "CH4" : 9,
  "Time" : "2019年11月29日13時19分12秒"
>
<
  "_id" : ObjectId("5de0aa51be9c5b04bc5302b4"),
  "pm25" : 9,
  "pm10" : 11,
  "CO2" : 417,
  "CO" : 12,
  "CH4" : 9,
  "Time" : "2019年11月29日13時19分13秒"
>
<
  "_id" : ObjectId("5de0aa52be9c5b04bc5302b5"),
  "pm25" : 9,
  "pm10" : 11,
  "CO2" : 417,
  "CO" : 12,
  "CH4" : 9,
  "Time" : "2019年11月29日13時19分14秒"
>
<
  "_id" : ObjectId("5de0aa53be9c5b04bc5302b6"),
  "pm25" : 10,
  "pm10" : 13,
  "CO2" : 417,
  "CO" : 12,
  "CH4" : 9,
  "Time" : "2019年11月29日13時19分15秒"
>
```

圖 6.3 MongoDB 資料庫儲存之數據畫面

6.2 C3.js 程式碼

圖 6.4 中根據所需要之參數物件創建圖表，例如 id 為 chart3、最小值為 0 對大值為 3000、根據值的高低改變顏色、單位為 ppm 等等....，若未指定則會使用預設值創建。

```
chart3.flow({
  columns:[
    ["日期", Time(1)],
    ["CO2", CO2A]
  ],
  length: (tckCount < totalTck) ? 0 : 1,
  done: function(){
    if(tckCount<totalTck) ++tckCount;
  }
});
}, 1000);
```

圖 6.4 C3.js 圖表建置程式碼

圖 6.5 中將資料寫入所創建之 C3.js 圖表中，資料格式為整數，並且持續寫入資料達到動態圖表之效果。

```
var chart3 = new JustGage({
  id : "chart3",
  value : 0,
  symbol : "ppm",
  pointer : true,
  gaugeWidthScale:0.8,
  max:3000,
  min:0,
  color:{
    pattern:['#00ff00','#ffff00','#ff0000'],
    threshold:{
      unite:'value',
      max:100000,
      values:[500,1000,3000]
    }
  },
  counter:true
});
```

圖 6.5 C3.js 圖表建置程式碼

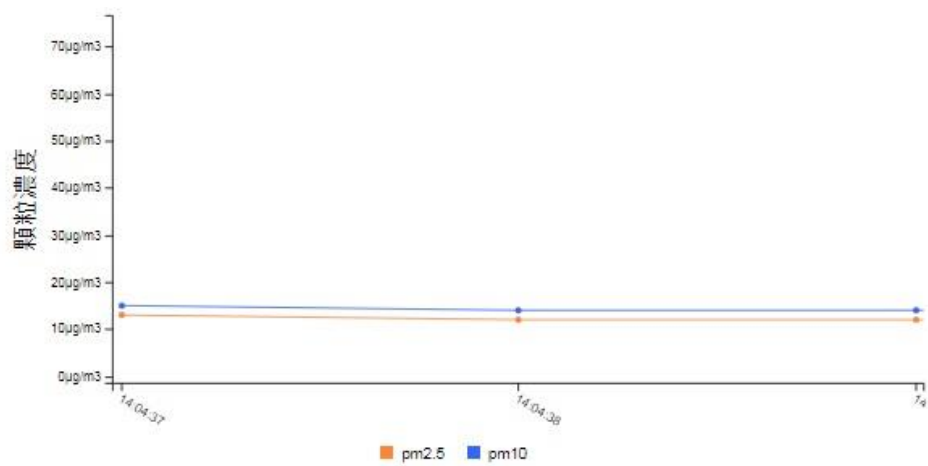


圖 6.6 C3.js 視覺化數據圖表

6.3 判斷異常數據並傳送警示訊息之流程

當數據發生異常時會經由 3 個步驟將警示訊息傳送給使用者，第一步驟如圖 6.7 由伺服器端判斷數據異常第二步驟如圖 6.8 透過 IFTTT 將警示訊息傳送至使用者 line，最後使用者的 line 會收到如圖 6.9 之訊息畫面。

```
if(CO2>=1000){worco2=1;}else{worco2=0;};  
if(NH4>=300){wornh4=2;}else{wornh4=0;};  
if(pma>=30){worpma=3;}else{worpma=0;};  
if(pmb>=75){worpmb=4;}else{worpmb=0;};  
socket.emit("WORRR",{ 'worco2':worco2.toString(), 'wornh4':wornh4.toString(), 'worpma':worpma.toString(), 'worpmb':worpmb.toString() });
```

圖 6.7 伺服器判斷異常數值程式碼

```
case 1:$.post('https://maker.ifttt.com/trigger/line/with/key/vEQJQvj5vaYd4G0olPg8j',{value1:'二氧化碳數值:'+CO2A+'ppm'+<br>+ '濃度過高，請打開門窗讓室內通風，並請留意室內空氣品質是否須開啟空氣清淨機'});break  
case 2:$.post('https://maker.ifttt.com/trigger/line/with/key/vEQJQvj5vaYd4G0olPg8j',{value1:'甲烷數值:'+NH4A+'ppm'+<br>+ '疑似瓦斯外洩!請勿點火，迅速打開門窗，並檢查瓦斯設備'});break  
case 3:$.post('https://maker.ifttt.com/trigger/line/with/key/vEQJQvj5vaYd4G0olPg8j',{value1:'PM2.5濃度:'+pmat1+'ug/m3'+<br>+ 'PM2.5濃度超標，建議開起空氣濾淨器'+<br>+ '前往控制網頁:'+http://192.192.2.41:3000'});break  
case 4:$.post('https://maker.ifttt.com/trigger/line/with/key/vEQJQvj5vaYd4G0olPg8j',{value1:'PM10濃度:'+pmat2+'ug/m3'+<br>+ 'PM10濃度超標，建議開起空氣濾淨器'+<br>+ '前往控制網頁:'+http://192.192.2.41:3000/indexx.html'});
```

圖 6.8 將訊息傳送至使用者 line 之程式碼



圖 6.9 Line 警示訊息畫面

參考文獻

- [1] “Arduino 的互動產品平台創新設計” 楊繼志， 郭敬，
楊繼志- 2012 - mesnet.com.cn
- [2] “An open framework for participatory PM2. 5 monitoring in
smart cities”LJ Chen, YH Ho, HC Lee, HC Wu, HM Liu... -
IEEE ..., 2017 - ieeexplore.ieee.org
- [3] “空氣清淨機裝置” 楊家銘， 楊家銘 - 2005 -
nthur.lib.nthu.edu.tw
- [4] “Arduino”SA Arduino - Arduino LLC, 2015 - csulb.edu
- [5] “Node. js: Using JavaScript to build high-performance network
programs”S Tilkov, S Vinoski - IEEE Internet Computing,
2010 - ieeexplore.ieee.org
- [6] T Bogaard, J Wielemaker, L Hollink... - Benelux Conference
on ..., 2016 – Springer
- [7] “MongoDB in action”K Banker - 2011 - dl.acm.org
- [8] “Enrichr: interactive and collaborative HTML5 gene list
enrichment analysis tool”EY Chen, CM Tan, Y Kou, Q
Duan... - BMC ..., 2013 - bmcbioinformatics.biomedcentral
- [9] “JavaScript: the definitive guide”D Flanagan - 2006 -
books.google.com
- [10] “Automate the internet with “if this then that”(IFTTT)”S
Ovadia - Behavioral & social sciences librarian, 2014 - Taylor
& Francis
- [11] 基因叔叔 痞克幫 Jun 16 Fri 2017”空氣汙染（PM2. 5和PM10）
對身體有什麼影響？”