



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系

空氣品質偵測系統製作

指導教授： 陳明華 教授

組員名單： 林 立 A38B002

巫國華 A38B037

楊振嘉 A38B059

中 華 民 國 1 0 7 年 5 月



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

空氣品質偵測系統製作

指導教授： 陳明華 教授

組員名單： 林 立 A38B002
巫國華 A38B037
楊振嘉 A38B059

指導教授：

口試委員：

中 華 民 國 1 0 7 年 5 月

嶺東科技大學

資訊管理系

空氣品質偵測系統製作

民國一〇柒年伍月

謝 誌

本專題報告得以順利於 107 年 5 月完成，首先要感謝恩師陳明華老師細心引導我們，耐心的協助我們，克服研究過程中所面臨的困難，給予我們最大的協助，使本專題得以順利完成。

報告口試期間，感謝沈坤耀老師、林孟源老師不辭辛勞細心審閱，不僅給予我們指導，並且提供寶貴的建議，使我們的專題內容以更臻完善，在此由衷的感謝。

最後，感謝系上諸位老師在各學科領域的熱心指導，以及口試委員們提供的協助與關心，給予我們許多的建議與教學，在此一併致上最高謝意。

林立 巫國華 楊振嘉
謹誌

中華民國一百零七年五月於嶺東

摘要

本專題使用 Arduino UNO R3 開發板搭配 SHARP PM10 懸浮微粒感測器與 DHT22 溫溼度模組來偵測所在環境數值，並搭配藍芽 HC-06 回傳數值至行動裝置軟體，使用者藉由 APP 介面功能來得知回傳的數值。

本專題主要功能為硬體的偵測數值 PM10、溫度、濕度為主，並搭配 App 軟體的線上抓取數據功能，讓使用者能夠透過硬體裝置得知所在區域的環境品質，並藉由 App 介面來整合感測器的數值與大數據。

本專題組裝一個能夠藉由藍芽得知所在環境品質的行動探測器，並搭配網路上環保局的數據可以直接搜尋各大城市的環境，藉由 App 能夠在儀器偵測到過於危險的環境時，能夠提醒使用者的空氣品質超標。

關鍵字：空氣品質、懸浮微粒、Arduino UNO R3、PM10、DHT22、HC-06。

目錄

謝誌	I
摘要	II
目錄	III
圖目錄	IV
表目錄	V
第壹章 緒論	1
1.1 研究動機	1
1.2 研究目的	1
第貳章 文獻回顧與探討	2
2.1 空氣品質介紹與分析	2
2.1.1 何謂 PM 10	2
2.1.2 PM 10 來源說明	2
2.1.3 PM 10 對人體影響	2
2.1.4 全台空氣品質分析與數據	3
2.1.5 AQI 空氣品質指標	3
2.1.6 空氣污染的防範與應變	4
2.2 Arduino IDE	4
2.2.1 Arduino 的介紹	4
2.2.2 Arduino 的起源	5
2.2.3 Arduino IDE 介紹	6
2.2.4 Arduino UNO R3 開發板	6
2.3 藍牙模組 HC-06	7
2.4 Android 手機	7
2.5 MIT App Inventor 2	7
2.6 DHT22 溫溼度感測器模組	9
2.7 SHARP PM10 空氣品質感測器	9
第參章 實作方法	10
3.1 訂立問題與實作目的	10
3.2 實作架構	10
3.3 空氣品質系統架構	11
3.4 系統功能解釋	11
3.5 實作流程	11
3.6 儀器偵測步驟與結果	12
第肆章 實作成果	13
4.1 實體與電路圖說明	13
4.2 Arduino UNO R3 程式碼	14
4.3 硬體測試校準	16
4.4 AppInventor2 程式碼	18
4.5 App 操作介紹	21
第伍章 結論與建議	23
參考文獻	24

圖目錄

圖 2.1 各縣市 PM2.5 排名	3
圖 2.2 Arduino IDE 的介面說明	6
圖 2.3 Arduino UNO R3 開發板圖片	6
圖 2.4 藍牙模組 HC-06 圖片	7
圖 2.5 DHT22 溫溼度感測器模組圖片	9
圖 2.6 SHARP PM10 空氣品質感測器圖片	9
圖 3.1 實作流程圖	11
圖 3.2 儀器操作示意圖	12
圖 3.3 Arduino 上顯示之數據	12
圖 4.1 實體儀器圖	13
圖 4.2 儀器電路設置圖	13
圖 4.3 程式宣告與匯入程式圖	14
圖 4.4 程式執行頻率設置圖	14
圖 4.5 程式 void loop 設置圖	15
圖 4.6 傳輸封包設定圖	15
圖 4.7 藍芽傳送迴圈設置圖	16
圖 4.8 Arduino 上顯示之數據	16
圖 4.9 數據回歸曲線圖	17
圖 4.10 清單選擇器設置	18
圖 4.11 按鈕與 App 初始化設定	18
圖 4.12 選擇藍芽與執行錯誤	18
圖 4.13 傳送簡訊按鈕程式	19
圖 4.14 藍芽傳送封包解析與設置圖	19
圖 4.15 程序定義圖	20
圖 4.16 抓取文本設置圖	20
圖 4.17 App 操作架構圖	21
圖 4.18 基本操作步驟示意圖	21
圖 4.19 連接儀器操作示意圖	22

表目錄

表 2.1 粒徑大小說明	2
表 2.2 懸浮微粒大小對人體影響	3
表 2.3 版本功能比較表	8
表 4.1 健康指數表	17
表 4.2 數據統計表	17

第壹章 緒論

此章節將說明我們為什麼想要去製作關於空氣品質偵測系統的動機與目的，並簡單的介紹我們的設計想法。

1.1 研究動機

空氣汙染是全世界的危害，現在地球汙染日漸嚴重，空氣中的懸浮微粒日漸增多，造成人們呼吸道的嚴重危害，遠從中國大陸席捲而來的沙塵陰影，更開始壟罩台灣，連環保署都對台灣的空氣品質惡化發出警告。今日，台灣空氣品質雖然已有改善，但是效果仍有限。

一般公共環境缺乏監測空氣系統，無法及時給予民眾與住戶及時的空氣只能自備小型的空氣偵測器來使用，大多都只偵測幾項空氣數值而資訊，民眾也無法提供更多且更精細的數據。

1.2 研究目的

本專題決定以一般大樓為主要開發製作一個裝置連結 App 為基底的空氣品質查詢，提供各功能在此 App 裡，只要在使用此 App 就能夠輕鬆得知此地區的空氣品質甚至健康指數，民眾們不用再花時間去網路上查詢，甚至帶著體積龐大的機器到處行走，這樣不但能節省時間也能隨時得知資訊內容。

設計想法:方便民眾查詢該地空氣汙染、與健康指數，多功能空氣品質偵測氣來連結App，得知空氣品質。

適用對象:本App適用於一般需要得知空氣品質的民眾，甚至有呼吸道過敏的患者來使用。

第貳章 文獻回顧與探討

此章節將介紹我們專題中，參考的資料與文獻，說明空氣品質的分析與使用到儀器的介紹。

2.1 空氣品質介紹與分析

一年比一年差的空氣品質究竟從何而來，又會帶給人們甚麼影響，將在此章節讓各位認識空氣中的PM10與台灣各地區的汙染程度。

2.1.1 何謂 PM 10

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，介於2.5微米(μm)的粒子至10微米(μm)的粒子，就稱為PM10，通稱懸浮微粒，單位以微克/立方公尺($\mu\text{g}/\text{m}^3$)表示之，它的直徑還不到人的頭髮絲粗細的1/20，非常微細可被吸入附著於人體呼吸系統，故對人體及生態所造成之影響是不容忽視的[1]。

表 2.1 粒徑大小說明

粒徑(μm)	粒徑大小說明
<100	稱總懸浮微粒(TSP)，約為海灘沙粒，可懸浮於空氣中。
<10	稱懸浮微粒(PM10)，約為沙子直徑 1/10，容易通過鼻腔之鼻毛與彎道到達喉嚨。
2.5~10	稱粗懸浮微粒(PM2.5~10)，約頭髮直徑的 1/8~1/20 大小，可以被吸入並附著於人體呼吸系統。
<2.5	稱細懸浮微粒(PM2.5)，約頭髮直徑的 1/28，可穿透肺部空氣，直接進入血管中隨著血液循環全身。

2.1.2 PM 10 來源說明

懸浮微粒(PM10)來源可分為自然及人為產生，自然產生經由火山爆發、地殼變動或風化等作用形成，人為產生如工業行為、燃燒等。衍生性PM10來源極基本上由前趨化學物質，經光化學反應後生成，一般來說黑煙、氣膠、霧、霾等均有可能為污染物或前趨物，其化學成分組成包含了離子物質、金屬元素、有機碳類、硫、硝酸鹽類等。

空氣汙染、PM10紫爆並非單一原因所造成的，目前研究認為，懸浮粒子成分複雜，主要是工廠排放的廢氣、地面上的塵土、海鹽、無廢氣處理的柴油車，甚至是燃燒垃圾的氣體都是汙染與PM10的來源。而室內，則是以塵蟎與二手菸為主要來源。[1]

2.1.3 PM 10 對人體的影響

空氣中的懸浮微粒會經由鼻、咽及喉進入人體，10微米以上的微粒可由鼻腔去除，較小的微粒則會經由氣管、支氣管經肺泡吸收進入人體內部。不同粒徑大小的懸浮微粒，可能會導致人體器官不同的危害[1]。

表 2.2 懸浮微粒大小對人體影響

粒徑(μm)	分佈特性	對人體生理的影響
>10	沉積於鼻咽	容易造成過敏性鼻炎，引發咳嗽、氣喘等症狀
2.5~10	沉積於上部鼻腔與深呼吸道	造成纖維麻痺、支氣管黏膜過度分泌、使黏液腺增生，引起可逆性支氣管痙攣，抑制深呼吸、並蔓延至小支氣管
<2.5	10% 以下沉積於支氣管，約20~30%於肺泡	形成慢性支氣管炎、細支氣管擴張、肺水腫或支氣管纖維化等症狀
<0.1	沉積於肺泡組織內	促使肺部之巨噬細胞明顯增加，形成肺氣腫並破壞肺泡

2.1.4 全台空氣品質分析與數據

台灣的空氣愈來愈髒，隨時隨地都存在於民眾身旁的懸浮微粒，讓許多人都深受其害，以下是環保署2014~2016年的各縣市PM2.5指數統計與排名，由下圖可知雲林縣為全台空氣品質最差的縣市，台東為最好[1]。

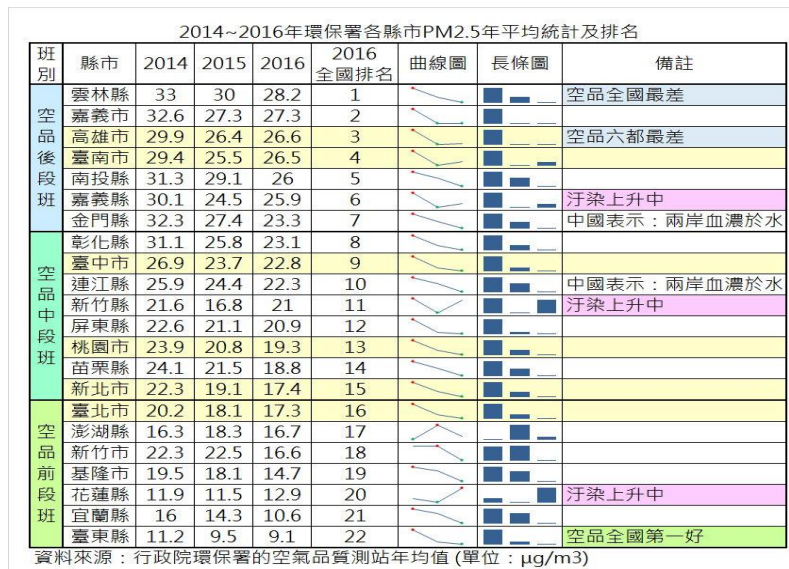


圖 2.1 各縣市 PM 2.5 排名

2.1.5 AQI空氣品質指標

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)。[2]

2.1.6 空氣汙染的防範與應變

越來越差空氣品質需要大家一起來做防範，甚至在超標的時候做出應變來改善。

平日空污管作法

- ◆ 總量管制指定削減。
- ◆ 燃燒設備加嚴標準。
- ◆ 汰換老舊車輛補助。
- ◆ 綠色運具推廣。
- ◆ 道路洗掃減少揚塵。
- ◆ 工地、堆置場灑水覆蓋。

空品不良時刻應變作法

- ◆ 要求工廠降載或加強防制效率。
- ◆ 要求台電減少發電量。
- ◆ 管制高污染老舊車輛。
- ◆ 加強道路洗掃。
- ◆ 要求工地、堆置場增加灑水覆蓋頻率。
- ◆ 跑馬燈、顯示看板、廣播放送空品不良訊息。
- ◆ 市府各局處聯合應變、通報[1][2]。

2.2 Arduino IDE

Arduino是一塊基於開放原始碼發展出來的I/O介面控制板，並且具有使用類似java,C語言的開發環境，讓使用者可以快速使用Arduino語言與Flash或Processing...等軟體，作出互動作品。[2]

2.2.1 Arduino 的介紹

Arduino 是在 2005 年 1 月由米蘭互動設計學院的教授 David Cuartielles 和 MassimoBanzi 所設計出來了，原始構想是希望讓設計師及藝術家們，透過 Arduino 很快的學習電子和感測器的基本知識，快速的設計、製作作品的原型，很容易與目前設計系所學的FLASH,MAX/MSP,Virtool 等軟體整合，使得虛擬與現實的互動更加容易。互動的內容設計才是設計師的主要訴求，至於怎麼拼湊一個單晶片開發板，或是當中涉及如何構築電路之類的知識，就並非設計師需要了解的，因此非常適合不具電子背景的人使用，以設計出各種不同的互動裝置。[2]

Arduino 控制板包含了硬體與軟體兩大部分，硬體部分是一個約手掌大小的控制板(寬 70mm 高 54mm)，核心使用八位元 ATMEGA8 系列的微控制器，提供 14 個數位式輸出/入端，6 個類比式輸出/入端，支援 USB 資料傳輸，可以使用自備電源(5V~9V)或是直接使用 USB 電源，使用者可以

在數位式輸出/入端上接上不同的電子裝置，例如 LED 燈、喇叭、馬達，然後再由控制器來驅動燈的亮滅、喇叭發聲、馬達運轉。Arduino 控制板採用開放式源碼設計的概念，電路設計圖、韌體都可以在網路上下載，稍具電子知識就可以自行製作；也可以在網路上用很便宜的價錢買到[2]。

2.2.2 Arduino 的起源

Massimo Banzi 之前是義大利 Ivrea 一家高科技設計學校的老師，他的學生們經常抱怨找不到便宜好用的微處理機控制器。西元 2005 年，Massimo Banzi 跟 DavidCuartielles 討論了這個問題，DavidCuartielles 是一個西班牙籍晶片工程師，當時是這所學校的訪問學者。兩人討論之後，決定自己設計電路板，並引入了 Banzi 的學生 DavidMellis 為電路板設計開發用的語言。兩天以後，DavidMellis 就寫出了程式碼。又過了幾天，電路板就完工了。於是他們將這塊電路板命名為『Arduino』。當初 Arduino 設計的觀點，就是希望針對『不懂電腦語言的族群』，也能用 Arduino 做出很酷的東西，例如：對感測器作出回應、閃爍燈光、控制馬達...等等。隨後 Banzi、Cuartielles、Mellis 把設計圖放到了網際網路上。他們保持設計的開放源碼(OpenSource)理念，因為版權法可以監管開放原始碼軟體，卻很難用在硬體上，他們決定採用創用 CC(Creative_Commons,2013) [3] 許可。

創用 CC 是為保護開放版權行為而出現的類似 GPL 的一種許可 (license)，來自於自由軟體基金會(FreeSoftwareFoundation)的 GNU 通用公共授權條款(GNUGPL)：在創用 CC 許可下，任何人都被允許生產電路板的複製品，且還能重新設計，甚至銷售原設計的複製品。你還不需要付版稅，甚至不用取得 Arduino 團隊的許可。然而，如果你重新散佈了引用設計，你必須在其產品中註解說明原始 Arduino 團隊的貢獻。如果你調整或改動了電路板，你的最新設計必須使用相同或類似的創用 CC 許可，以保證新版本的 Arduino 電路板也會一樣的自由和開放。

唯一被保留的只有 Arduino 這個名字：『Arduino』已被註冊成了商標『Arduino®』。如果有人想用這個名字賣電路板，那他們可能必須付一點商標費用給『Arduino®』(Arduino,2013)的核心開發團隊成員。『Arduino®』的核心開發團隊成員包括：MassimoBanzi，DavidCuartielles，TomIgoe，GianlucaMartino，DavidMellis 和 NicholasZambetti。(Arduino,2013)。

『Arduino®』，是一個開放原始碼的單晶片控制器，它使用了 AtmelAVR 單晶片(Atmel_Corporation,2013)，採用了基於開放原始碼的軟體硬體平台，構建於開放原始碼 SimpleI/O 介面版，並且具有使用類似 Java，C 語言的 Processing/Wiring 開發環境，對資訊科技的發展是一個非常大的貢獻。

您可以快速使用 Arduino 語言作出互動作品，Arduino 可以使用開發完成的電子元件：例如 Switch、感測器、其他控制器件、LED、步進馬達、其他輸出裝置...等。Arduino 開發 IDE 介面基於開放原始碼，可以讓您免費下載使用，開發出更多令人驚豔的互動作品[3]。

2.2.3 Arduino IDE 介紹

IDE 的英文全名是 Integrated Development Environment，中文譯為：整合開發環境[3]。

Arduino IDE 的介面，我們使用的是 1.6.5 的版本。你可以在這裡進行程式的編寫，然後將寫好的程式上傳（燒錄）到 Arduino 板子上，觀察程式執行的成果[3][8]。

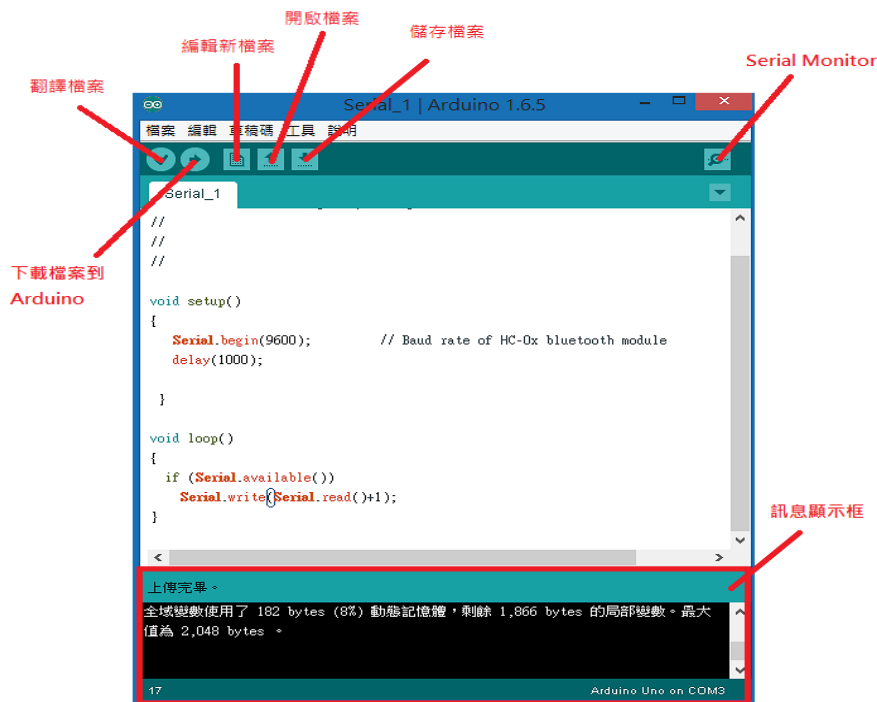


圖2.2 Arduino IDE 的介面說明

2.2.4 Arduino UNO R3 開發板

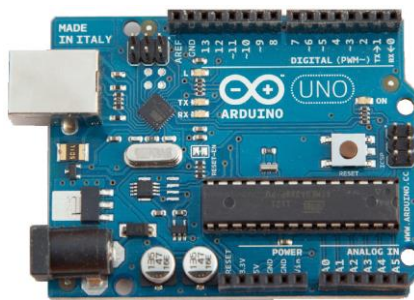


圖2.3 Arduino UNO R3 開發板圖片

Arduino，是一個開放原始碼的單晶片微控制器，它使用了 AtmelAVR 單晶片，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入介面板，並且具有使用類似 Java、C 語言的 Processing/Wiring 開發環境[3]。

我們將空氣品質偵測儀器連接到開發版上，並將程式碼透過 Arduino 燒入置此開發版中，成功上傳後便可執行程式獲得資訊。

2.3 藍牙模組 HC-06

藍牙模組 HC-06，都採用英國劍橋的 CSR (Cambridge Silicon Radio) 公司 BC417143 晶片，支援藍牙 2.1+EDR 規範，只是晶片內部的韌體不同。CSR 是全球市佔率最高的藍牙通訊晶片廠，基本的藍牙序列通訊模組沒有引出接腳，而是在印刷電路板的四周留下郵票般的齒孔，方便焊接。如下圖 2.4。[4]

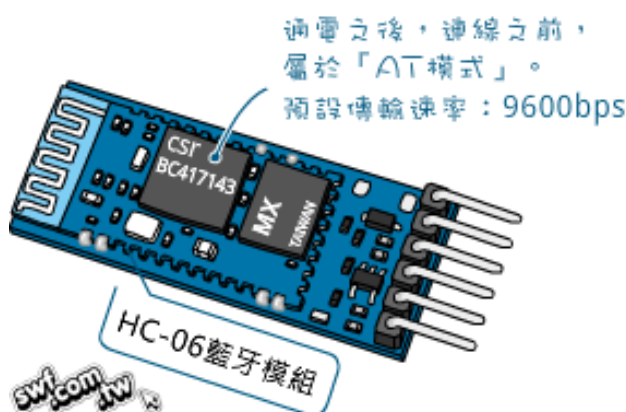


圖2.4 藍牙模組 HC-06 圖片

2.4 Android 手機

Android 常見的非官方中文名稱為安卓，是一個基於Linux 核心的開放原始碼行動作業系統，由Google成立的Open Handset Alliance (OHA，開放手機聯盟) 持續領導與開發，主要設計用於觸控螢幕行動裝置如智慧型手機和平板電腦與其他可攜式裝置。

Android 系統最初由 Android 之父 Andy Rubin 等人開發製作，於 2005 年 7 月 11 日被美國科技企業 Google 收購，2007 年 11 月 5 日，在 Google 的領導下，成立開放手持裝置聯盟 (Open Handset Alliance)。

隨後，Google 以 Apache 免費開放原始碼許可證的授權方式，發布了 Android 的原始碼，任何廠商都不須經過 Google 和開放手持裝置聯盟的授權隨意使用 Android 作業系統；但是製造商不能在未授權下在產品上使用 Google 的標誌和應用程式，例如 Google Play 等。[5]

2.5 MIT AppInventor 2

Appinventor2 是 Google 所提供 Android 的開發環境，目前版本為第二版，只要有 Google 帳戶皆可免費使用。Appinventor2 使用拼塊程式來撰寫，以圖像方式來呈現讓程式初學者能夠更快上手，同時了解程式設計的脈絡及邏輯架構。[6][9]

(一)AppInventor2的由來與特點

AppInventor 最初是由 Google 設計的應用軟體，現在由 MIT 行動學習中心所維護及運營。它使用圖形化介面可以讓任何熟悉或不熟悉程序設計的人來創造 Android 程式。

2013 年 12 月 MIT 行動學習中心推出了 AppInventor2。AppInventor2

有新增許多功能，例如：支援中文介面、整合版面設計和程式設計為單個頁面、圖形化介面、完全雲端設計和積木式建構。

AppInventor 簡單上手的特性也很適合用在中學或者是大學中的一些基礎程式設計課程，讓學生能夠快速的上手、了解一些基本應用程式的原理卻又不需要接觸到困難艱深的部份。事實上，AppInventor 便是架構在 Scartch 這個由 MITMediaLab 所研發的教學用、視覺化的程式語言架構。

(二)AppInventor2的使用

AppInventor2 的使用只需要登入 Google 帳號後進入 AppInventor2 的網頁，建立專案後開啟專案即可開始設計。如需要使用模擬器，再另外安裝即可。[6]

(三)AppInventor2的優點

- 1.適合無Java基礎的初學朋友。
- 2.操作概念很類似Scratch。
- 3.全雲端，所有作業都在瀏覽器完成。
- 4.免安裝，只要有Google帳號，使用瀏覽器開發。
- 5.支援樂高機器人。

(四)AppInventor2的缺點

- 1.無法動態產生(NEW)物件功能。
- 2.效能較差，檔案較大。
- 3.開發功能及美化受限制。
- 4.目前無中文版介面[6]。

(五)AppInventor與AppInventor2版本功能比較

表 2.3 版本功能比較表

	AppInventor2	AppInventor
設計介面	Browser	Browser
程式撰寫	Browser	Javawebstart(jnlp)
Blocks改善	MoreSmallSpace	MoreBigSpace
原始碼檔案	.AIA	.ZIP

2.6 DHT22 溫溼度感測器模組

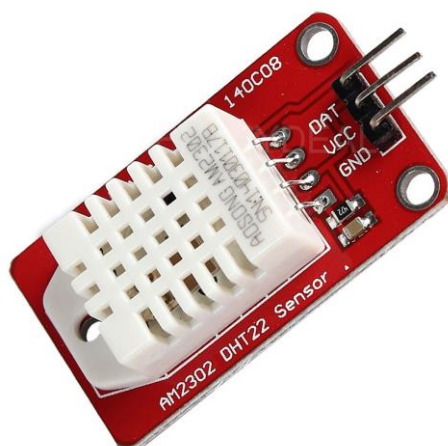


圖2.5 DHT22 溫溼度感測器模組圖片

DHT22溫濕度感測器是一款含有已校準數位信號輸出的溫濕度複合感測器。它應用專用的數位模組採集技術和溫濕度傳感技術，確保產品具有極高的可靠性與卓越的長期穩定性。感測器包括一個電容式感濕元件和一個NTC 測溫元件，並與一個高性能8位單片機相連接。因此該產品具有品質卓越、超快回應、抗干擾能力強、性價比極高等優點。[2]

2.7 SHARP PM10空氣品質感測器



圖2.6 SHARP PM10空氣品質感測器圖片

光學空氣品質感測器，設計用來感測空氣中的塵埃粒子，其內部對角安裝紅外線發光二極體和光電晶體，使其能夠探測空氣中塵埃反射光，即使非常細小的如煙草煙霧顆粒也能夠被檢測到，通常應用在空氣淨化系統中。[2]

第參章 實作方法

此章節說明本專題的目的與實作架構，並介紹操作步驟與實作的流程。

3.1 訂立問題與實作目的

Arduino 的應用在台灣顯然比在其他國家來得少，所以台灣只有少數廠商會製作這個產品，不過外國已經把這項技術研發得差不多了，Arduino 搭配各種偵測空氣的儀器可以得知各種數值，加上藍牙的功能後會有更多用處。專題實測中不足之處是對程式需有一定認知，硬體與線路組裝與空氣品質的偵測。所以本次專題的過程中把系上所學的知識用在上面，而這次所要製作的是空氣探測器，它便是把手機 App 與系統整合所學到的知識運用在一起所做出來的成品，用空氣感測器連接 Arduino 控制板，加上藍牙控制模組來完成整個空氣感測器，藉著無線操作的方式來感測空氣中 PM2.5 的濃度數據，並達到知識運用與實際操作的成果。

3.2 實作架構

(1) 規劃:

參考文獻收集及分析、Arduino 板子認識及選用、相關開發元件的認識及安裝、藍芽傳輸器的認識及安裝、App Inventor 2 的使用、Arduino IDE 安裝與程式設計、系統測試及建置。

題目:Arduino 的應用在台灣顯然還太少，所以我們決定利用 Arduino 來製作此專題。

選定工具:因為 Arduino 是一個新穎的工具就算是沒有電子背景的人都能夠輕鬆上手，所以我們選了 Arduino 以及相關元件為主要工具。

可行性評估:透過教師教導 Arduino 相關元件、App Inventor 2、Arduino IDE，並使用 Arduino 這個核心來做各種設計以及設計 App 來製作本專題。

(2) 分析:

收集參考文獻:在網路上尋找 Arduino 程式的寫法以及 App 的相關寫法，研究 Arduino 開發版的配置與藍牙儀器的使用。

需求分析:使用者可以利用 App 與 Arduino 連結以感測空氣品質的變化。

(3) 設計:

介面設計我們直接以 App Inventor 2 上的功能設計並連接至 Arduino 開發版上的藍牙儀器傳輸資料回手機。

(4) 建置:

撰寫程式以 App Inventor 2 為工具開發版面以及功能，App 的版面有空氣品質、連結藍芽、發送訊息，並用 Arduino IDE 撰寫 Arduino 裡的程式，抓取資料後回傳置手機上做管理。

3.3 空氣品質系統架構

系統功能分別為 App 設定暨蒐集子系統、Arduino 感知元件控制系統，利用程式透過 Arduino 做到測量空氣品質及藍牙控制之功能。

3.4 系統功能解釋

1. App 設定暨蒐集子系統方面，我們透過了 Arduino 上的藍牙板與 Arduino 做傳輸，我們可以利用 App 來傳輸空氣中 PM 10 的濃度，程式方面我們使用了 App Inventor 2 及 Arduino IDE，設計 App 版面及傳送訊息設定、藍牙設定等功能。

2. Arduino 感知元件控制子系統方面，我們以空氣感測器 PM 10 的濃度，並以藍牙板傳輸數據到手機 App 顯示濃度值，當測量到濃度值過高的時候就會啟動警報器並且開啟發送訊息功能。

3.5 實作流程

本實作先確立了研究動機與背景，並在網路上蒐集資料與問卷調查，了解民眾的需求與意見後，經過討論後再設計出符合大眾需求的品質偵測器。

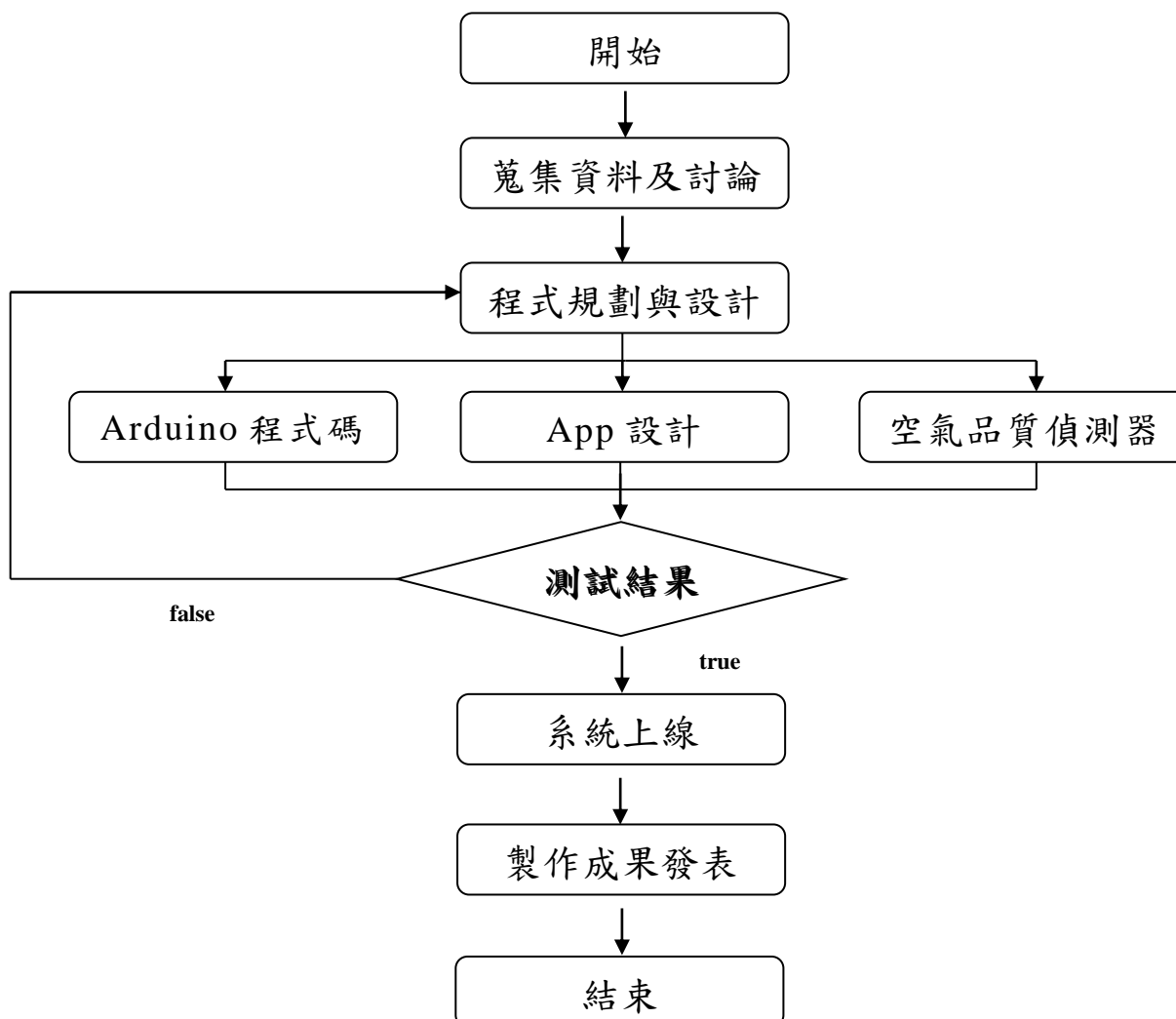


圖3.1 實作流程圖

3.6 儀器偵測步驟與結果

將儀器都設定完成，便可透過簡單的步驟去執行 Arduino，以下是我們的步驟解說和在嶺東科大測到的數值。

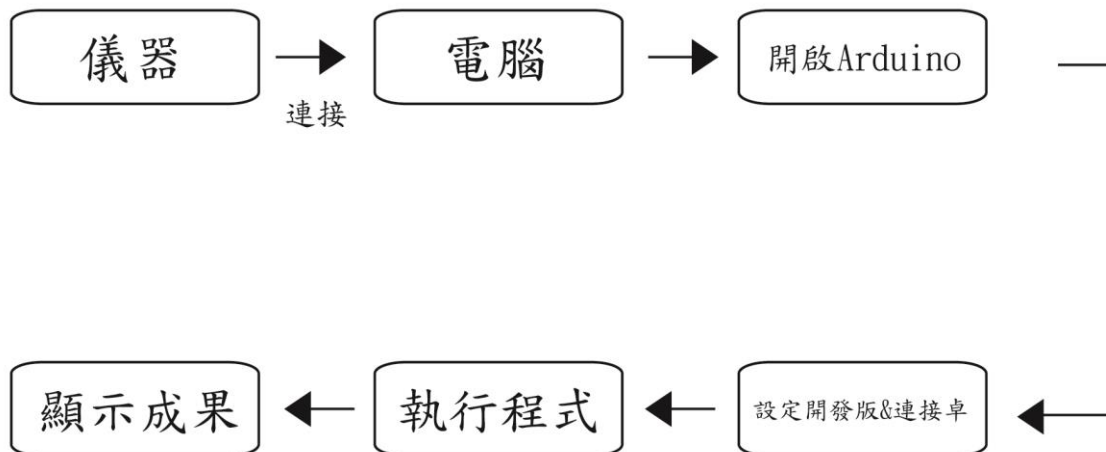


圖3.2 儀器操作示意圖

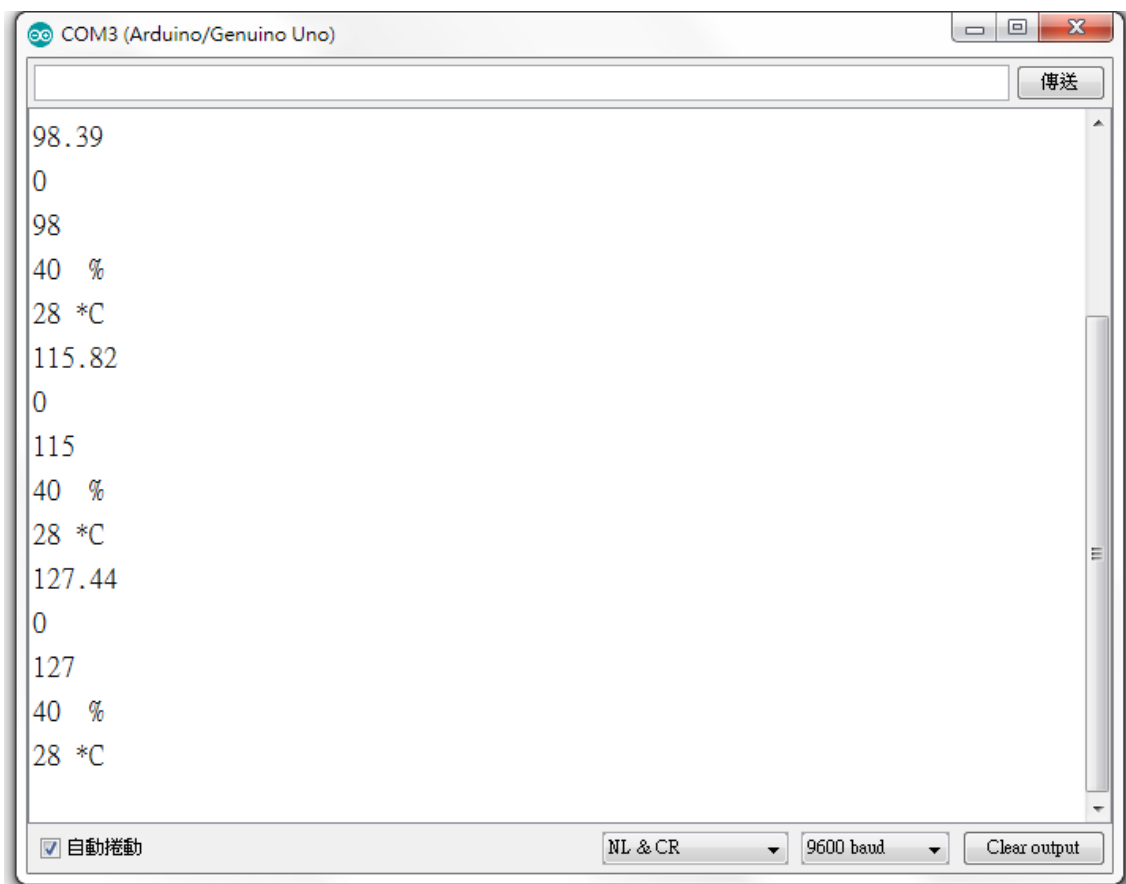


圖3.3 Arduino上顯示之數據

第肆章 實作成果

本章節將介紹我們儀器的使用和說明匯入儀器中的程式碼，並示範 App 的操作，逐一說明設計的想法與操作方法。

4.1 實體與電路圖說明

以下為我們儀器的展示圖，使用到 UNO R3 開發板、懸浮微粒感測器、DHT22 溫溼度感測模組、HC-06 藍芽模組和麵包版。

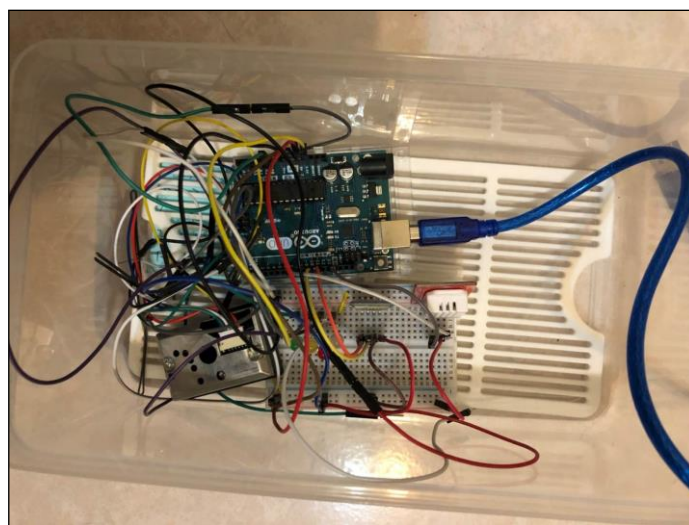


圖4.1 實體儀器圖

以下圖片為儀器與 UNO R3 開發板利用麵包板連接的電路圖，上方為 UNO R3 開發版，下方為 HC-06 藍芽模組，左方為夏普空氣品質感測器、右方為 DHT22 溫溼度感測器，線路配置方面，紅色線為輸出的 5V 電壓，棕色為接地線 GND，藍、橘、黃、白為儀器輸出與輸入端的接線。[8]

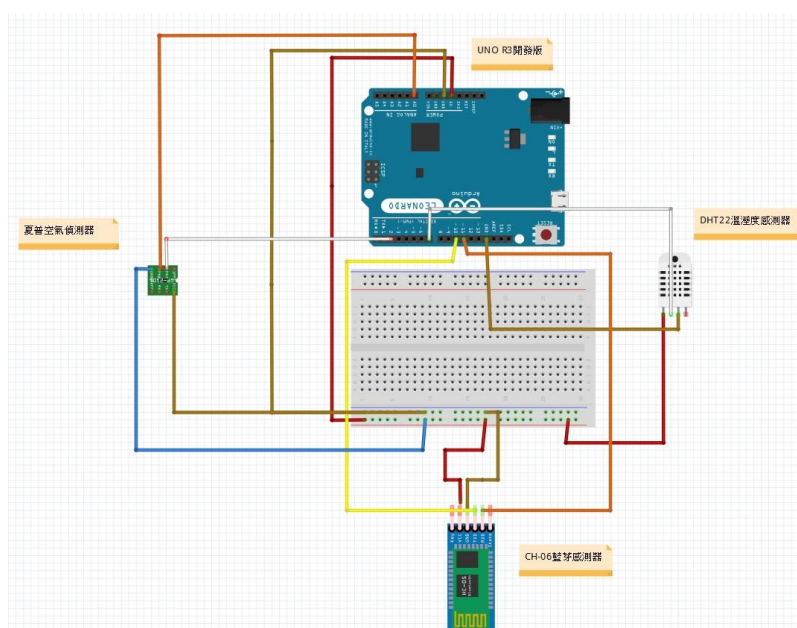


圖4.2 儀器電路設置圖

4.2 Arduino UNO R3 程式碼

以下介紹我們匯入UNO R3開發版內的程式編碼與說明設定內容。

圖 4.3 利用#include 宣告使用到的溫溼度感測器、夏普空氣品質感測器和藍芽的程式，並匯入 DHT 與藍芽的程式庫與設置藍芽傳輸的 RX/TX 腳位，在下方利用 int 宣告時間與 LED 的 pin 腳，並可以完成執行程式的基本設定。

```
#include "DHT.h"
#include <SoftwareSerial.h>
#include <Wire.h>
#define DHTPIN 7
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial BTSerial(10,11); // Arduino RX/TX

|
int sleepTime = 9680;
byte serialA;
int measurePin = A1; //Connect dust sensor to Arduino A1 pin
int ledPower = 2;
int samplingTime = 280;
int deltaTime = 40;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
```

圖4.3 程式宣告與匯入程式圖

圖 4.4 設定好以上宣告語法後，進入 void setup 的設定，將 DHT22 與 HC-06 的傳輸頻率為 9600，若程式執行成功將在序列卓監控視窗顯示 DHTxx text!(圖 4.8 說明)，若傳輸頻率設置錯誤則無法順利傳輸給 App 上作分析。

```
void setup() {
    Serial.begin(9600);
    BTSerial.begin(9600);
    Serial.println("DHTxx test!");

    dht.begin();
}
```

圖4.4 程式執行頻率設置圖

圖 4.5 在 void loop 設定中，設定夏普空氣品質的輸出電流，並計算夏普空氣品質回傳開發版的數值，將其設定為整數 i，將 DHT22 回傳的溼度設為 x、溫度設為 y，以利傳送封包時的數值設定。

```
void loop() {
  digitalWrite(10,HIGH);
  digitalWrite(ledPower,LOW); // power on the LED
  delayMicroseconds(samplingTime);

  voMeasured = analogRead(measurePin); // read the dust value

  delayMicroseconds(deltaTime);
  digitalWrite(ledPower,HIGH); // turn the LED off
  delayMicroseconds(sleepTime);
  // 0 - 5V mapped to 0 - 1023 integer values
  // recover voltage
  calcVoltage = voMeasured * (5.0 / 1024);
  dustDensity = 0.17 * calcVoltage - 0.1;
  if (dustDensity<=0)
  {
    dustDensity=0;
  }
  Serial.print(dustDensity * 1000);
  Serial.println(" ");
  int i = dustDensity * 1000;

  int x = dht.readHumidity();

  int y = dht.readTemperature();
```

圖4.5 程式void loop設置圖

圖 4.6 說明將偵測到的數值打包成 Data 封包，並將空氣品質數值 i 除以 256 並取小數位，利用藍芽傳送至手機的 App 上，並利用 Serial.println 將數值顯示在序列卓監控視窗。

```
byte cmd[20];
int insize;
serialA=BTSerial.read();
Data[0]='a';
Data[1]=i/256;
Data[2]='b';
Data[3]=i%256;
Data[4]='c';
Data[5]= x;
Data[6]='d';
Data[7]= y;
Data[8]='e';

Serial.print( Data[1]);
Serial.println(" ");
Serial.print( Data[3]);
Serial.println(" ");
Serial.print( Data[5]);
Serial.print(" ");
Serial.println(" %\t");
Serial.print( Data[7]);
Serial.println(" *C ");
```

圖4.6 傳輸封包設定圖

最後設置 if 迴圈，傳送一個訊息 49 給 App 若手機有收到訊息，就在監控視窗上顯示 Send data，並將封包傳送至 App 上分析，在設定延遲 0.5 秒來讓程式運作。

```
if (serialA = 49) {  
  Serial.println("Send data"); // 如果有在視窗上看到這個訊息表示Arduino有收到  
  for (int j = 0; j <= 8; j++)  
    BTSerial.write(Data[j]);  
  
  serialA = 0;  
}  
delay(500);  
}
```

圖4.7 藍芽傳送迴圈設置圖

圖4.8為Arduino IDE上顯示之數據，將開發版上的數值分析完後傳送至電腦上的監控視窗，即可確認儀器偵測數值的執行，並利用封包傳送至手機做分析。

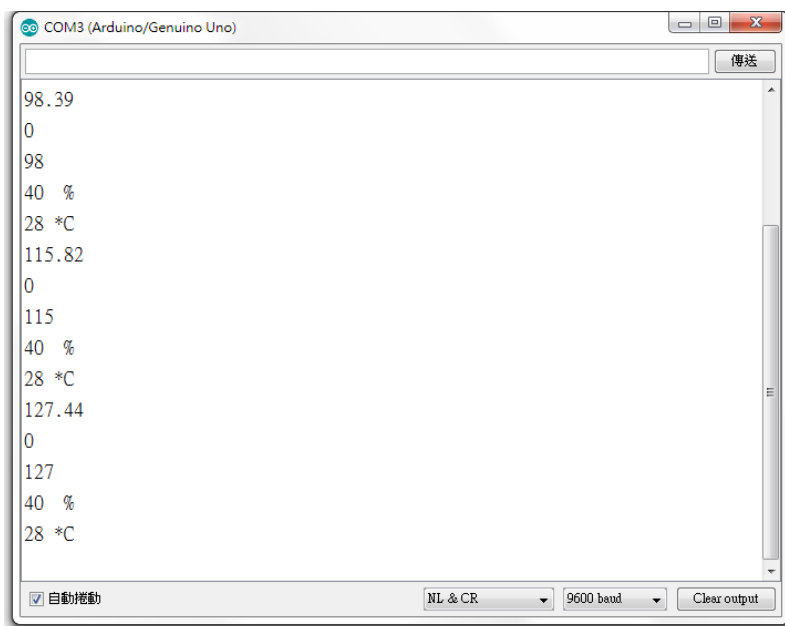


圖4.8 Arduino上顯示之數據

4.3 硬體測試校準

以下是我們一個月統計以來統計的數據，透過儀器偵測的數值與台中市空氣品質文山偵測站得數值作比對結果，並與AQI標準指對照校準。[7]

表4.1是由行政院環保署設置的健康指標指數，我們參考這個指標測量該地區的PM10，實際使用我們的儀器去測試和比較之間的誤差，並能使對照以下數據，讓我們能夠清楚的得知空氣品質PM10的狀況，以警惕使用者做好防護措施。由表4.2可看到我們偵測到的數值跟台中市監測的數值是差不多的，這些細微差距可能是偵測的範圍跟位置的關係，或是儀器的差距所造成，但是差距值也差距在正負5上下。[7]

4.4 AppInventor2 程式碼

設定使用者點選請單選擇器後，進入清單選擇器顯示藍芽的地址與名稱，在此專題中我們選用儀器藍芽名為 1234。



圖4.10 清單選擇器設置

當 App 初始化後，設定清單選擇器可以點選，按鈕與計時器為不可使用，按鈕一被點選後，切斷藍芽與儀器的連結，並將標籤 2、4、6 數值設定為空值。



圖4.11 按鈕與App初始化設定

當進入清單選擇器並選擇要連結之藍芽後，停用清單選擇器按鈕，並啟用按鈕和計時器，若藍芽執行錯誤，關閉錯誤訊息螢幕。

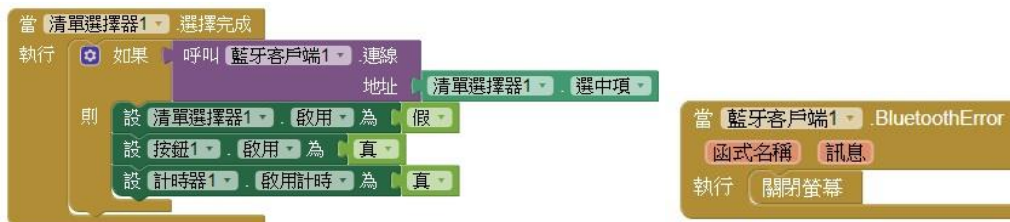


圖4.12 選擇藍芽與執行錯誤

設定發送簡訊之按鈕，將傳送出去簡訊文字設定為”需要注意空氣品質!!”，使用者想傳送簡訊號碼將其輸入在文在輸入盒內，點擊按鈕 2 將可以傳送給想警告目標，傳送成功後顯示發送完畢訊息，並將輸入框內的號碼清初恢復為空值。



圖4.13 傳送簡訊按鈕程式

我們設置一個計時器，當 App 連接到儀器時，呼叫手機上藍芽傳送 49 數字給儀器上，確認與儀器正確連接後，將標籤 1 到 9 設定為接收到的封包，並將分析完的封包數據設為變數去計算與顯示，我們將空氣品質傳送數據暫存在變數 TEXT 裡面並將標準值定在 256，若空氣品質的數據超過標準值則將顯示”空氣汙染嚴重”，反之顯示”空氣品質優良”，並將文字顏色設為超標時變成紅色以警告使用者，當數值為超標時則變回黑色文字，在超過標準值的時候 App 也會發出警示的鈴聲來提醒使用者。[9]

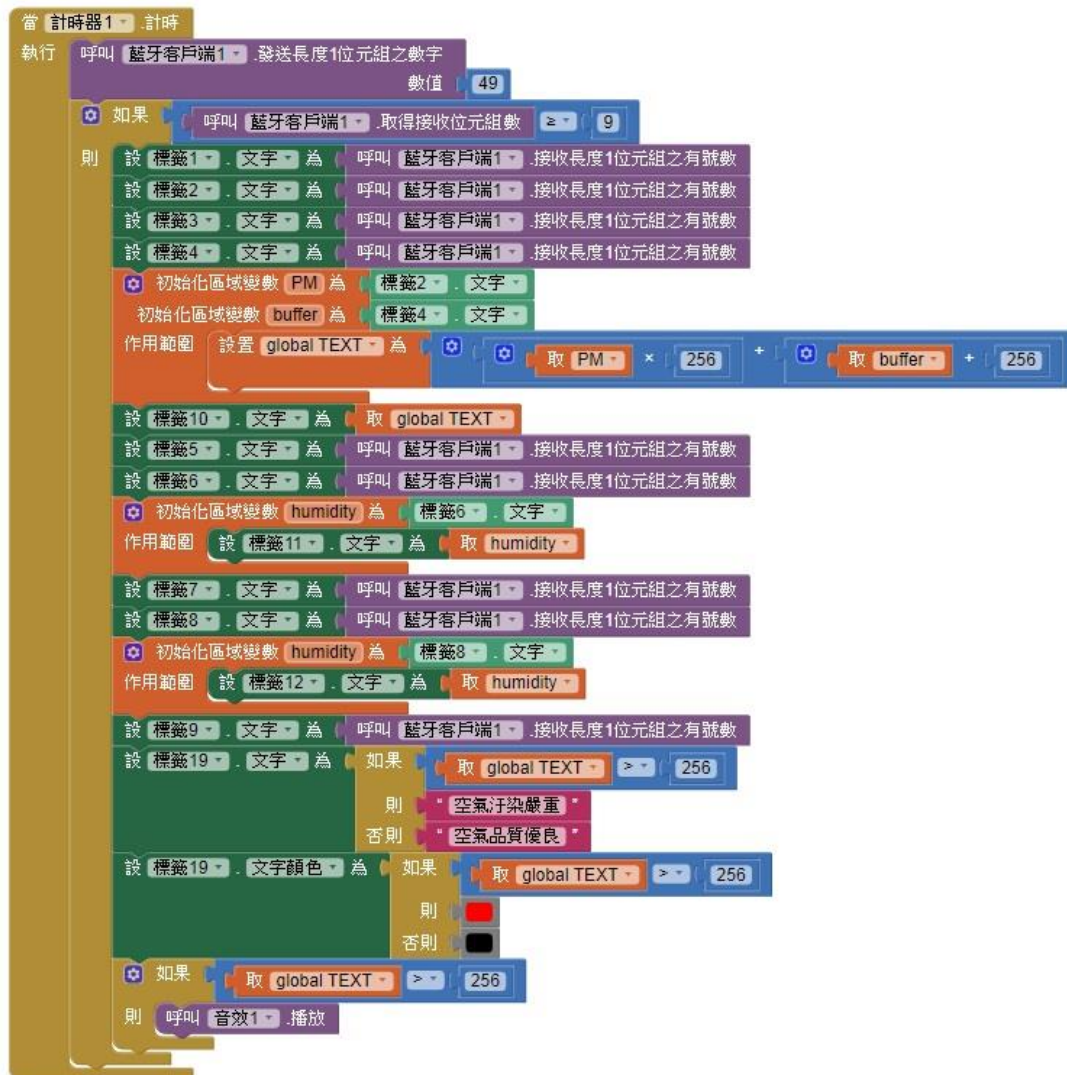


圖4.14 藍芽傳送封包解析與設置圖

圖4.15說明在線上取功能取得資料後經由App Inventor 2去抓取所需欄位14中的第2欄位的資訊並定義成變數值，並將整個步驟定義成CityList。



圖4.15 程序定義圖

圖4.16說明App取得線上json文本後，去抓取第9項便會得知PM10的數值，抓取第10項則可以取得PM2.5的數值，之後將設置好的變數資訊顯示至所需標籤上，便可以顯示於App上。



圖4.16 抓取文本設置圖

4.5 App操作介紹

本專題以簡單操作與簡潔的介面設計App並呈現出幾個重要的功能，其中有設定藍牙、顯示空氣品質數值、警報器，簡單明瞭讓使用者更好上手。

以下介紹App的架構圖與基本操作、連接儀器時的操作設定。

App操作架構：

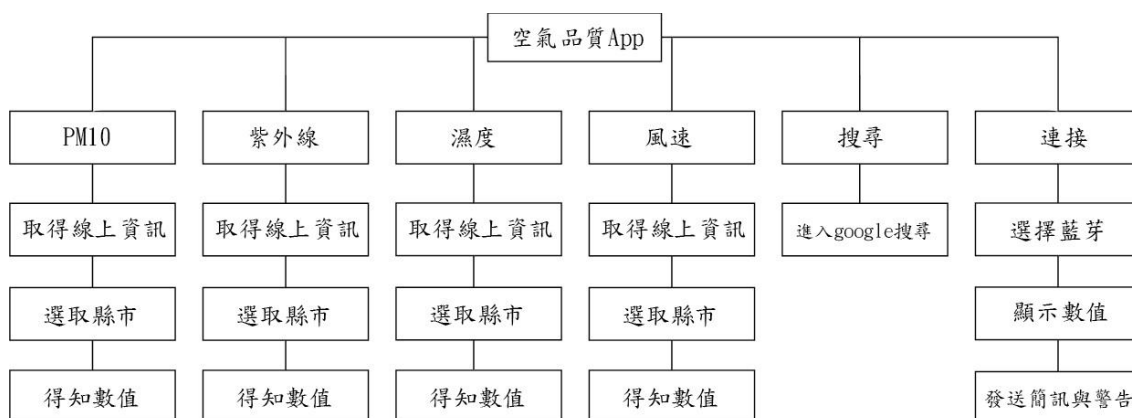


圖4.17 App操作架構圖

基本操作步驟：

1. 點選App進入選單畫面。
2. 選擇想偵測的項目。
3. 進入選單後點選取得縣市資料，並選取程式。



圖4.18 基本操作步驟示意圖

連結儀器操作步驟：

1. 點選選擇藍芽按鈕，並進入清單中選取藍芽。
2. 等待儀器連上App後，數值將會回傳至手機。
3. 輸入想傳送警告的手機號碼。

4. 發送簡訊。



圖4.19 連接儀器操作示意圖

第五章 結論與建議

本專題使用 DHT22 溫溼度模組、SHARP PM10 感測器來測量濕度和懸浮微粒指數，透過藍芽 HC-06 回傳數值至行動裝置告知使用者目前儀器所在地的環境品質。

App 提供使用者線上抓取資料的功能，使用者可以直接抓取線上提供的數據來知道 PM10、PM2.5、紫外線、風速、風向、溼度數值，也可透過實體儀器來偵測環境品質，並在數值超標時可以警告使用者與發送警告訊息。

我們統計實作儀器與政府文山觀測站的偵測數值，並實際校準後，發現兩者數值差距極小，與政府偵測出來的數值有一致性，代表我們的儀器功能是擁有可信度的。

本專題在硬體與軟體的部分，經過校準與測試完成後，仍有不足的方面需要改進，以下是我們統整出更完善的功能建議：

- 1.數據資料庫化:系統資料庫雖然因為時間的不足導致無法建置完成，但希望往後有機會也能將此專題的資料往這方面去建置，讓使用者可以得知長期以來的空氣數據，更可以去加以防範空氣汙染。
- 2.使用者互動即時化:App與使用者互動上，我們的設計也是朝向使用者主動去觸發App上的操作，才執行所需程式的運作，在老師的建議上，希望我們能往即時訊息去製作，不過在這方面的問題仍有許多需要去研究與解決，所以我們專注於使用者和App的互動精進，讓使用者能在使用上得知更多數據。

參考文獻

[1] 高雄市環保局空噪科，相關網址如下：

<http://air.ksepb.gov.tw/Article/Detail/3>。

[2] 維基百科，相關網址如下：

<https://zh.wikipedia.org/wiki>。

[3] 台灣物聯科技，相關網址如下：

<https://www.taiwaniot.com.tw/>。

[4] 藍牙模組(HC05、HC06)常見的指令使用教學，相關網址如下：

<https://reurl.cc/vjxj>。

[5] Android - 簡介發展歷史 與系統設計，相關網址如下：

<http://mropengate.blogspot.tw/2015/12/android-ch1-android.html>。

[6] AppInventor2，相關網址如下：

<http://appinventor.mit.edu/explore/>。

[7] 行政院環境保護署，相關網址如下：

<https://taqm.epa.gov.tw/taqm/tw/b0201.aspx>。

[8] 超圖解 Arduino 互動設計入門:

作者：趙英傑，出版社:旗標，出版日期：2016/12/22。

[9] MIT App Inventor 2 易學易用 開發 Android 應用程式:

作者：王安邦，出版社:上奇資訊，出版日期：2014/02/26。