

賣場之智慧自走車監控介面

專題學生： 顧恆瑞

黃馨輝

大 同 大 學

資 訊 工 程 學 系

專題報告

中 華 民 國 110年 6月

指導教授：黃國軒 教授

|  |  |  |
| --- | --- | --- |
|  | 賣場智慧自走車監控介面 |  |
|  | 大同資工系專題報告 |  |
|  | 顧恆瑞  黃馨輝 |  |

大同大學

資訊工程學系

專題報告

賣場之智慧自走車監控介面

顧恆瑞

黃馨輝

專題考試委員

指導教授

**摘要**

經考試合格特此證明

中 華 民 國 110 年 6 月 6 日

**摘要**

今年5月份，台灣疫情爆發，引發無數人進入賣場搶購物資，恐造成群聚感染，為減低群聚的風險，本專題打算設計一款能監測空氣品質和具備導航功能的自走車，目前已知新冠肺炎(covid-19)可經由飛沫傳染，且病毒存活期極長，因此本專題將透過各項感測器監測溫溼度、PM2.5與二氧化碳的數值，並應用自走車導航來進行賣場內部的巡邏，以判斷該區域是否是高風險該區域。

肺炎空氣傳播途徑主要為人體吸入空氣中含有病原的懸浮粒子，因此本專題透過感測二氧化碳濃度、懸浮微粒與溫濕度等數值，作為判斷空氣品質是否合格的標準，並建立一空氣品質監測介面，以隨時監控賣場各區域空氣品質，若是即時監測出空氣品質不合標準，監控介面將會亮起警示燈，以便可以及時的疏散人群，以期降低群聚感染的風險。

並實作出自走車控制介面，使用了反射地標的方法以建立室內地圖，除卻上述與空氣品質監測做結合的功能外，還加入了line bot message api，可清楚的回答使用者各商品的存貨量，並且加入了導航功能，規劃出點到點的最短路徑，以避免使用者在賣場中行經了許多不必要的路徑，以降低感染風險。

**第1章  
概述**

**1-1 專題動機**

自疫情爆發以來，只要公布了任何可能封城的消息，許多人都會因預期心理，紛紛湧入賣場搶購物資，從而導致了群聚感染。為能夠有效降低人們在賣場群聚感染的風險。本專題將設計一款導航智慧監控自走車，以隨時監測賣場的空氣品質，以便在高感染風險時，能及時疏散群眾，並加入導航功能，能為使用者規劃到達目標商品的最短路徑，透過減少不必要的路徑，以降低群眾在賣場的逗留時間。

**1-2 專題目標**

1.設計一空氣品質監測系統，能反應及時的空氣品質數據，另透過簡單分析，用以判斷該區域空氣品質是否合格，並實作出一介面，透過MQTT將個數據傳至網頁上，並將其數據圖形化，介面上包含了各項空氣品質數據和空氣品質是否合格的警示燈。

2.透過反射地標的方式建構室內地圖，並為自走車建立操作頁面，並利用Line bot message api來實現與使用者互動的情境，以達到路徑規劃、商品存貨量查詢等功能。

**1-3 所遇問題及其解決方法**

**1-3.1 網頁相容性**

要實作其介面，第一個想到的自然是網頁。基於自走車程式碼為Python為主體，所以網頁也用Python其下的Flask架構建設。

**1-3.2 簡易地圖資料處理**

一樣由於本體為Python，所以用其numpy套件處理相關資料。比較麻煩的部分為地圖，地圖分為程式內部資料和外部顯示，內部資料一樣用numpy處理;外部顯示則用Image合成一張簡單的圖片負責顯示。

**1-3.3 簡易聊天機器人**

要求的介面還有一項為使用聊天機器人，本專題選擇使用LINEBOT環境架設聊天機器人，基於聊天機器人主要目標為控制自走車，所以聊天機器人主要程式碼放置於自走車上面。

**1-3.4** 數據傳輸的替代方案

本專題原本是透過使用聯發科專門為linkit 7697 所開發接收資料的網頁MediaTek Cloud Service (MCS)，來進行接收並顯示感測器數據，然而在2021年1月份MCS終止了服務，因此將改為使用mqtt來進行傳送資料，並傳送至adafruit io來顯示感測器資料。

**1-4 專題進度表**

表 1.1 專題時間進度表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 2021年  1月 | 2021年  2月 | 2021年  3月 | 2021年  4月 | 2021年  5月 | 2021年  6月 |
| 選定題目 |  |  |  |  |  |  |
| 專題實作 |  |  |  |  |  |  |
| 專題結束 |  |  |  |  |  |  |

**第2章  
相關技術介紹**

**2-1 Flask**

要開始用 Python 寫網站的話，絕大多數的的第一個選擇大概會是 Flask。Flask 這個套件提供了不少架設網站需要的基本工具，包括路由(Routes)、網頁模板(templates)、權限(authorization)等等的，從架設網站中最簡單的元素到最複雜的應用，Flask 和其衍生而來的套件幾乎都能實現。

**2-1.1 Flask程式基礎**

用 Flask 架設網站時，一定會從類似下面這段程式碼開始：

圖 2.1 Flask 基礎程式碼

‧第一行 : from flask import Flask

因為要用到Flask這個套件，所以一開始就要進行宣告。

‧第二行 : app = Flask(\_\_name\_\_)

初始化Flask物件，並貼上app標籤，之後就可以物件裡的各種功能。

‧第三行 : @app.route(“/”)

創造出主網域底下的“/”這個網址，並定義了請求(request)該網址時 可用的手段。

‧第四行 : def home():

定義一個函數，只要使用者請求相應網址，就執行該函數。

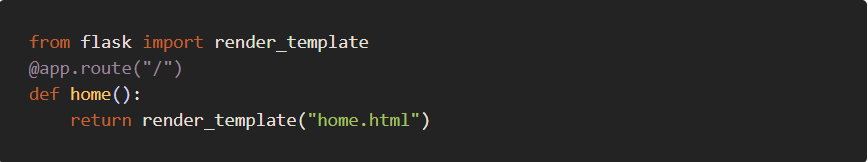
‧第五行 : return

函數內容，意指使用者請求該網址之後，根據函數運算之後回傳給使用 者的資料，也就是網頁的資料。

‧第六行 : app.run()

Flask執行。

以上為Flask最基礎的程式介紹，然而提到網頁還有更為基礎的就是HTML，Flask理所當然有提供方法套入HTML模板之中，如下所示 :

圖 2.2 Flask + HTML 程式碼範例

跟圖2.1不一樣的地方是它呼叫了render\_template()函式，並且放在了return的後面，這樣Flask的函式本體依然會在背景執行，最後回傳給HTML，這樣使用者看到網頁將會是HTML所顯示出來的內容。

**2-2 Line Bot** **Messaging API**

基於LINE為最廣泛使用的社交軟體之一，而且LINE本身也有提供介面給官方帳號建立聊天機器人，所以本專題使用Line Bot來建立簡易聊天機器人介面，且所使用的介面其名稱為Messaging API，所以特別拉出來介紹一下。

**2-2.1 Messaging API系統架構**

Messaging API 讓 data 可於 bot server 及 LINE Platform 之間傳遞，於 HTTPS 上傳送的 Request 將為 JSON 格式。

1.用戶發送訊息至 LINE 官方帳號。

2.LINE Platform 將一個 webhook 事件傳送至 bot server 的 webhook URL。

3.Bot server 將依據 webhook event，透過 LINE Platform 回應用戶。

圖 2.3 Messaging API 系統架構 [2]

**2-2.2 Line Messaging ADI SDK for Python**

Python有專門寫Line Bot的套件，所以就直接套用了。

圖 2.4 Line Bot 基礎程式碼 [3]

主要修改的位置在handle\_message的函式裡面，event.message.text為客戶端輸入內容，對其進行字串處理並搭配reply\_message和push\_message，即可針對特定內容做出特定回應。

**2-3** **室內定位**

本計畫的場域是賣場內，所以在定位方面我們選擇使用紅外線定位系統與反射地標確認機器人的位置與方位。此定位方法分為準備階段和運行階段。

1. 準備階段

準備階段的目的是使用反射地標(Landmark)建立需要的地圖。在新環境中，需要先確認反射地標放置的一致方位，因為在路徑規劃時，每個反射地標之間會有相對方向，為了讓機器人判斷正確方位。在貼上天花板時需要注意每兩張反射地標的距離不能太過接近，避免可能因為出現不同點的資訊，造成判斷上出現錯誤；需要以水平的方式貼上，避免點的角度會有所誤差；還有有不同方向點出現時，需要在轉彎點也貼置反射地標，讓機器人進行轉彎判斷。

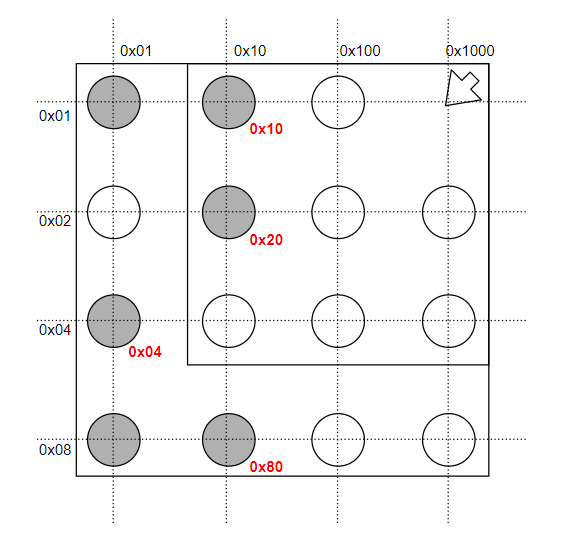
2. 運行階段

在機器人啟動時，紅外線裝置會偵測天花板上的反射地標，分析其的相關資訊(編號、角度、X軸相對位置、Y軸相對位置)並傳回給機器人，再開始從地圖中使用最短路徑演算法找出座標點與目標點的路線並告知機器人，如圖2.3。

步驟：

* + 1. StarGazer上的多組紅外線發射器向天花板的紅外線反射地標(Landmark)投射(圖2.4)。
    2. 反射地標依照本身的不同位置的座標貼紙而產生各自的ID編號，例如圖2.5的反射地標編號為180(=0x04+0x10+0x20+0x80)，還有與StarGazer的相對位置計算出角度和方位(圖2.6)。
    3. 反射地標將資訊反射回StarGazer，StarGazer接收到後轉化為影像。
    4. 進行影像分析後會得到字串，再經由處理就能知道偵測到的地標ID編號、角度、X軸相對位置、Y軸相對位置。
    5. 藉由以上資訊，機器人在移動時就能明確定位並做判斷。

圖2.5 StarGazer紅外線定位模組示意圖

圖2.6 反射地標(Landmark)

2-4 Arduino Arduino將控制晶片和燒錄功能整合在一小塊電路板上，配合麵包板設計上讓Pin腳更容易接線，且Arduino有專門開發軟件 (IDE) 可以輕鬆編寫程式並將其上傳到開發板。該開發軟體可用於任何種類的Arduino 板。且在Arduino官網或網路上都能找到大量不同功能的函式庫，讓控制各種不同用途的感測器變得更容易。

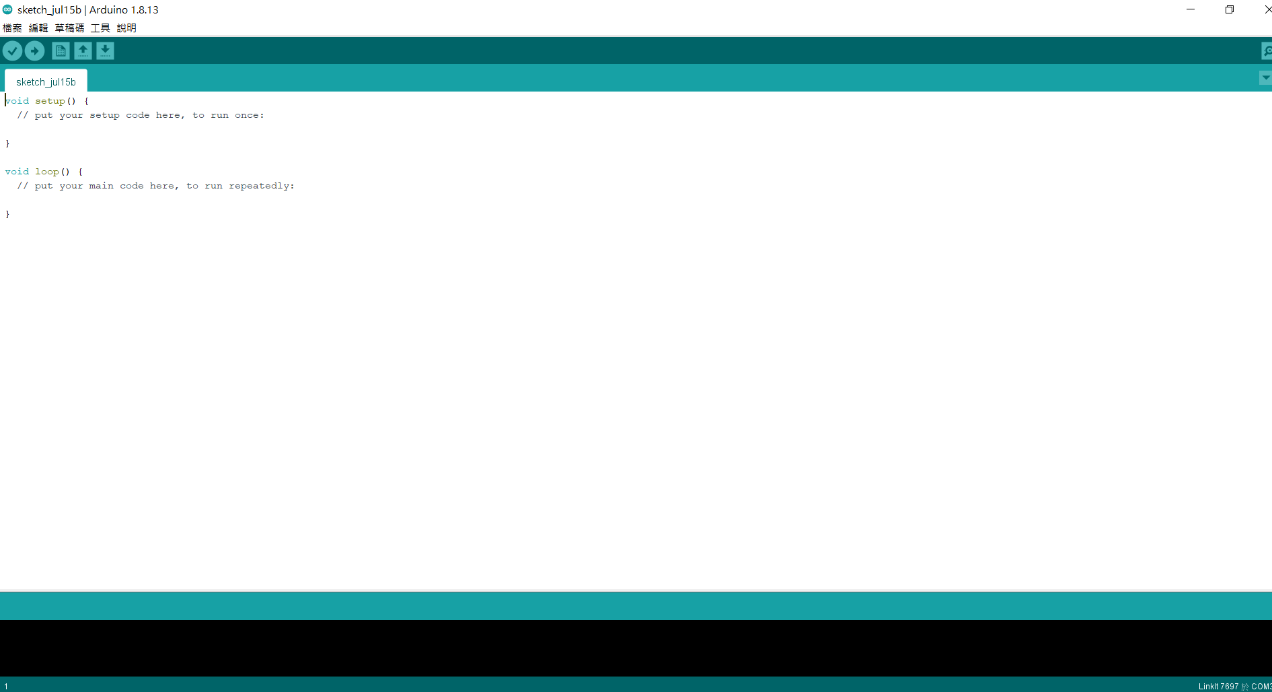


圖2.7 Arduino IDE

2-5 adafruit io

adafruit io是一個免費的IoT平台，提供數據雲端化、圖像化的服務整合，Adafruit IO可以讓使用者自由定義傳輸上去的資料內容；除了支援MQTT(S)、HTTP(S)、WebSocket外，也提供多樣化的API供使用者使用，大部分程式語言都可以透過HTTP或MQTT協定對AdafruitIO存取。

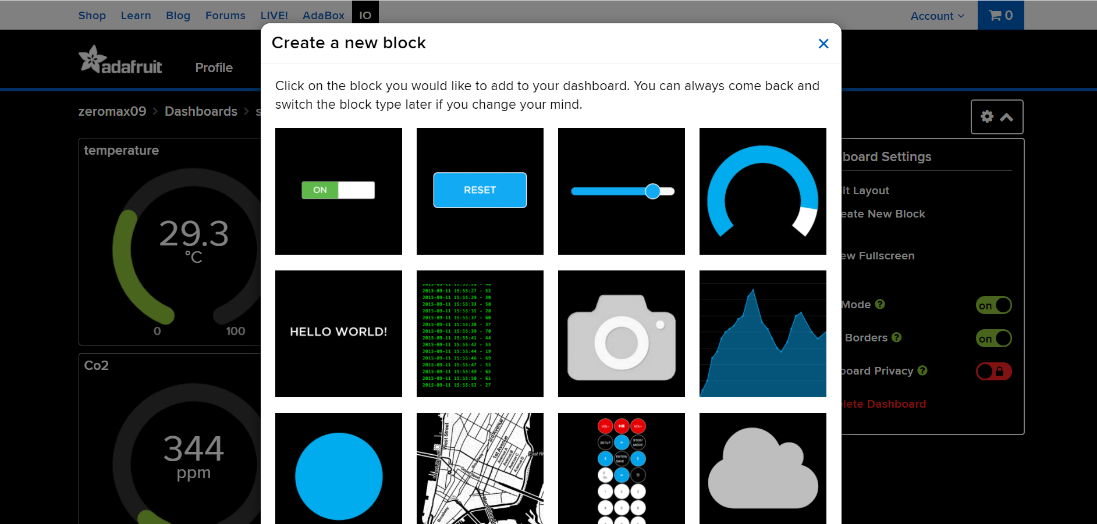


圖2.8 adafruit io

**第3章**

**研究方法  
3-1 Line Bot 建置**

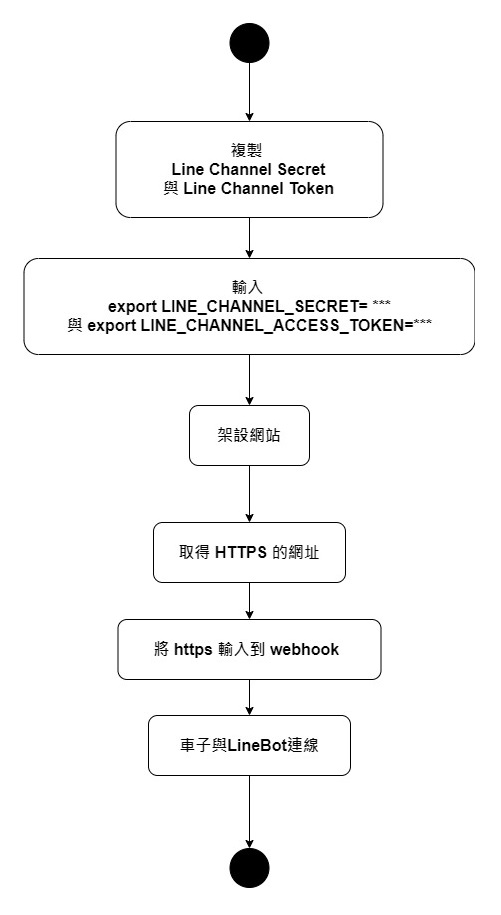
本專題選擇Line bot作為跟消費者溝通的媒介，因為Line是目前台灣使用率最高的軟體之一，在開發上也較為方便：不需再設計安裝APP、只要加入Line Bot帳號就可以直接使用其本身的開發平台套件與伺服器，除此以外藉由Line本身簡單的介面及操作可以提高顧客在使用方面的意願。

建立Line Bot服務前需要先註冊Line的帳號，完成後再到Line Developers上建立機器人並開始設定與設計。當開始使用機器人時，需要在Line Developers上取得「Line Channel Secret」和「Line Access Token」，再製作出自己的「Webhook URL」與Line Platform連線與溝通，如圖3.1。

Line Bot建置步驟：

1. 安裝ngrok，讓Line Bot能與樹莓派進行連結。
2. 申請並登入Line developers，在Console home的Provider標題旁點選Create。
3. 選擇Create a Messaging API channel ，將基本資料填入欄位後到最下方按下Create。
4. 在Basic settings中找出並複製Channel secret
5. 在Messaging API中找到Channel access token，點下issue後會出現一串訊息，也一樣複製下來。
6. 點開樹梅派的命令列，將剛剛複製的Channel secret、Channel access token 放入「export LINE\_CHANNEL\_SECRET= 自己的Channel secret」和「export LINE\_CHANNEL\_ACCESS\_TOKEN= 自己的Channel access token」並執行。
7. 執行「./ngrok http 8000 > /dev/null &」
8. 到程式的資料夾下執行「python3 geturl.py」，會得到http與https的網址，再進入Line Developers頁面點選Messaging API settings下Webhook URL的Edit鍵，將https的網址放入並點下update。
9. 回到樹莓派上執行「python3 echo\_bot.py &」。

在使用Line Bot時，我們能透過在對話框輸入關鍵字了解相關的資訊或對機器人下達命令，比如詢問「餅乾是屬於哪一個部門的商品？」或是告訴機器人從「餅乾到衛生用品」的指令等，Line Bot就會即時做出回應。

圖3.1 Line Bot建置流程圖

3-2 新冠病毒與各項氣體的關聯

3.2.1 PM2.5

哈佛大學公共衛生學院的研究指出[7]，PM2.5濃度和新冠肺炎死亡率有正相關聯，當空氣中PM2.5含量每立方米增加1微克，新冠肺炎患者死亡率也增加，最高達15%；新英格蘭醫學期刊[8]也從499個城市中的提取PM2.5數據，數據表明 PM10和 PM2.5濃度與全因死亡率之間存在顯著的正相關，且在在年平均 PM 濃度較低和年平均溫度較高的地區，相關性更強。這些文獻明確的說明了新冠病毒確實會在懸浮微粒上存活，而且還可以透過這種空氣污染的傳播途徑，讓更多的人遭受感染。

3.2.2 二氧化碳

3-3 **感測器選擇**

**3.3.1 MG811**

二氧化碳感測器，用於測量二氧化碳之濃度。其具有類比輸出及數位輸出兩種訊號輸出，對二氧化碳有高靈敏度和穩定性；能調整信號放大至2V，並富有有溫度補償功能，更精確數值。此外，還有電源工作指示燈，方便確認運作狀態。其二氧化碳測試濃度範圍為0到10000ppm，如圖3.2所示。

圖 3.2 二氧化碳感測器

**3.3.2 PMS7003**

攀藤G7，全稱PMS7003，高精度雷射型粉塵感測器及懸浮微粒感測器，可連續采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。其量測數據種類更多達12種，包含： PM1.0、PM2.5、PM10濃度以及顆粒物個數等12個參數輸出，。在本專題中將使用此感測器來測量PM2.5之濃度數值。圖3.3所示。



圖 3.3 懸浮微粒感測器

**3.3.3 DHT11**

DHT11數位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。感測器包括一個電阻式感濕元件和一個NTC測溫元件，並與一個高性能8位單片機相連接，信號傳輸距離可達20米以上，使其成為各類應用甚至最為苛刻的應用場合的最佳選則，如圖3.4所示。

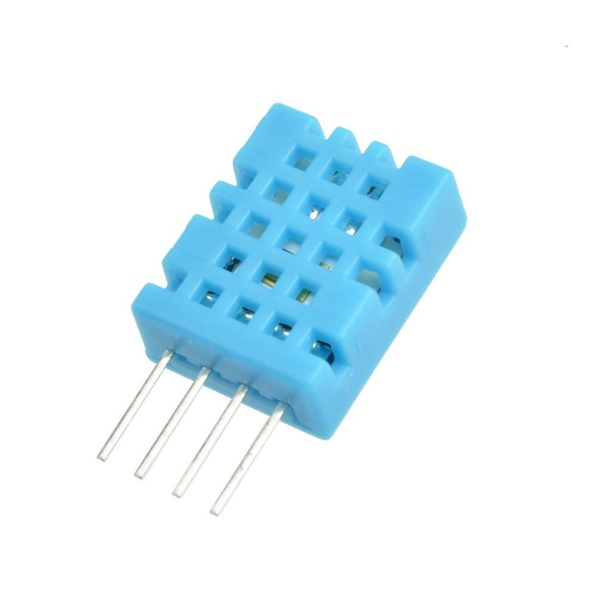


圖 3.4 溫溼度感測器

3-4 **電路配置圖**

將Linkit 7697利用杜邦線與各感測器進行連接。如圖3.5。各項感測器與Linkit 7697之PIN腳連接如表3.1、表3.2、表3.3所示。

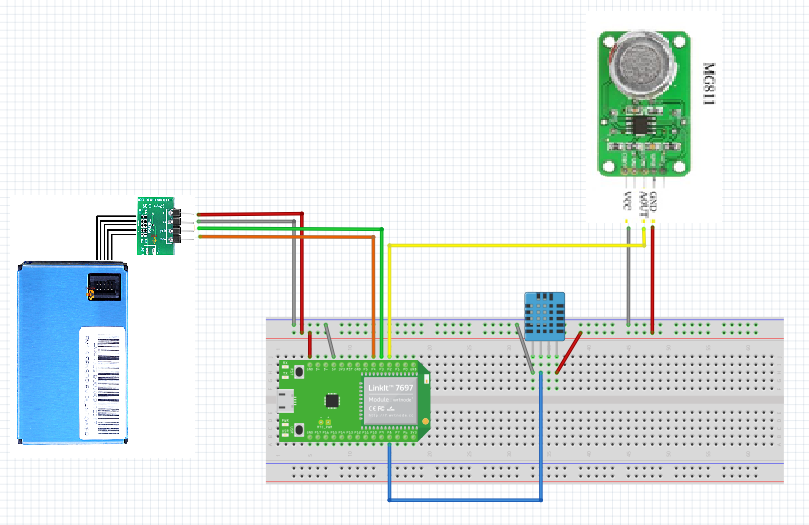


圖3.5電路配置圖

|  |  |
| --- | --- |
| **DHT11** | Linkit7697 |
| **VCC** | **5V** |
| **GND** | **GND** |
| **DATA** | **Pin8** |

表3.1 DHT11感測器與Linkit7697連接針腳

|  |  |
| --- | --- |
| MG811 | Linkit7697 |
| VCC | 5V |
| GND | GND |
| Aout | Pin2 |

表3.2 MG811感測器與Linkit7697連接針腳

|  |  |
| --- | --- |
| PMS7003 | Linkit7697 |
| VCC | 5V |
| GND | GND |
| rx | Pin3 |
| tx | Pin4 |

將於Arduino IDE撰寫各感測器的執行程式，再將程式碼上傳至Linkit 7697執行，最終將透過wi-fi和mqtt將數據傳輸至Adafruit IO已產生圖表。

**第4章  
程式實作**

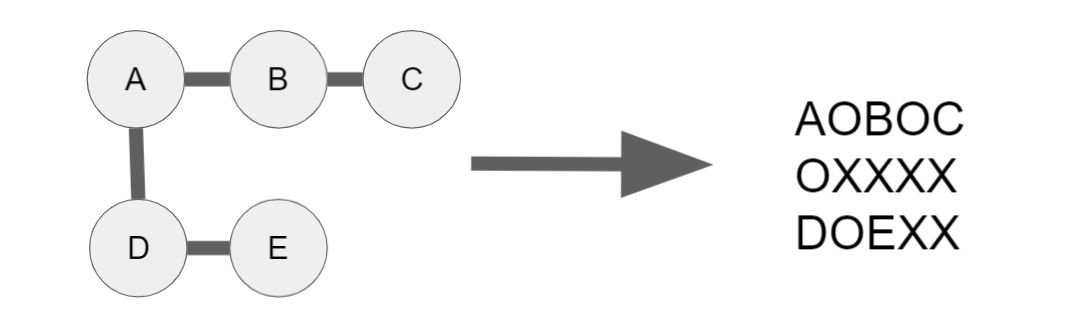
**4-1 網頁部分**

網頁部分主要功能在於，提供介面修改系統內部地圖資料，以及單純選擇起始點。著重於資料處理，並把輸入資料分析成地圖資料保存起來。

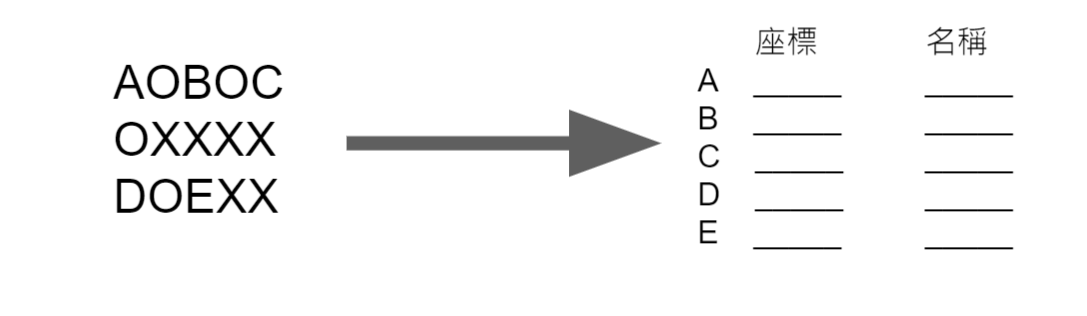
**4-1.1 地圖資料設定與處理**

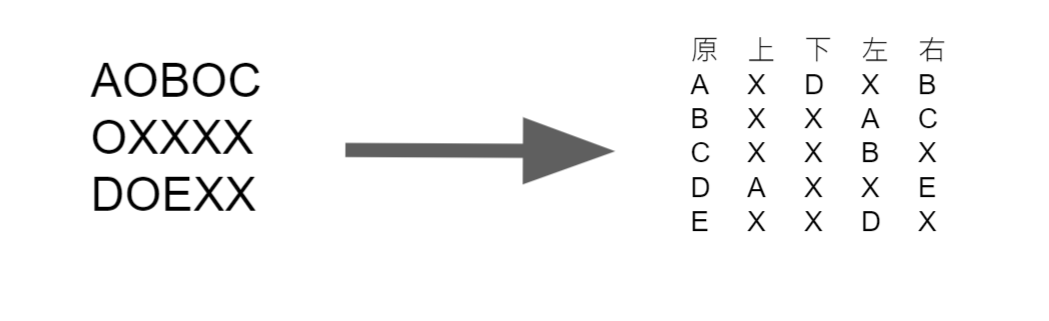
本專題網頁為了使自走車能夠行動所必須的地圖能夠更改與設定，而對其地圖做出了對其應用與資料的限制。首先，自行車的行動方向設定為只能夠前後左右的進行移動，設定完這點之後就可以簡單地單純使用文字來進行地圖資料的傳輸、設定與更改。

地圖資料格式具體設定如下:只使用大寫英文字母，各自代表為位置座標，其中O與X則抓出來賦予其特殊意義，O代表相連兩點為互通的，X則為障礙物或無義，並在周圍補齊X成為最小的矩形，實際表現如下圖:

圖 4.1 地圖格式示範

實際的情況為在網頁端使用form接收地圖輸入資訊，本機端則使用二維矩陣進行存取，並開始分析地圖資訊，抓出各點位置並把各點的相對關係存起來，之後會輸出各點提供給使用者設定實際代號以及座標值。

圖 4.2 抓取格式示範

圖 4.3 分析格式示範

最後，所有資料儲存於本機端供自走車使用，然後提供選擇起點終點的介面，讓使用者操作自走車。

**4-1.2 物品資料設定**

另外，因為應用場景大致上為賣場，所以有提供業者將該點大概所擁有的商品分類上傳以供使用者查詢(LINEBOT)，簡單格式如下。



圖 4.4 物品格式示範

**4-2 Line Bot部分**

Line Bot部分為這次專題互動部分，主要任務為跟使用者互動。本專題簡單設計了幾個互動，分別為:簡易地圖資料介紹、設定起終點以及配合資料庫的座標物品搜尋。

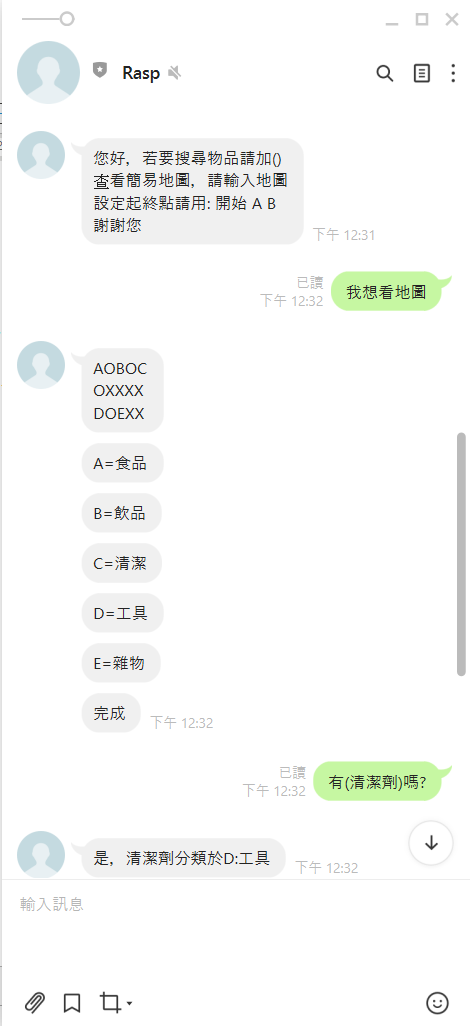
****

圖 4.5 Line Bot實際示範

**4-3 arduino程式**

本程式目標為使linkit 7697 成功讀取感測器，並透過wifi和mqtt訂閱至adafruit io以產生數據圖表







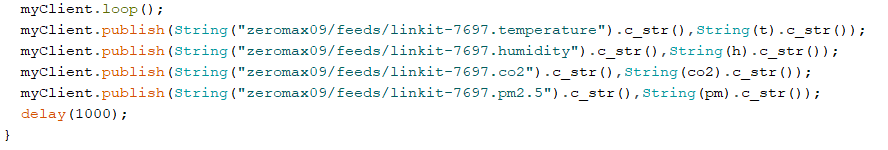


圖 4.6 arduino程式碼



圖 4.7 arduino程式執行結果

**4-5 adafruit io圖表**

Linkit7697燒錄完成後，將會把實時數據利用mqtt訂閱將資料傳adafruit io，並利用網站本身的功能產生圖表。如圖4.8。

****

**圖4.8 adafruit io圖表**

**第5章  
結論**

本專題主要分為兩個部分，一個為提供內部使用者設定與建立地圖資料的網頁端與供外部使用者操作自走車並解決疑惑的通訊軟體端。設計理念為設想基礎狀況並設定基礎功能供使用者解決問題。

另一部分為空氣品質的監測，可透過adafruit io 所提供的數據圖表，讓賣場管理方能即時的收到目前賣場各區域的空氣品質數據，也能配合衛福部所提供的資料，在有群聚風險時，即時的疏散群眾並進行消毒。

所以，本專題主要是在沒有外部界面的自走車上提供基礎界面，供使用者能以比較簡易的方式操作自走車，並同時提供自走車行經區域的空氣品質監測的圖像數據。

**參考資料**

[1] <https://ithelp.ithome.com.tw/articles/10222132>

[2] <https://developers.line.biz/zh-hant/docs/messaging-api/overview/>

[3] <https://github.com/line/line-bot-sdk-python>

[4] <https://mjjoe1017.pixnet.net/blog/post/217770873-%5Bfritzing%5Dfritzing%E8%BB%9F%E9%AB%94%E5%85%8D%E8%B2%BB%E4%B8%8B%E8%BC%89>

[5] <https://sites.google.com/jes.mlc.edu.tw/ljj/linkit7697/mqtt%E7%A9%8D%E6%9C%A8?authuser=0>

[6] <https://learningsky.io/mtk-linkit-7697-connect-mosquitto-use-node-red/>

[7]哈佛大學公共衛生學院- Fine particulate matter and COVID-19 mortality in the United States

<https://projects.iq.harvard.edu/covid-pm/home>

[8] 新英格蘭醫學期刊-Ambient Particulate Air Pollution and Daily Mortality in 652 Cities

https://www.nejm.org/doi/full/10.1056/nejmoa1817364