

淺談IAQ氣體品質量測用於 IoT之應用

三聯科技股份有限公司／王敬淳 

一、前言

隨著工業革命後排放大量的溫室氣體一路到現在21世紀，大氣中的二氧化碳濃度不斷上升，會造成身體上的負擔，減緩二氧化碳濃度是刻不容緩。

大氣中的二氧化碳來源有很多種，其中只要燃燒化石燃料就會產生二氧化碳，化石燃料包含石油、天然氣和煤等，這為二氧化碳的主要來源。

另外，從台灣排放的二氧化碳量來看，工業部門的排放量佔47.84%是最多的，其次是運輸部門佔14.6%，最少的是農業佔1.14%。

台灣也是製造二氧化碳的幫兇之一，根據行政院環保署的統計，台灣於2014年在能源使用上排放約249百萬噸二氧化碳，相當於68萬座台北101在2015年排放的二氧化碳當量（4.7萬公噸），而全球總共排放量為32,381百萬噸二氧化碳，台灣佔全球總排放量的0.77%，全球排名第21名。

當空氣中二氧化碳含量正常的時候，它對人體無危害，但超過了一定含量之後會影響人們的呼吸系統，因為血液中的碳酸濃度增大，酸性增強，並產生酸中毒。

在大自然環境裡，空氣中二氧化碳的正常

含量是0.04%(400 PPM)，在大城市裡有時候達到500 PPM。室內沒有人的情況下，二氧化碳濃度一般在500到700 PPM左右。

如果我們短期處於二氧化碳過高的房間裡，二氧化碳對我們的身體會起到不良的影響，辦公室的空氣中CO₂含量依不同濃度與身體狀況參考如下：

1. CO₂濃度達到1000PPM:員工們會覺得昏昏沉沉、疲倦、工作情緒受到影響。
2. CO₂濃度達到1500-2000PPM:員工們會感覺到氣喘、頭痛、眩暈。
3. CO₂濃度達到2000PPM:員工們會感覺很暈，注意力不集中，精神疲勞。
4. CO₂濃度達到5000PPM以上:人體機能嚴重混亂，使人喪失知覺、神志不清。

短期處於二氧化碳濃度過高，或許不會立即對我們的身體帶來太大的影響。但如果我們長期居住和工作在二氧化碳過高的環境，它會對人體帶來極大的危害。

有鑑於前述，我們除了即知即行，以身作則，在生活中多留意並落實各個能夠減碳的小動作，像是隨手關燈、冷氣不要調太低溫等之外，監測環境中的CO₂濃度，或是辦公場所的CO₂濃度，也成了我們的任務。



二、IAQ氣體品質量測設備介紹

本次的空氣品質監測研發，主要力求能結合IoT，而IoT在當前最主要用到了兩項技術，即MQTT與Modbus。量測的是CO2、VOC、溫度、濕度。

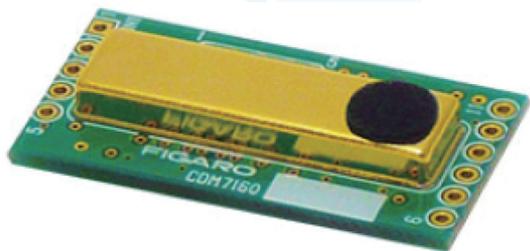
開發平台：Arduino IDE

材料：ESP 32控制器、萬用板、sensor、變壓器、防水塑膠盒、power module、LED、電阻、Relay(5V)

數量：5組

地點：三聯科技總公司、三聯科技新店辦事處

本次系統開發，感測器使用FIGARO的CDM7160及TGS8101這兩顆傳感器(圖1, 圖2)。



▲ 圖1



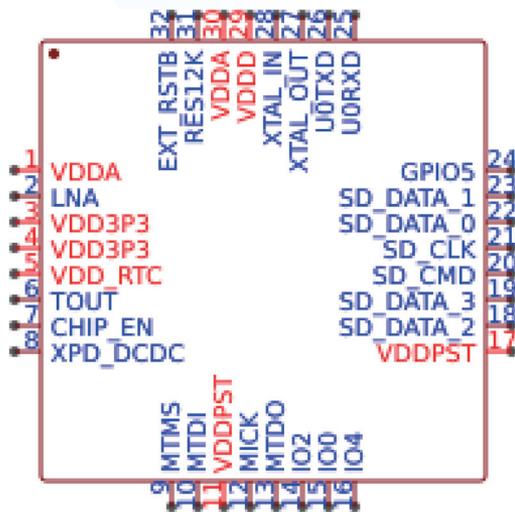
HLGA-10L (2.5 x 2.5 x 0.87 mm)

▲ 圖2

三、氣體品質量測應用方法

首先，先看系統的輸入端，使用ESP32晶片，透過RS232介面，抓取CDM7160上的即時CO2數據，而透過SPI介面，抓取TGS8101上的即時VOC、溫度、濕度資料。

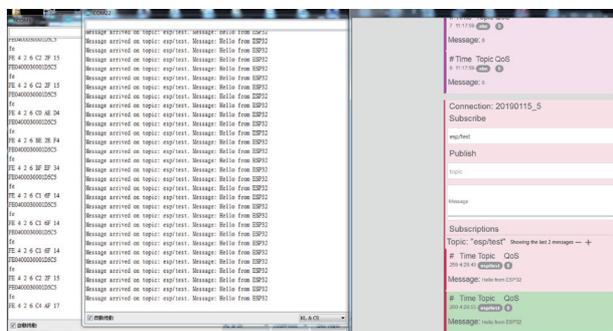
我們可以了解到，ESP32(圖3)是一個微電腦系統，開發者透過程式碼的撰寫，就可擷取傳感器傳回的資料，當然，我們要先確認到，要使用ESP32上的哪一個IO埠來取得資料。當然，我們要知道sensor的供電是須要多少，是3.3V或是5V，地線的腳又是在何處。



▲ 圖3

初步的測試，我們可以先用麵包板來做測試接線，接下來就可搭配程式碼做開發。

撰寫程式時，使用Arduino IDE來編寫程式碼，完成時，再將程式碼燒錄至ESP32來運作。程式的原理，就是依照傳感器規格書所訂定的指令模式下達指令，它便會做出相對應的回應。(圖4)



▲ 圖4

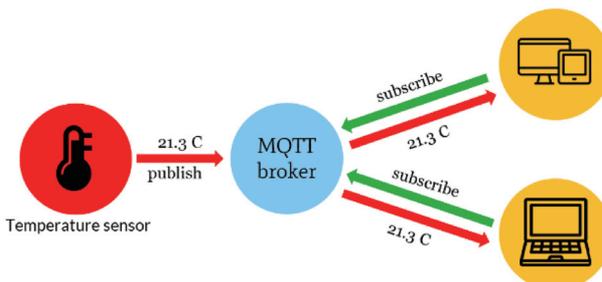


▲ 圖5

撰寫牽涉到硬體的程式，除錯及試誤上，不可或缺的就是電子信號量測設備了。從最簡易的三用電表至示波器或邏輯分析儀等，主要是觀測電子信號是否如我們預期的輸出。有個簡單又重要的概念是，先確認信號有或無，再確認信號的正確性。因為有時會因我們自己主觀的認定，而使得問題的分析變得更複雜。舉個例子來說，假使預期sensor會因我們給他下了指令而回覆我們該有的信號，前提是，有先確認sensor的供電腳位是有電的嗎？電壓是足夠的嗎？如果沒先確認過，後面都是枉然。再舉一個我自己開發上遇到的經驗，擷取TGS8101的資料，是透過SPI傳輸，碰到的情況是，MOSI一直擷取不到sensor回傳的資料，這時有兩種可能方向可以推斷，其一是硬體面，而另一是程式面，用邏輯分析儀量測也量不到信號，但想不到的是，原來ESP32上定義MOSI的腳位IO23，恰巧有兩個，他的信號一直都是從另IO23腳位出來的。

做這樣的開發，就是要硬體及程式兩部分，不斷地做驗證與測試。(圖5)

待資料擷取的部分完成後，接下來也是重要的部分，就是MQTT的部分，也就是要把擷取到的資料，透過MQTT以網路傳送出去，簡單來講，MQTT在架構上，可以分成broker、publish及subscribe；可以想成publish是某雜誌公司，broker是中間的郵務轉發機構，subscribe則是有訂閱某種雜誌主題的讀者。先前提到的ESP32所做的事情，其實目的就是要擷取到的資料，按照定義好的MQTT資料格式，透過ESP32上的WiFi晶片傳送到broker，ESP32扮演的就是publish的角色。那麼誰會收到資料呢，那就是有透過MQTT註冊這訊息的使用者，也就是前述的subscribe。只要ESP32每推送一次訊息，有subscribe訂閱的使用者，便可從broker上收到一次訊息。(圖6)



▲ 圖6



前述從感測器的接收資料，到發送資料至使用者，已經有個清楚的脈絡，我們可以看到，空氣感測器的資料，透過網路，無論是有線或無線網路，就可以即時的傳送給使用者。當然，要達成IoT或物聯網，不僅僅是讓使用者知道訊息，更進一步地，還能做到控制，我們要知道，物聯網有很大一部分，是著重在設備與設備之間的溝通，屬於machine-to-machine(M2M)的輕量級通訊協定。後續接著下來，我們來探討，如何讓這應用，能更符合業界的使用需求，譬如說，某種物理值達到某種程度後，電動機或繼電器便能做出反應，以達到自動控制的效果。

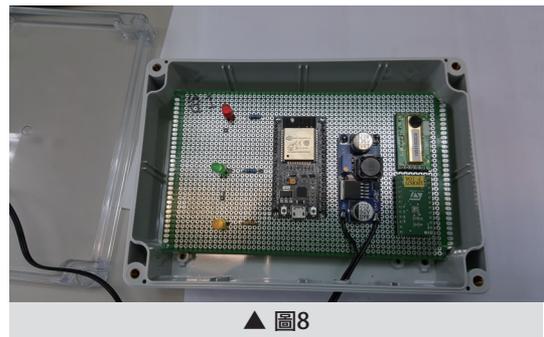
我們將延伸至兩個部分來探討，其一是曾提及的Modbus，另一是創易智能公司的產品-sonoff。Modbus的部分，就是利用先前提到的MQTT broker，在其上將感測器傳來的資料，轉換成Modbus協定所須的封包格式，然後再去驅動PLC，這樣就可以去控制不同的電器開關了。至於sonoff，它其實是一個智慧開關，由於該產品也使用到MQTT，所以將其做為一個研究參考的對象。sonoff硬體裡面有一個relay，可以決定是否導通110V的交流電。如果只是這樣，當然沒什麼特別的，最主要在於，sonoff能夠連上網路，並且藉由MQTT，來控制電器用品。簡單來說，他的這項產品可以直接裝在電器用品的電源輸入的前端，然後使用者可以使用該產品的App，在手機上透過網路，遠端控制電器的on與off。當然，亦是透過MQTT的機制來完成。舉個簡單的例子，在sonoff內訂閱了/home/light1/onoff這個主題(topic)，那麼如果手機App內也對此同一個主題，透過MQTT由網路送出了一個on的訊號，

那此sonoff的這個主題收到on訊號後，就知道要把連接電燈1(light1)的sonoff內部的relay導通，電燈就會亮。

回過頭來看，既然在電路上，以及功能上，已經都具備了，接下來可以進一步的，把原本在麵包板上的電路，移到萬用板或是做layout的設計。最後就是防水盒的設計，本次使用的是光華商場採購的防水盒，買回來再做好鑽孔的加工，孔本身可用矽利康或防水接頭來做防水，這樣就能讓5V的電源線由此孔穿出。(圖7,圖8)



▲ 圖7



▲ 圖8

四、結語

看完了以上整個流程下來的介紹，我們可以初步的了解，如何基本的實作出空氣感測器應用在物聯網上，裝置數量少的時候，或許還感受不出明顯的差異，但是如果有許多不同的裝置，互相交織成一個複雜的網時，將可以改變了我們以往的生活習慣。