



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系

瓦斯外洩警報裝置

指導教授：黃光宇 教授

陳健忠 教授

組員名單：賴喬為 A68C007

陳品沅 A68C031

陳顥交 A68C036

王浚羽 A68C043

中 華 民 國 一 一 〇 年 五 月

嶺東科技大學

資訊管理系

瓦斯外洩警報裝置

中華民國一一〇年五月



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

瓦斯外洩警報裝置

指導教授：黃光宇 教授

陳健忠 教授

組員名單：賴喬為 A68C007

陳品沅 A68C031

陳顥交 A68C036

王浚羽 A68C043

指導教授：_____

口試委員：_____

中 華 民 國 一 一 〇 年 五 月

謝 誌

本專題報告得以順利完成，首先要感謝恩師黃光宇老師、陳健忠老師細心引導本專題，耐心的協助本專題，克服研究過程中所面臨的困難，給予本專題最大的協助，使本專題得以順利完成。

研究報告口試期間，感謝陳健忠老師、黃光宇老師不辭辛勞細心審閱，不僅給予本專題指導，並且提供寶貴的建議，使本專題的專題內容以更臻至完善，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，增進商業管理知識範疇，在此一併致上最高謝意。

賴喬為
陳品沅
陳顥交
王浚羽

謹誌

中華民國一一〇年五月於嶺東

摘 要

由於生活所帶來的便利使人們放鬆警惕，忘記周圍的危險，像是家裡的老人家年紀大了變得比較健忘，使用完瓦斯後忘記關導致危險的發生，迄今為止，光台北市在2015至2019年間，總共就發生了1萬3751件瓦斯外洩案件，其中也造成了不少人員傷亡家庭破碎，為了不再發生類似事件而所產生的遺憾，藉由這個專題的機會與老師共同決定了「瓦斯外洩警報裝置」這個為主題。

而本專題的策劃如下；本專題利用Nodemcu esp8266結合MQ2氣體感測器來感測瓦斯濃度並在濃度指數過高時控制繼電器模塊自動關閉瓦斯，並將MQ2氣體感測器感測到的瓦斯濃度異常數據藉由Nodemcu esp8266傳送到樹莓派，利用樹莓派上的攝影鏡頭捕捉現場畫面，並將圖片以及數據傳送到資料庫和thingspeak數據網站，樹莓派將數據和現場情況圖利用MAIL連動IFTTT傳遞GMAIL和LINE第一時間讓使用者了解現場的情形以便商討解決方法，以及夠有效的警告以及疏散群眾。

本系統達到以下之功能：

1. 接收並傳送感測器資訊：通過Nodemcu串接瓦斯感測器來獲取資訊，並傳送給後端的資料控管伺服器。
2. 數據異常時，控制繼電器開關：當偵測到瓦斯濃度過高之時將會自動關閉瓦斯開關。
3. 數據異常時，傳送訊息告知：確認數值異常後，留存數據並傳送MAIL以及LINE告知用戶。

關鍵字：物聯網、瓦斯外洩、Raspberry Pi、ESP8266

目錄

摘要	I
目錄	II
表目錄	IV
圖目錄	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 物聯網.....	2
2.2 資料視覺化.....	2
第參章 研究方法.....	3
3.1 研究架構.....	3
第肆章 系統分析與設計	4
4.1.系統架構.....	4
4.2 系統開發環境及軟體.....	5
4.2.1 C語言	5
4.2.2 Python語言	5
4.1.3 Raspbian 開發系統.....	5
4.1.4 Arduino ide程式	6
4.1.5 Thingspeak.....	6
4.1.6 IFTTT	7
4.1.7 VNC Viewer	7
4.1.8 Net Analyzer	8
4.1.9 EMAIL	8
4.1.10 LINE.....	9
4.1.11 PuTTY Configuration.....	9
4.2 系統開發環境及硬體.....	10
4.2.1 樹莓派四代開發板.....	10
4.2.2 Nodemcu esp8266.....	11
4.2.3 樹莓派專用攝影鏡頭模組	12
4.2.4 繼電器模塊	13
4.2.5 MQ2氣體感測器	13
第伍章 實作過程與結論	14
5.2 系統配線圖說明.....	15
5.3 ESP8266.....	16
5.3.1 MQ2電壓值轉換ppm公式.....	19

5.4 Python程式碼	21
5.5 實際操作過程	26
第陸章 結論	32
參考文獻	33

表目錄

表 4.1 樹莓派四代規格.....	10
表 4.2 Nodemcu esp8266 規格.....	11

圖目錄

圖 3.1 研究流程圖	3
圖 4.1 系統架構圖	4
圖 4.2 Python.....	5
圖 4.3 Raspbian	5
圖 4.4 Arduino ide.....	6
圖 4.5 ThingSpeak.....	6
圖 4.6 IFTTT	7
圖 4.7 VNC Viewer	7
圖 4.8 Net Analyzer.....	8
圖 4.9 EMAIL	8
圖 4.10 LINE.....	9
圖 4.11 PuTTY Configuration.....	9
圖 4.12 樹莓派四代.....	11
圖 4.13 Nodemcu esp8266	12
圖 4.14 樹莓派專用攝影鏡頭模組.....	12
圖 4.15 繼電器模塊.....	13
圖 4.16 MQ2 氣體感測器	13
圖 5.1 甘特圖	14
圖 5.2 路線配置圖	15
圖 5.3.1 ppm 轉換公式圖	19
圖 5.3.2 Arduino ide 系統流程圖	20
圖 5.4.1 樹莓派系統流程圖.....	25
圖 5.5.1 操作過程 1	26
圖 5.5.2 操作過程 2.....	27
圖 5.5.3 操作過程 3.....	27
圖 5.5.4 操作過程 4.....	28
圖 5.5.5 操作過程 5.....	29
圖 5.5.6 操作過程 6.....	30
圖 5.5.7 操作過程 7.....	30
圖 5.5.8 操作過程 8.....	31

第壹章 緒論

1.1 研究動機

在科技爆發的時代，人們的生活越來越便利，也更容易忘記生活中的小細節，因此為了避免一時疏忽，而抱憾終生的事情發生，本專題將探討該如何避免瓦斯外洩的事情發生。

此系統利用MQ2來感測瓦斯，而實際程序如下：(1)若MQ2偵測到瓦斯之時，將會及時關閉瓦斯口並將數據透過Nodemcu esp8266傳送數據；(2)利用樹莓派接收Nodemcu esp8266的數據，透過文字模式 (Command Line)即可查看及時接收到的數據；(3)透過樹莓派[1]傳送數據到Thingspeak，日後使用者也能到Thingspeak查看數據；(4)當數據異常時樹莓派連動IFTTT及時傳遞EMAIL及LINE告知使用者空氣中的瓦斯濃度異常。

1.2 研究目的

本系統使用Nodemcu和樹莓派結合瓦斯感測器打造瓦斯外洩警報裝置，裝置在廚房檢測及時數據，透過偵測即時數據將其顯示至動態網頁圖表，使用者可透過thingspeak網頁來觀看即時數據。

- 1.解決瓦斯無法自動關閉。
- 2.及時提供瓦斯數據及現場的情形。
- 3.透過樹莓派連動IFTTT及時通知使用者瓦斯外洩的情況。

第貳章 文獻回顧與探討

2.1 物聯網

IoT技術，又稱物聯網，是網際網路、傳統電信網等的資訊承載體，讓所有能行使獨立功能的普通物體實現互聯互通的網路。IoT一般為無線網路，而由於每個人周圍的裝置可以達到一千至五千個，所以IoT可能要包含500兆至一千兆個物體。在IoT上，每個人都可以應用電子標籤將真實的物體上網聯結，在IoT上都可以查出它們的具體位置。通過IoT可以用中心電腦對機器、裝置、人員進行集中管理、控制，也可以對家庭裝置、汽車進行遙控，以及搜尋位置、防止物品被盜等，類似自動化操控系統，同時透過收集這些小事物的資料，最後可以匯聚成巨量資料，包含重新設計道路以減少車禍、都市更新、災害預測與犯罪防治、流行病控制等等社會的重大改變，實現物和物相聯。

2.2 資料視覺化

為了清晰有效地傳遞資訊，資料視覺化使用統計圖形、圖表、資訊圖表和其他工具。可以使用點、線或條對數字資料進行編碼，以便在視覺上傳達定量資訊。有效的視覺化可以幫助使用者分析和推理資料和證據。它使複雜的資料更容易理解和使用。使用者可能有特定的分析任務（如進行比較或理解因果關係），以及該任務要遵循的圖形設計原則。表格通常用於使用者尋找特定的度量，而各種類型的圖表用於顯示一個或多個變數的資料中的模式或關係。本專題運用資料視覺化將軟體及硬體互相連接的關係以圖片的方式更能簡易清晰的表達。

第參章 研究方法

3.1 研究架構

1.規劃

本專題會使用瓦斯感測器MQ2來感測瓦斯濃度，並在即將到達人體危害值得時候，以繼電器控制大電流來開關風扇的方式模擬控制瓦斯的閘門，且透過Python負責傳遞接收到的數據及現場照片至Thingspeak資料庫跟使用者的EMAIL、LINE。

2.分析

使用者需要瓦斯外洩警報器在於當使用者不在家中，家中有任何瓦斯外洩狀況發生可以第一時間通知使用者與應對方法，避免災難發生。且蒐集相關並文獻學習如何撰寫Python語言、C語言，反覆的修改、測試，確認程式可執行之後，並將MQ2感測到的數據透過WIFI的方式傳送給樹莓派，樹莓派再將其數據傳遞至資料庫，日後使用者可以透過Tingspeak查看及時數據了解現場情況。

3.設計

運用Fritzing程式模擬MQ2與Nodemcu esp8266接線，反覆模擬成功之後再將實體的硬體進行接線設計，避免不必要的浪費與困擾，而資料庫的部分則是可以去Thingspeak進行設定，可以透過設定來達到使用者想要的網頁畫面。

4.建置

前端利用Thingspeak來呈現視覺化，而後端則是利用Arduino ide撰寫的Nodemcu esp8266接收MQ2數據再將其數據利用Python撰寫的樹莓派傳送數據到Thingspeak資料庫以及傳遞訊息警告使用者，再透過網路上的文獻以及直接執行找出問題，不斷得修改程式，使程式能夠正常運作且達到設立的目標。

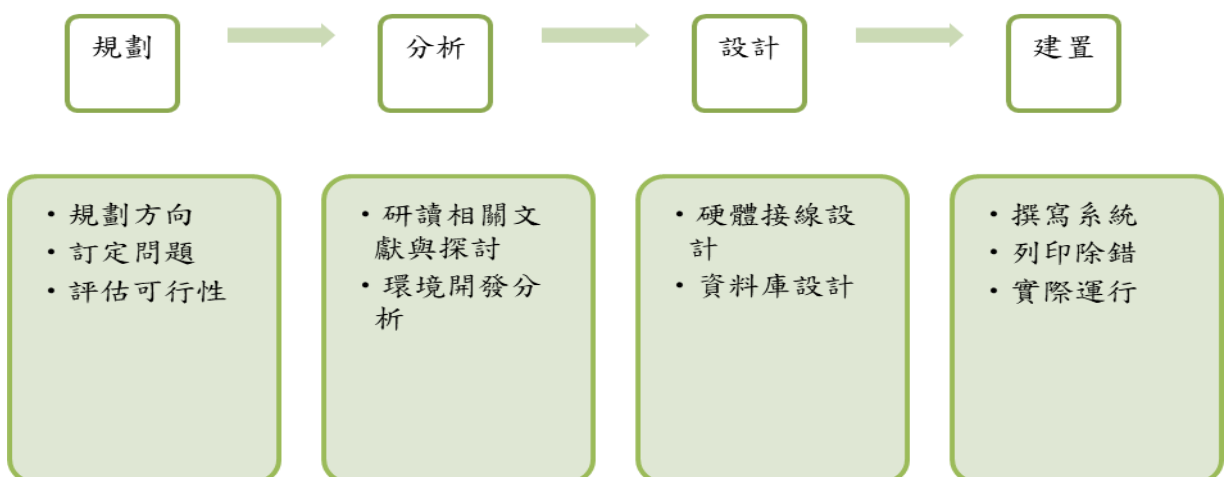


圖 3.1 研究流程圖

4.2 系統開發環境及軟體

4.2.1 C語言

C語言是一種通用的程式語言，廣泛用於系統軟體與應用軟體的開發，C語言具有高效、靈活、功能豐富、表達力強和較高的可移植性等特點，本專題使用Arduino傳送、接收數據及控制繼電器模塊的開與關動作。

4.2.2 Python語言

Python是一種廣泛使用的直譯式、進階和通用的程式語言。Python支援多種程式設計範式，包括函數式、指令式、結構化、物件導向和反射式程式。本專題中Python負責傳遞收集到的瓦斯濃度數據及現場照片到Thingspeak後台資料庫及使用者的Line及E-Mail。



圖 4.2 Python

4.1.3 Raspbian 開發系統

雖然Raspberry Pi提供多款作業系統(OS)給大家選擇，還是建議大家安裝Raspbian，Raspbian是Debian 7.0 wheezy的定制版本，為Raspberry Pi進行優化移植而成，還附帶35000個軟件包及預編譯的軟件，用戶數量最多，教學資源也最豐富，最重要也是Raspberry Pi官方推薦的首選系統。

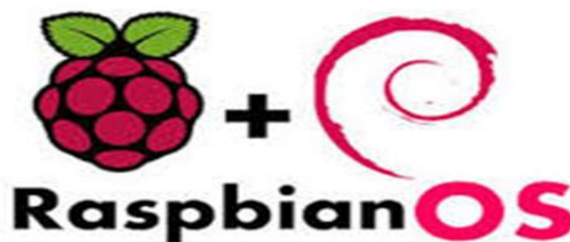


圖 4.3 Raspbian

4.1.4 Arduino ide程式

Arduino Software IDE源自於Processing程式語言以及Wiring計劃的整合開發環境。它是被設計於介紹程式編寫給藝術家和不熟悉程式設計的人們，且包含了一個擁有語法突顯、括號匹配、自動縮排和一鍵編譯並將執行檔燒寫入Arduino硬體中的編輯器。



圖 4.4 Arduino ide

4.1.5 Thingspeak

ThingSpeak是一種開源的物聯網（IoT）應用程序和API，可通過Internet或通過局域網使用HTTP和MQTT協議來存儲和檢索來自物聯網的數據。ThingSpeak支持創建傳感器日誌記錄應用程序，位置跟踪應用程序以及狀態更新的事物社交網絡”。



圖 4.5 ThingSpeak

4.1.8 Net Analyzer

Net Analyzer為手機app，因手機及家用無線網路皆為浮動IP，因此需要此程式來查詢連皆無線網路之後的浮動IP才得以進行連動以達到此裝置的功用。

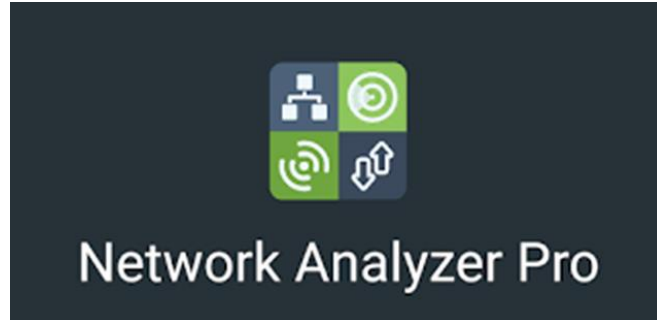


圖 4.8 Net Analyzer

4.1.9 EMAIL

電子郵件（英語：electronic mail），簡稱電郵（email、e-mail），是指一種由一寄件人將數位資訊傳送給一個人或多個人的資訊交換方式，一般會通過網際網路或其他電腦網路進行書寫、傳送和接收信件，目的是達成發信人和收信人之間的資訊互動。一些早期的電子郵件需要寄件人和收件人同時線上，類似即時通訊。現在的電子郵件系統以是儲存與轉發的模型為基礎。郵件伺服器接受、轉發、送出及儲存郵件。寄信人、收信人及他們的電腦不用同時線上。寄信人和收信人只需在寄信或收信時簡短的連線到郵件伺服器即可。

以往電子郵件會用來泛指所有電子式的檔案轉送。例如在1970年代初期有幾位作家用「電子郵件」來描述檔案的傳真。因此很難確定什麼時候開始用「電子郵件」來描述符合現在定義的電子郵件。

電子郵件（符合現代定義，下同）包括三個部份，訊息的「信封」、郵件標頭及郵件內容。標頭會至少包括一些傳遞郵件相關的資訊，例如寄信人的郵件位址及一至多個收信人的郵件位址，一般會包括一些敘述性的內容，例如郵件的標題以及時間等。

電子郵件最早是純文字（ASCII）的溝通媒介，但後來擴展像可以加入多媒體的附件、其他字元集的文字，其程式RFC2045到2049中，同時也產生了多用途網際網路郵件擴展（MIME）。

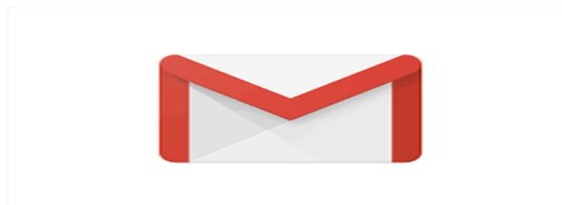


圖 4.9 EMAIL

4.1.10 LINE

LINE是由Z Holdings Corporation旗下LINE株式會社所開發的即時通訊平台，於2011年6月發表。使用者之間可以透過網際網路在不額外增加費用情況下，與其他使用者傳送訊息及觀看直播，並可透過LINE使用購物、行動支付及取得新聞資訊等功能。



圖 4.10 LINE

4.1.11 PuTTY Configuration

PuTTY是一款整合虛擬終端、系統控制台和網路檔案傳輸為一體的自由及開放原始碼的程式。它支援多種網路協定，包括SCP，SSH，Telnet，rlogin和原始的通訊端連接。它也可以連接到序列埠。

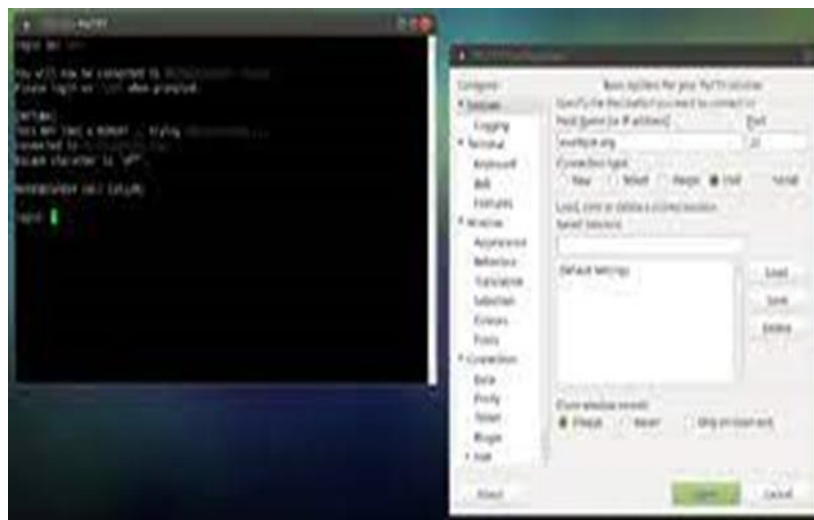


圖 4.11 PuTTY Configuration

4.2 系統開發環境及硬體

4.2.1 樹莓派四代開發板

樹莓派 (Raspberry Pi) [11]，是一款基於Linux的單晶片電腦，具有高效能且豐富的語言開發環境。它由英國的樹莓派基金會所開發，目的是以低價硬體及自由軟體促進學校的基本電腦科學教育，通過 Python、JavaScript、C/C++ 等語言，可以控制樹莓派的 GPIO。通過樹莓派推薦的 Python 或 JavaScript 語言實現串列通信和網路通信，容易實現智慧家居或物聯網的實務應用，與傳統的開發方式比較，更加突顯樹莓派的實用價值。

名稱	技術規格
系統單晶片(SoC)	Broadcom BCM2711
作業系統	Micro SD 卡開機，支援 Linux 與 Windows 10 IoT
中央處理器(CPU)	四核心(ARM v8) Cortex-A72、1.5GHz
工作電流	5 伏 / 800 安培
顯示輸出	兩個 micro-HDMI 埠（可達 4K 解析度、60 幀顯示輸出）
記憶體	4GB LPDDR4-3200 SDRAM
網路功能	2.4 GHz/5.0 GHz IEEE 802.11b/g/n/ac 無線網路，藍牙 5.0 BLE
影音輸出	2-lane MIPI DSI 顯示埠 2-lane MIPI CSI 相機埠 4-pole 立體聲音和複合視訊埠
USB	兩個 USB 3.0 埠、兩個 USB 2.0 埠
顯示器連接功能	Display Serial Interface 端子 (DSI)
讀卡機	Micro SD 讀卡機，支援 SDIO
尺寸	85 毫米 x 56 毫米 x 17 毫米
工作環境溫度	攝氏0至50度

表 4.1 樹莓派四代規格



圖 4.12 樹莓派四代

4.2.2 Nodemcu esp8266

Ai-Thinker生產的ESP-01晶片[17]於2014年8月首次引起了西方的創客們的注意,這個小模塊允許微控制器通過海斯命令集使用TCP/IP協議棧連接到Wi-Fi網絡。然而,在最初的晶片上幾乎沒有使用英文的文檔及可接受的命令。由於模塊上外部組件非常少的原因,導致了其價格十分便宜,並吸引了許多創客來研究和使用的該模塊、開發配套該晶片使用的軟體以及對其使用中文的使用指南進行多語言翻譯。

名稱	技術規格
微控制器	Microchip ATmega328P
CPU	32-bit Tensilica Xtensa LX3
ROM/RAM	64K Boot ROM 64K Instruction RAM與64K Data RAM。
網路應用模式	WiFi 802.11b/g/n 2.4G Radio
GPIO引腳	16個
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	16MiB
隨機存取記憶體	2 KB
EEPROM	1 KB
時脈	16 赫茲
尺寸	25.4mm(W)*48.3mm(H)
重量	25 克

表 4.2 Nodemcu esp8266規格

Nodemcu esp8266接腳說明:

1.Vin: Vin接腳有二種模式,(1)如果有USB供電時,Vin可以供應5V到感測器。(2)沒有透過USB供電時,Vin具備5V穩壓接腳,當輸入5V電壓會轉為3.3V的電壓,再供應給ESP8266晶片。

2.D0: D0接腳直接與ESP8266 GPIO16連接,此接腳內嵌一個active-low LED

燈。

3.D1~D12接腳，每個接腳都能配置做數位的輸出/輸入與PWM數位的輸出，也接供SPI，UART，I2C與one-wire等傳輸協定，使用D1~D12要注意下列三點：

(1)D9與D10分別為UART的傳送接腳(Tx)與接收接腳(Rx)

(2)D11連接到ESP8266 GPIO09使用時，會讓系統自動重置，所以不可以使用

(3)D1~D12接腳都可以使用I2C的clock與data接腳

4.A0，即ESP8266 ADC0接腳，連接到一顆10位元類比轉數位轉換器(ADC)，可用來讀取外界的類比訊號，而讀取A0接腳時，如果不接任何感測器實際上讀出的值為1024。

5.Rx/Tx與D7/D8支援兩組 UART通訊協定。



圖 4.13 Nodemcu esp8266

4.2.3 樹莓派專用攝影鏡頭模組

透過物聯網及視訊攝影機整合應用，使用遠端控制可以將影像即時回傳到使用者手機及管理網頁上，讓決策者可在相同時間不同地點，有效並快速地執行決策方案。



圖 4.14 樹莓派專用攝影鏡頭模組

4.2.4 繼電器模塊

電一體化及電力電子設備中，是最重要的控制元件之一。歸結起來有如下的作用：

1.擴大控制範圍：例如，多觸點繼電器控制信號達到某一定值時，可以按觸點組的不同形式，同時換接、開斷、接通多路電路

2.放大：例如，靈敏型繼電器、中間繼電器等，用一個很微小的控制量，可以控制很大功率的電路。

3.綜合信號：例如，當多個控制信號按規定的形式輸入多繞組繼電器時，經過比較綜合，達到預定的控制效果。

3.自動、遙控、監測：例如，自動裝置上的繼電器與其他電器一起，可以組成程序控制線路，從而實現自動化。

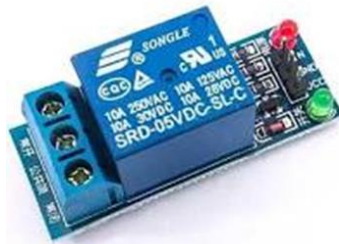


圖 4.15 繼電器模塊

4.2.5 MQ2氣體感測器

- 1、具有信號輸出指示[6]。
- 2、雙路信號輸出（類比量輸出及TTL電平輸出）
- 3、TTL輸出有效信號為低電平。（當輸出低電平時信號燈亮，可直接接單片機）
- 4、類比量輸出0~5V電壓，濃度越高電壓越高。
- 5、對液化氣，天然氣，城市煤氣有較好的靈敏度。
- 6、具有長期的使用壽命和可靠的穩定性。
- 7、快速的回應恢復特性。



圖 4.16 MQ2氣體感測器

第五章 實作過程與結果

5.1 甘特圖

本專題先擬定專題，建立系統架構、撰寫，進行數據資料收集及專題撰寫，將系統整合之後透過測試與除錯確定系統能夠正常執行，最後完成本專題。

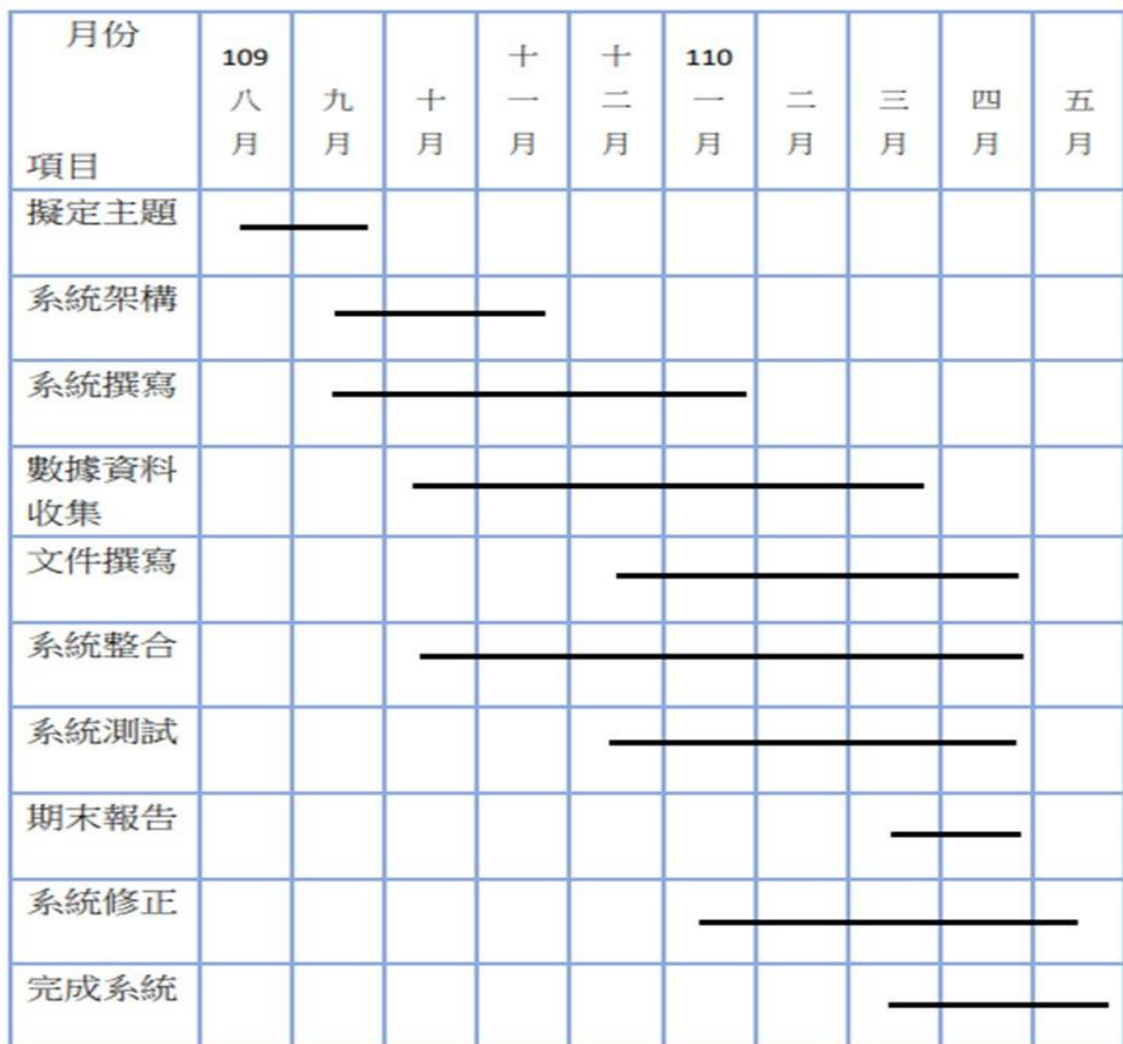


圖 5.1 甘特圖

5.2 系統配線圖說明

紅色實線為電源線，利用USB供電給Nodemcu，Nodemcu的Vin接腳可供應5V到感測器跟繼電器，黑色線為地線，因此將所有的負極(黑色線)接在一起共地，白線代表的是MQ2數位輸出的配置，而MQ2的DO則是能提供存在的瓦斯氣體含量，黃線為AO類比輸出，將其接到nodemcu的A0，A0能夠讀取類比資料，而AO能夠提供與瓦斯濃度成比例的模擬輸出電壓，而會運用到電阻是因為要確定高電壓的一方，它的輸出電壓經過轉換後不能超過低電壓一方的絕對最大定額值，不然就會把低電壓這邊的 IC 弄壞。

這裡有一個優雅又簡潔的解決方法：分壓電路；我們可以很簡單地用一個兩顆電阻組成的分壓電路，來讓高電壓側的輸出轉換成低電壓側可以接受的電壓範圍。

而繼電器的IN(控制開關腳位)為藍線，接到Nodemcu的D2，使Nodemcu能用開關瓦斯的管道。

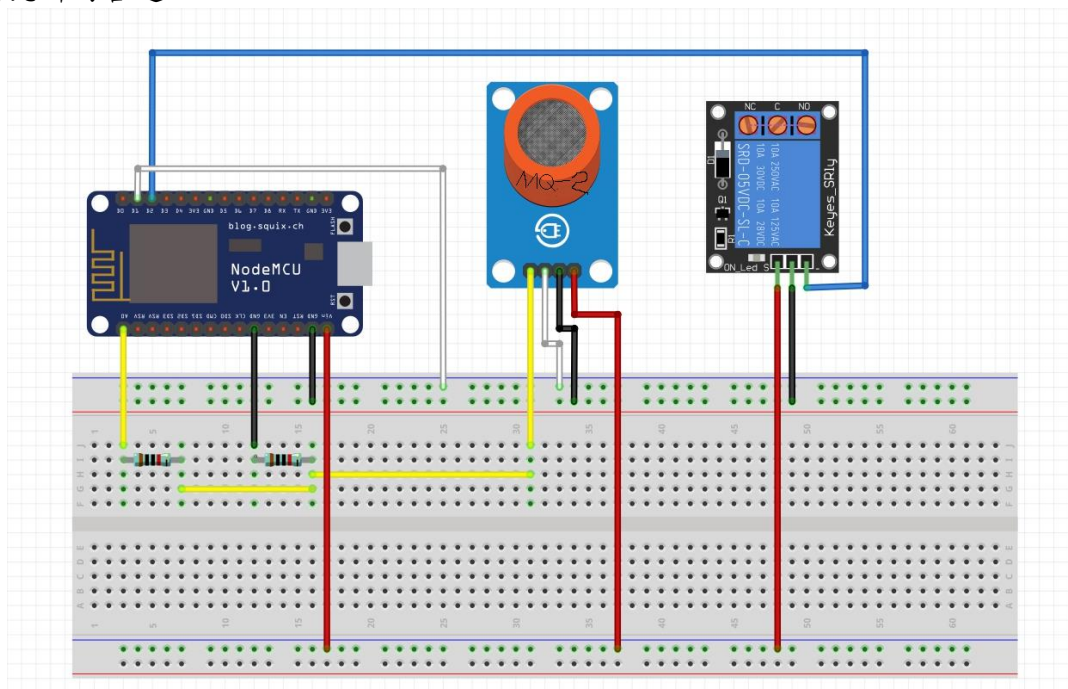


圖5.2 路線配置圖

5.3 ESP8266

下方程式碼用意是啟動ESP8266內建WIFI[13](啟動WIFI)

```
#include <ESP8266WiFi.h>
```

本專題先設定WIFI的熱點名稱以及密碼，然後創建TCP server對象，且將端口指定為80，#define gas A0 為氣體感測器的類比輸出接到nodemcu的A0，#define gas_Dout 5為氣體感測器的數位輸出接到nodemcu的5(D1)，#define Relay_switch 4為繼電器控制腳接到nodemcu的4(D2)。

```
const char* ssid = "35353";  
const char* password = "jj871104";  
WiFiServer server(80);  
#define gas A0  
#define gas_Dout 5  
#define Relay_switch 4
```

下方pinMode(gas, INPUT);為氣體感測器的類比輸出接到nodemcu的A0,為輸入

pinMode(gas_Dout, INPUT);為氣體感測器的數位輸出接到nodemcu的5(D1),為輸入(將類比訊號做連接)

```
void setup() {  
  Serial.begin(115200);  
  delay(10);  
  pinMode(gas, INPUT);  
  pinMode(gas_Dout, INPUT);  
}
```

(此段為設定WIFI模式及連接WIFI。)

```
Serial.println();  
Serial.println();  
Serial.print("Connecting to ");  
Serial.println(ssid);  
WiFi.mode(WIFI_STA);  
WiFi.begin(ssid, password);
```

本專題設定每隔0.5秒檢測WIFI連線是否成功，沒有連線成功會印出連線失敗的訊息。server.begin(); 為服務器開始偵聽傳入的連接。

Serial.println(WiFi.localIP());意旨為印出IP地址。(測試WIFI有沒有連接成功)

```
while (WiFi.status() != WL_CONNECTED) {  
  delay(500);  
  Serial.print(".");  
}  
Serial.println("");  
Serial.println("WiFi connected");  
server.begin();  
Serial.println("Server started");  
Serial.println(WiFi.localIP());  
}
```

宣告狀態字串變數，一開始為Safety(安全)，讀取AO電壓，*2是因為用了2個10K的電阻分壓。每0.1秒讀取氣體感測器狀態並打開瓦斯管道，如果狀態字串出現警告將關閉瓦斯管道。(偵測數值是否安全若不安全將關閉瓦斯管道)

讀取用戶是否已加入連線，如果無則回到程式最開始[5]。(讀取用戶連線若沒連線重新連線)

```
WiFiClient client = server.available();  
if (!client) {  
  return;  
}
```

(此段為列印信息以及等待數據)。

```
Serial.println("new client");
while (!client.available()) {
  delay(5);
}
```

本專題設定每隔0.5秒檢測WIFI連線是否成功，沒有連線成功會印出連線失敗的訊息。server.begin(); 為服務器開始偵聽傳入的連接。

Serial.println(WiFi.localIP());意旨為印出IP地址。(測試WIFI有沒有連接成功)

```
void loop() {
  String Status = "Safety";
  int vol = analogRead(A0) * 2;
  delay(100);
  if (digitalRead(gas_Dout) == 1 )
  {
    pinMode(Relay_switch, OUTPUT);
  }
  else
  {
    Status = "warning";
    pinMode(Relay_switch, INPUT);
  }
}
```

(將ESP8266所接收到的訊息整理完後，全部傳送給樹梅派4代。)

```
String req = client.readStringUntil('\r');
Serial.println(req);
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nStatus=" + Status + "data=" + String(vol);
client.print(s);
```

(中斷連線。)

```
delay(1);
Serial.println("Client disconnected");
client.flush();//結束連線
}
```

5.3.1 MQ2電壓值轉換ppm公式

將感測到的MQ2電壓值轉換成瓦斯濃度(ppm)，透過轉換能夠更了解當前的瓦斯濃度是否對人體有害，以及如何做接下來的處置。

```
1 #define CAL_PPM 20 // 校准环境中PPM值
2 #define RL 5 // RL阻值
3 static float R0 = 6.00; // 元件在纯净空气中的阻值
4
5 //得到ADC采样内部传感器的值
6 //取10次,然后平均
7 vu16 Get_ADCValue_MQ2(void)
8 {
9     u32 val = 0;
10     u8 times = 10;
11     u8 count;
12     for(count = 0; count < times; count++)
13     {
14         val += ADC_ConvertedValue[0]; //获取DMA通道值
15         delay_ms(5);
16     }
17     return val/times;
18 }
19
20 // 传感器校准函数
21 void MQ2_PPM_Calibration(float RS)
22 {
23     R0 = RS / pow(CAL_PPM / 613.9f, 1 / -2.074f);
24 }
25
26 // MQ2传感器数据处理
27 float MQ2_GetPPM(void)
28 {
29     float Vr1 = 3.3f * Get_ADCValue_MQ2() / 4096.f;
30     Vr1 = ( (float)( (int)( (Vr1+0.005)*100 ) ) )/100;
31     float RS = (3.3f - Vr1) / Vr1 * RL;
32
33     if(times_mq < 6000) // 获取系统执行时间, 300ms前进行校准
34     {
35         R0 = RS / pow(CAL_PPM / 613.9f, 1 / -2.074f);
36     }
37
38     float ppm = 613.9f * pow(RS/R0, -2.074f);
39
40     return ppm;
41 }
```

圖5.3.1 ppm轉換公式圖

利用USB供電，供電之後會自動連線WIFI[16]，若是沒有連接到WIFI將會跳出警訊，並重新連線，若有連接到WIFI將進行Arduino ide 迴圈設定，以每秒0.1偵測瓦斯濃度，若指數正常將進行重複動作偵測，若指數不正常將進行繼電器關閉的動作，抓取數據[16]並將數據傳送樹莓派。

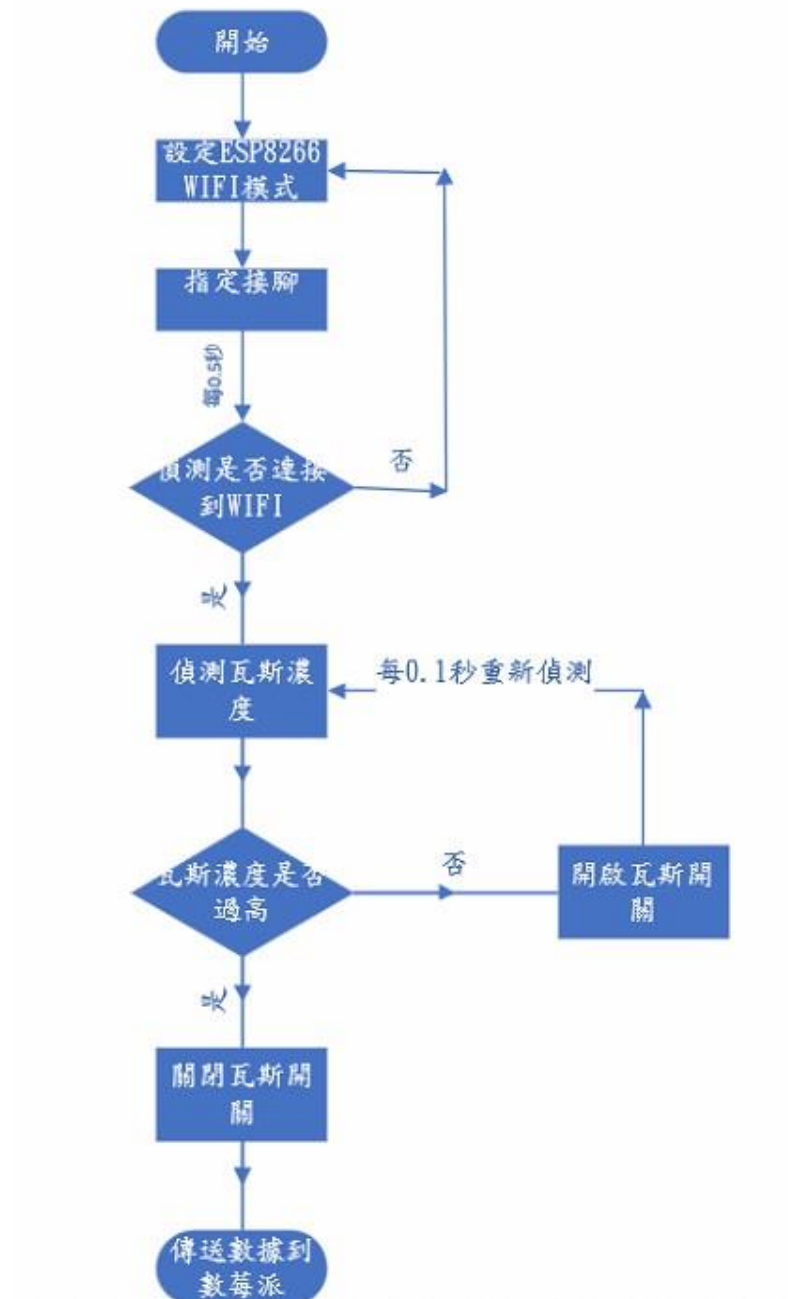


圖5.3.2 Arduino ide 系統流程圖

5.4 Python程式碼

本專題先進行編碼設定為 utf-8，再匯入各種pip函式庫來運作python系統。
(將編碼設定為可變長度自原編碼(utf-8)，再匯入Python電腦程式語言寫成的軟體包管理系統(pip)運作python系統)

```
# coding: utf-8
import cv2
import sys
import urllib2
from time import sleep
import time
import smtplib
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
```

(之後本專題設定獲取內建攝影機，如果獲取失敗的話，將終止程式)

```
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open webcam")
```

(這邊本專題進行宣告資料變數)

```
data = "
string = " "
old_time_nodemcu = 0.1111
old_time_Upload = 0.1111
old_time_photo = 0.1111
```

之後要將數據上傳到thingspeak，因此需要將API密鑰寫入，才能進行傳送數據的動作

(將數據傳到thingspeak平台，呼叫應用程式介面(API)呼叫密鑰進行傳出)

```
def Upload_cloud():
    baseURL =
    'https://api.thingspeak.com/update?api_key=0BYSDDZBUVC2MDOC'
    try:
        global data
        f = urllib2.urlopen(baseURL + "&field1=" + data)
        print data
        f.close()
    except:
        print 'exiting.'
```

接下來要連上nodemcu的IP地址來獲取nodemcu回傳的數據資料，並找到關鍵字data=,後面的資料並且截取之後回傳

(連接nodemcu的IP地址內得data等於後的關鍵字做回傳數據。)

```
def get_nodemcu_data():
    baseURL = 'http://192.168.1.105/'
    try:
        f = urllib2.urlopen(baseURL)
        string = f.read()
        global data
        global string
        data = string[5 + string.find('data=):']
        print(data)
        print(type(data))
        f.close()
    except:
        print 'exiting.'
```

(獲取攝影機畫面，以及設定縮小0.5倍，顯示攝影機畫面視窗，讀取鍵盤按鈕，如果按下ESC鍵就結束程式。)

```
while True:
    ret,frame =cap.read()
    frame =
cv2.resize(frame,None,fx=0.5,fy=0.5,interpolation=cv2.INTER_AREA
)
    ##cv2.imshow('Input',frame)
    c = cv2.waitKey(1)
    if c == 27:
        break
```

前者為列印除錯，如果nodemcu傳來資料內含'warning'文字，每隔30秒傳送一封email，還有透過IFTTT服務傳LINE還有EMAIL

(如果數值過高系統做警告每30秒傳送email以及line)

```
print time.time() - old_time_nodemcu >= 2
print time.time() - old_time_nodemcu >= 15
if (time.time() - old_time_nodemcu >= 2):
    get_nodemcu_data()
    global data
    global string
    if(time.time() - old_time_photo > 30 and
string.find('warning') > 0):
```

```

cv2.imwrite("/home/pi/10.jpg", frame)
        with open("/home/pi/10.jpg", "rb") as file:
            filecontent=file.read()
            mime=MIMEImage(filecontent, "image/jpeg")
            mime["Content-Type"]="image/jpeg"
            mime["Content-Disposition"]='attachment;
filename="10.jpg"'
            mime["Subject"]="Sensor abnormal--Value = " +
data
            mime["From"]="Raspberry Pi"
            mime["To"]="mailgroup"
            msg=mime.as_string()

```

這邊是設定EMAIL以及IFTTT的設計，透過傳遞電子郵件[7]的方式觸發程序，並且將數值與圖片利用程序傳遞使用者與瓦斯公司。

(設定EMAIL與IFTTT，透過電子郵件觸發程序，將數值與圖片傳遞使用者與公司)

```

        print(msg)
        smtp=smtplib.SMTP("smtp.gmail.com", 587)
        smtp.ehlo()
        smtp.starttls()
        smtp.login("jooe871104@gmail.com",
"123456789")
        from_addr="jooe871104@gmail.com"
        to_addr=["hea345789@yahoo.com.tw"] +
["trigger@applet.ifttt.com"]
        status=smtp.sendmail(from_addr, to_addr, msg)
        if status=={}:
            print("SENT OK")
        else:
            print("SENT ERROR")
        smtp.quit()

```

(每隔15秒上傳資料到雲端資料庫，並且關閉攝影機以及視窗。)

```
old_time_photo = time.time()
old_time_nodemcu = time.time()
if (time.time() - old_time Upload >= 15):
    Upload_cloud()
    old_time = time.time()
cap.release()
cv2.destroyAllWindows()
```

利用USB進行供電，透過WIFI使樹莓派與Nodemcu在相同網域進行運作，若數據異常，接收數據，並只擷取Data後面的數值並將其傳輸至使用者以及數據平台網站，樹莓派會立即拍攝一張現場情況圖一同傳輸給使用者。



圖5.4.1 樹莓派系統流程圖

5.5 實際操作過程

將撰寫好的Arduino ide燒入至ESP8266的過程。

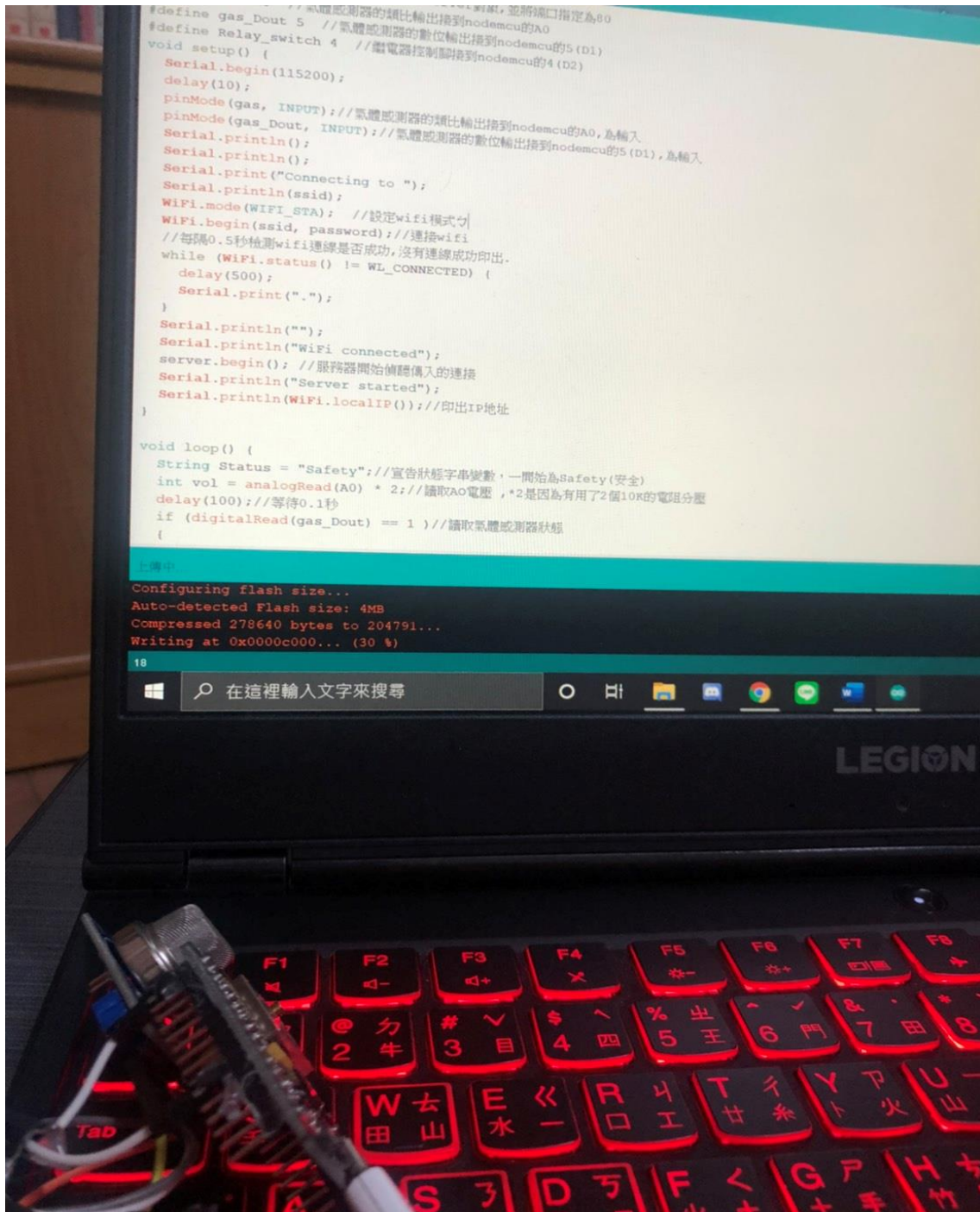


圖5.4.1 操作過程1

Arduino ide燒入成功之後並嘗試執行查看是否能夠正常運作。

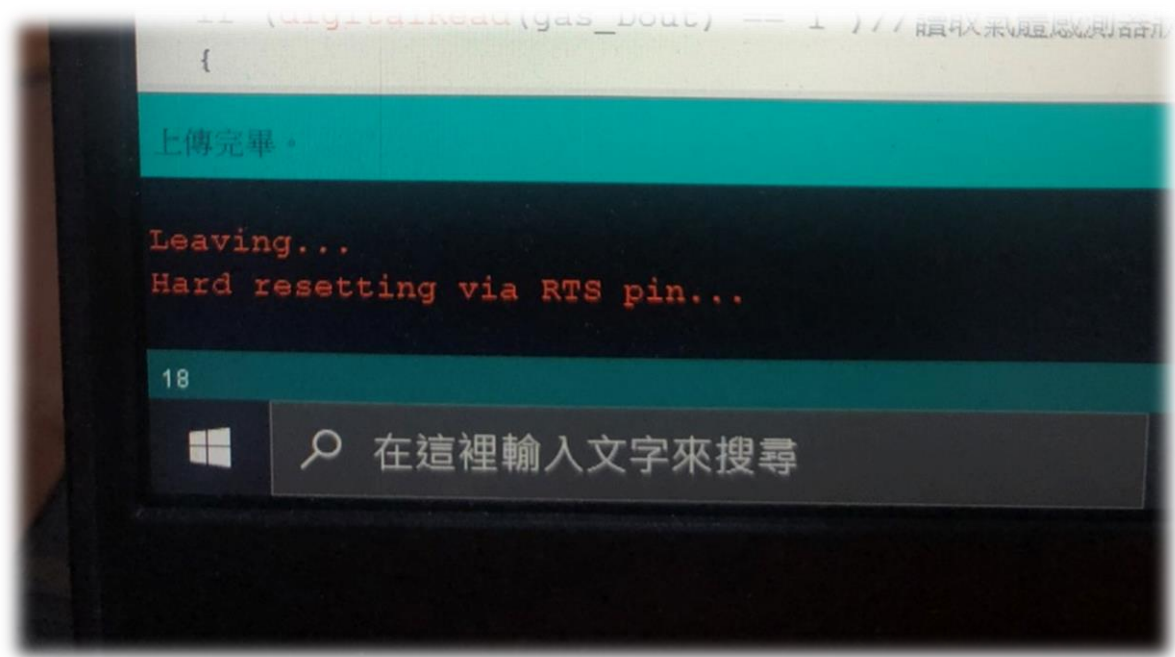


圖5.4.2 操作過程2

之後將撰寫好的python程式碼利用USB複製到樹莓派的檔案總管裡面。

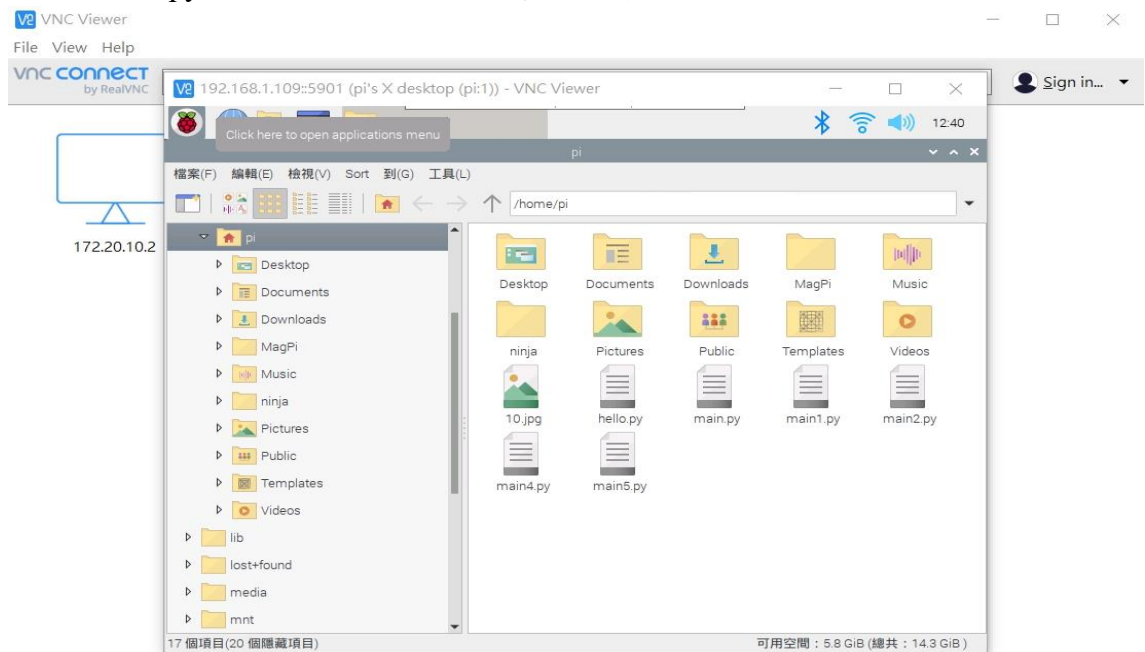


圖5.4.3 操作過程3

而這透過電腦利用VNC來連接樹莓派的畫面來提升便利性並且能查看是否有誤以及正常執行。

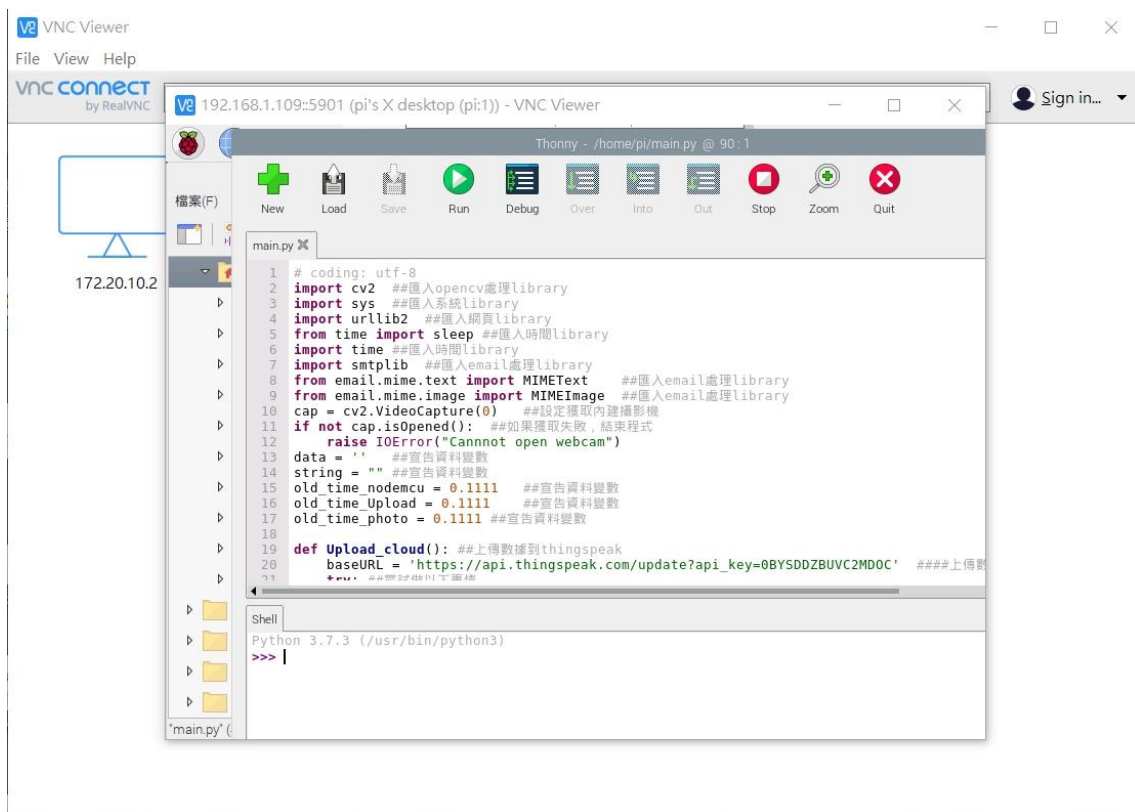


圖5.4.4 操作過程4

利用手機APP Net Analyzer查看ESP8266確實有連到WIFI以及跟樹莓派在同個網域。

P-可ping通(綠色):設備響應ICMPping請求

B-Bonjour(棕色):設備提供Bonjour服務

U-UPNP/DLNA(藍色):設備提供UPNP/DLNA服務

G-網關(紅色):設備是網關

S-掃描設備(黃色):執行掃描的設備

6-IPv6(紫色):設備的已知IPv6地址

W-可用的網絡界面(青色):端口80或443打開



圖5.4.5 操作過程5

確定連上WIFI並在相同網域之後打開PUTTY輸入裝置名稱和密碼之後執行系統。

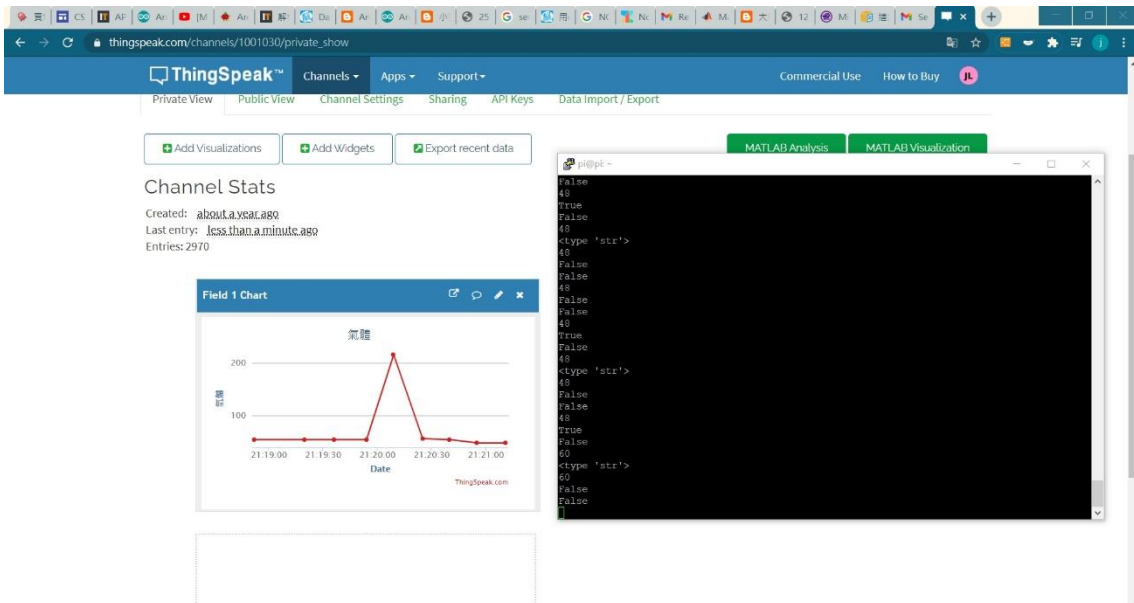


圖5.4.6 操作過程6

當感測到瓦斯濃度飆升時，控制繼電器關閉瓦斯閘門，以及立即的傳輸數據與照片警告使用者與瓦斯公司。

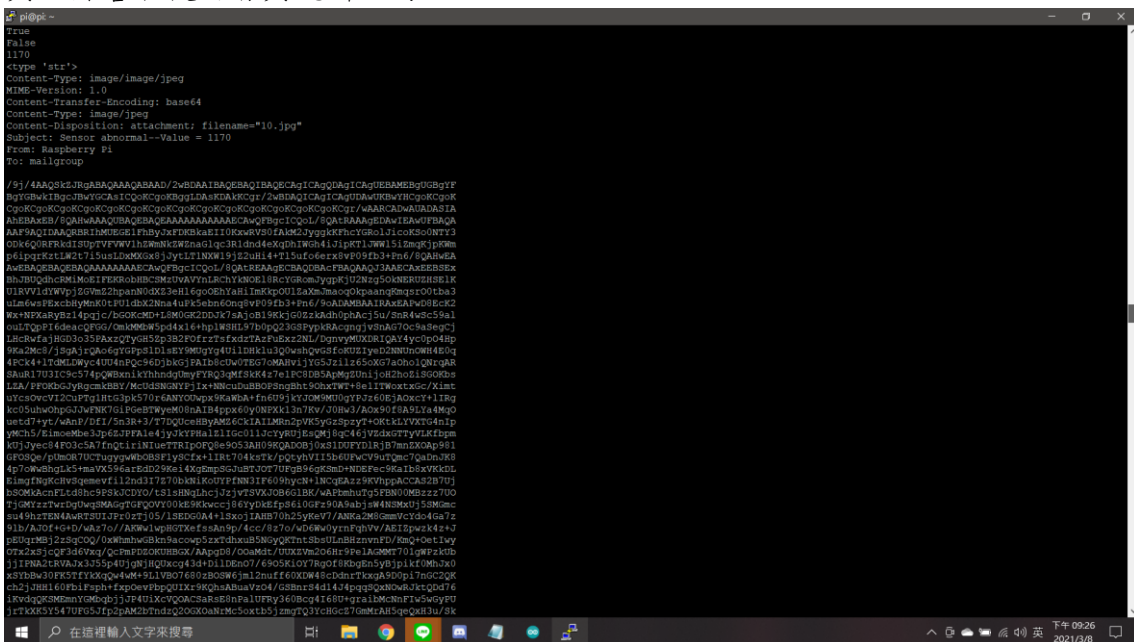


圖5.4.7 操作過程7

當瓦斯濃度飆升時，使用者會接收到預先設定好的警示訊息，也會傳送至瓦斯公司，以此達到警告之作用與預防疏漏。

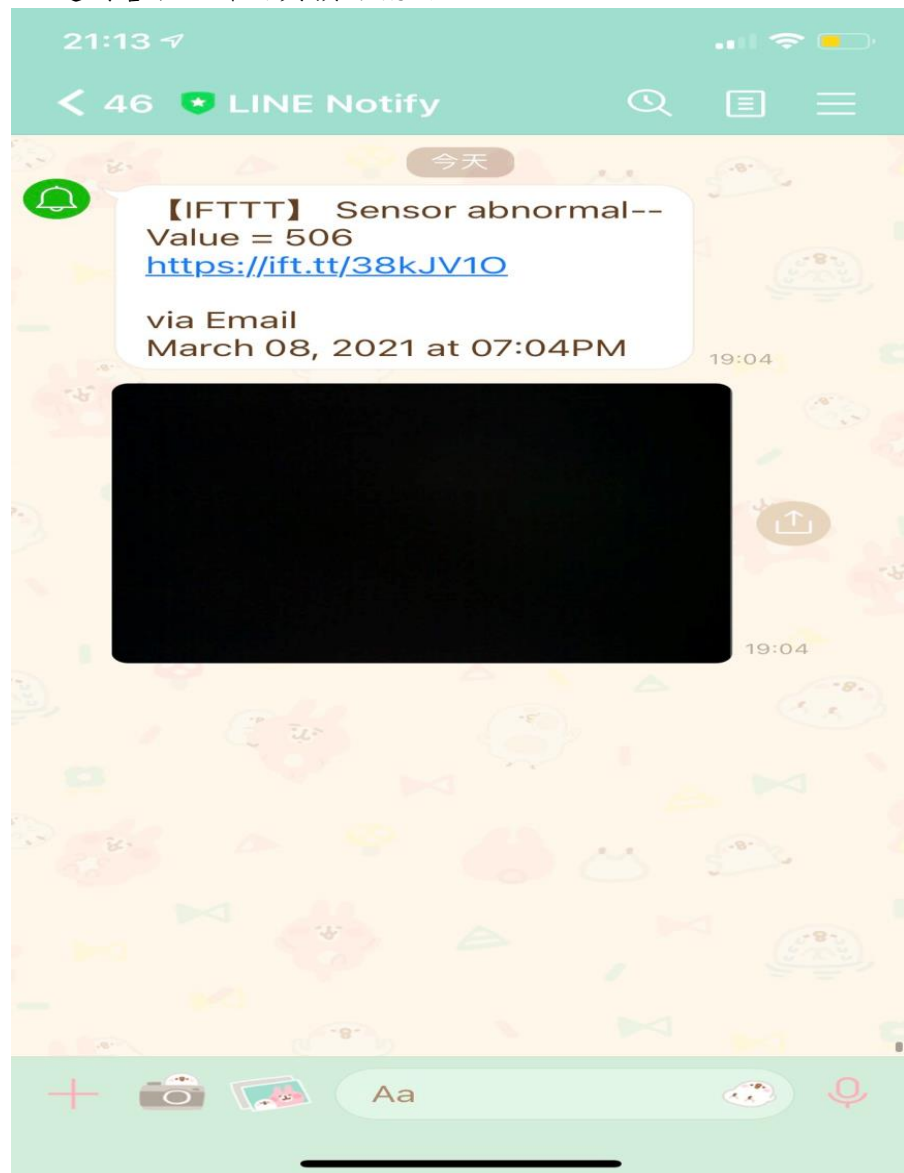


圖5.4.8 操作過程8

第陸章 結論

本專題利用Nodemcu esp8266結合MQ2氣體感測器來感測瓦斯濃度並在濃度指數過高時控制繼電器模塊自動關閉瓦斯，並將MQ2氣體感測器感測到的瓦斯濃度異常數據藉由Nodemcu esp8266傳送到樹莓派，利用樹莓派上的攝影鏡頭捕捉現場畫面，並將圖片以及數據傳送到資料庫和thingspeak數據網站，樹莓派將數據和現場情況圖[2]利用MAIL連動IFTTT傳遞GMAIL和LINE第一時間讓使用者了解現場的情形以便商討解決方法，以及夠有效的警告以及疏散群眾。本次專題學習到了學習如何使用C++來控制繼電器及MQ2，並用樹莓派來連接整個接收與傳送的功能；學習物聯網的技術，讓使用者能夠更快且安全的了解現在情況;學習如何分工合作解決問題並且學以致用。

本專題達到以下目的：

1.接收並傳送感測器資訊: 通過Nodemcu串接瓦斯感測器來獲取資訊，並透過樹莓派傳送給後端的資料控管伺服器。

2.數據異常時，控制繼電器開關: 當偵測到瓦斯濃度過高之時將會透過Nodemcu關閉瓦斯開關。

3.數據異常時，傳送訊息告知: 當偵測到瓦斯濃度過高之時將會透過Nodemcu關閉瓦斯開關。

參考文獻

- [1] 維基百科, 2018, 樹莓派。(2018-06-06)。
網址：
<https://zh.wikipedia.org/wiki/樹莓派>。
- [2] 蘇威宇, 2016, 遠端無線網路控制機器人暨雲端
視訊攝影機系統實作, 碩士論文, 高雄: 義守大學電機工程學系。
- [3] 智會物聯網研究中心, 2015, DHT11 溫濕度感測器介紹。(2018-11-16)。
網址:
https://iot.ttu.edu.tw/2015/11/16/dht11_溫濕度感測器介紹/。
- [4] JavaScript | MDN, 2019, JavaScript。(2018-03-23)。
網址:
<https://developer.mozilla.org/zhTW/docs/Web/JavaScript>。
- [5] 維基百科, 2019, C 語言。(2019-04-04)。
網址：
<https://zh.wikipedia.org/wiki/C%E8%AF%AD%E8%A8%80>。
- [6] 曹永忠、許智誠、蔡英德 (2016)。《Ameba程式教學(MQ氣體模組篇)》。彰化縣: 渥瑪數位有限公司。
網址:
http://ebooks.lib.ntu.edu.tw/1_file/ntulib/106031586/86-%E6%B0%A3%E9%AB%94%E6%A8%A1%E7%B5%84%E6%95%99%E5%AD%B8%E6%95%99%E5%AD%B8V8%E5%AE%8C%E6%95%B4%E7%89%88.pdf
- [7] 小狐狸事務所(2018,07)。以Gmail寄送郵件的方法。(2021.1.15)
網址:
<http://yhhuang1966.blogspot.com/2018/10/python-gmail.html>
- [8] 小狐狸事務所(2015,09)。Arduino 基本語法筆記。(2021.1.15)
網址：
http://yhhuang1966.blogspot.com/2015/09/arduino_14.html
- [9] LazyTomato Lab 懶番茄工作室(2017,10)。Arduino IDE 內建的人機互動工具。(2019.12.13)
網址：
<https://www.lazytomatolab.com/as-11/>
- [10] Arduino.cc(2019,07)。Arduino開發板包含一個多通道10位模數轉換器。(2021.04.01)
網址：
<https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>
- [11] CTW網站(2020,09)。第四代樹莓派 Raspberry Pi 4 Model B 4GB RAM 的規格。(2019.11.06)
網址：
<https://blog.gtwang.org/iot/raspberry-pi-4-model-b-4gb-ram/>

- [12]阿德的STEAM教學網(2020,02)。 NodeMCU簡介。(2020.10.15)
網址：
<http://itcgs.tcgs.tc.edu.tw/yute?cid=2254>
- [13] ESP8266 Arduino Core(2017)。 ESP8266 WiFi庫。(2020.11.24)。
網址：
<https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
- [14] Vincent,VIN電腦程式筆記本(2016,11)。 ESP8266 的AP測試。(2021.03.25)。
網址：
<http://twincati.blogspot.com/2016/11/esp8266-ap.html>
- [15] 新北中山國中創客社群(2016,06)。 ESP8266獨立操作。(2021.03.25)。
網址：
<https://sites.google.com/site/csjhmaker/esp8266-xiang-guan/esp8266du-li-yun-zuo>
- [16] 大榔頭的電腦隨筆(2018,07)。 ESP8266抓取程序教學。(2021.03.25)。
網址：
<http://hammer1007.blogspot.com/2017/12/esp82663-2.html>
- [17]IT邦幫忙,阿旺(2019,09)。 ESP8266晶片介紹。(2021.04.02)。
網址：
<https://ithelp.ithome.com.tw/articles/10222071> ESP8266
- [18]MQ2電壓值與PPM轉換公式,代碼先鋒網(2018,09)。 MQ2電壓值與PPM轉換公式。(2021.04.05)。
網址：
<https://www.codeleading.com/article/54803735384/>