1. 研究過程或方法**:**
2. **架構圖:**
3. **利用uno單獨測試DHT溫濕度感測器:**

溫濕度偵測是本實驗最核心的資訊，第一步必須先確定DHT11可以正常工作。由於選用TBIOTMCM電路板，其中整合DHT11溫濕度感測器，在esp01主板還未能燒錄程式時，可以先利用uno板來做驗證，只需將感測器三個針腳焊接出來，連接uno的針腳，程式碼燒錄在uno上，即可快速確定是否能正常抓取溫濕度數值。由於uno燒錄速度遠比esp01快，此步驟，不單在除錯上提供獨立的驗證環境，可方便後續做交叉驗證。

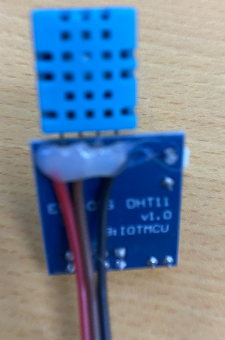


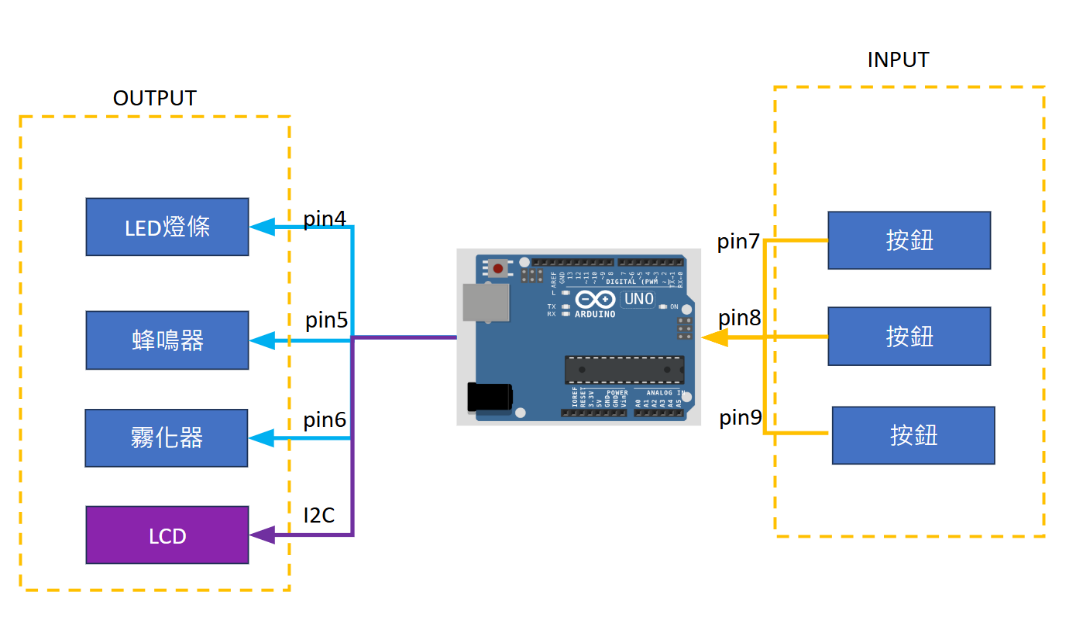
圖 2 uno 測試dht11 感測器

圖 1焊接出vcc,data,gnd針腳



圖 3 uno程式運行狀況

1. **Arduino 感測器歸類:**

****

在 Arduino 程式碼中，將 Input 與 Output 做明確的歸類有助於程式碼的結構化和可維護性。在程式開頭使用 #define 或 const 定義所有輸入與輸出的引腳，這樣在需要修改或擴展時，只需調整定義即可，減少出錯機率。接著，在 setup() 函數中使用 pinMode() 明確設定每個引腳是 INPUT 或 OUTPUT，確保 Arduino 正確識別硬體的功能。最重要的目的，在排除硬體問題上，可以更有效率找到問題。

1. **自製esp01燒錄器:**

esp01沒有usb hub接口，無法直接連接電腦，透過網路上提供的方式，可以自己做一組燒錄器，比較麻煩的是它的電壓是3.3V，透過電腦usb接出來的電源都是5V，會造成esp01主板毀損，幸運的是Arduino uno主板有3.3v的輸出電壓，可以解決這個問題。

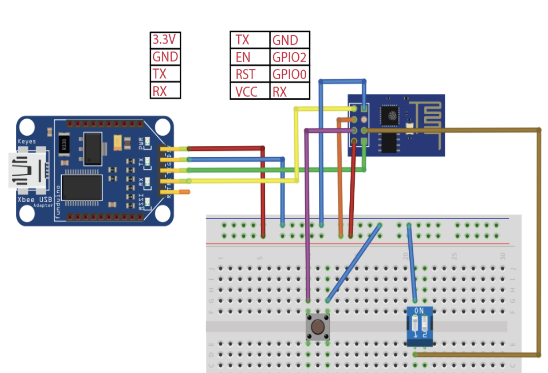
Esp01燒錄模式與運行模式的切換，只需將gpio 0 針腳導入ground即可切換，沒接ground時為正常模式，接入地線後即為燒錄模式。

圖 4傑森創工所提供方法

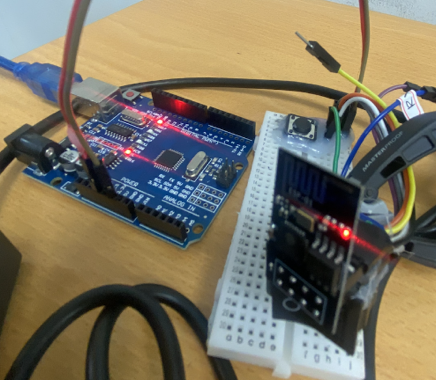
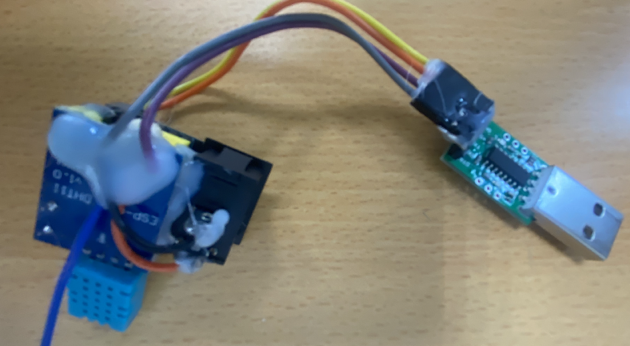
1. 第一版本：

圖 5自製esp01燒錄器-v1.0

此版本穩定性其實不是很理想，常常燒錄失敗，最主要原因在切換燒錄模式時，忘記reset，有時即便reset，也無法進入燒錄模式，必須將esp01斷電才能正常工作，由於需要uno提供3.3v電壓，arduino IDE上會顯示兩組usb port可以選擇，容易選錯連接埠，造成燒錄失敗。

1. 第二版本：

圖 6將tx, rx ,gpio0 gpio2四個針腳分別焊出

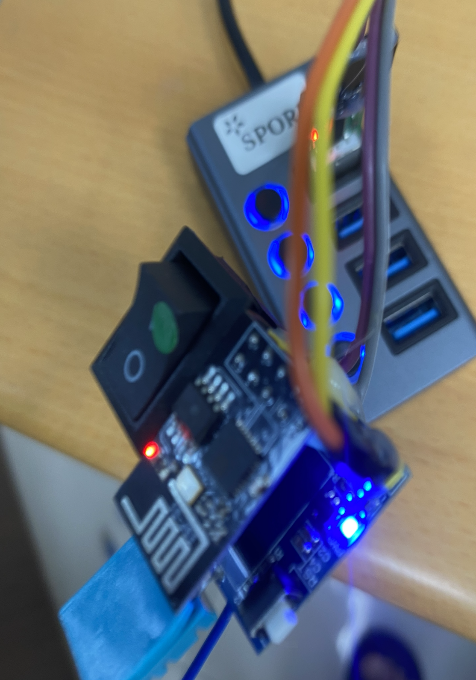
利用一組TBIOTMCM 來改良， TBIOTMCM連接esp01的腳位已經降壓為3.3v，因此只需準備一組usb 轉ttl 來連接TBIOTMCM 主板，將主板中tx,rx,gpio0拉出，分別接usb rx,tx,與ground即可。此版本，一方面解決兩個Arduino ide 上兩組usb 連線問題，也同時解決電壓供給的問題。

圖 7 v2.0運行狀態

1. **除錯機制：**
2. 序列埠監控

由於TBIOTMCM電路板，無usb 接口，無法利用序列埠監控來即時除錯，不過只要將tx, rx兩根針腳拉出，即可連接電腦，來監看esp01運行狀況。

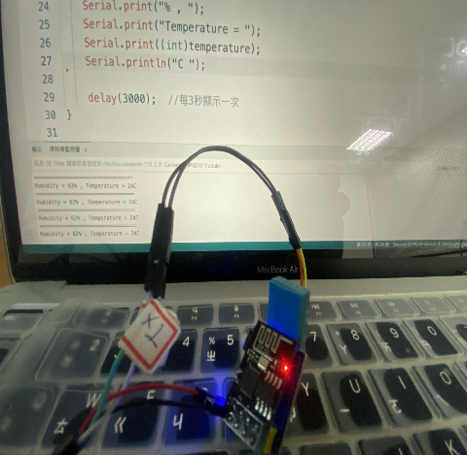


圖 8 焊接tx, rx兩根

圖 9 TBIOTMCM 除錯訊息

1. 從服務端監控

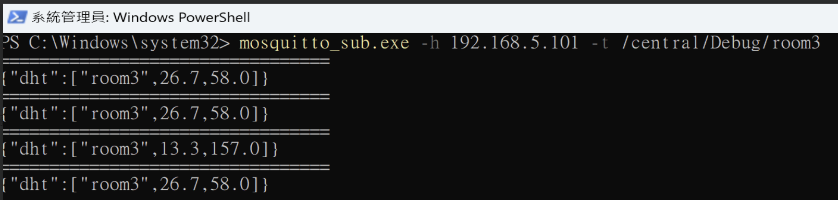
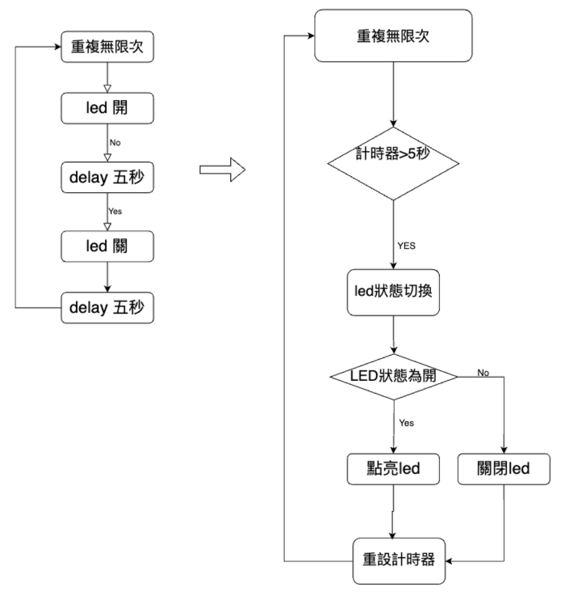
實驗設計要將uno 與esp01 串連，esp01中tx , rx 用於傳遞兩款主板訊息，一但進入整合狀態，就無法間看esp01執行過程。因此，我們將所有出錯訊息傳遞至網上，由服務端來監控並除錯。

圖11服務端訊息監控

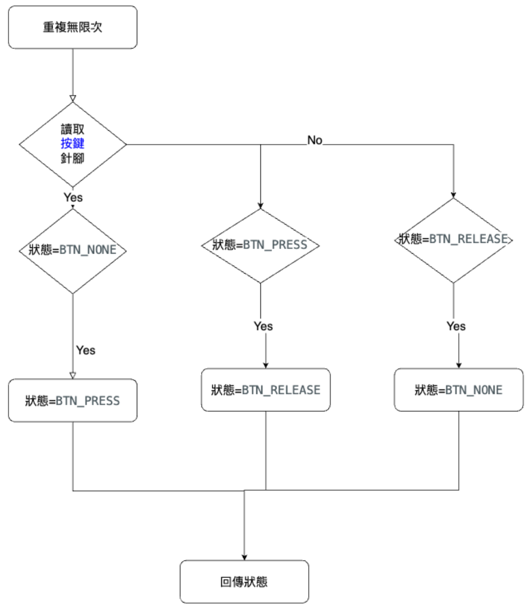
圖 10 除錯訊息流程

1. **設計功能模塊計時器：**
2. Led 燈號顯示狀態：

利用uno內建led，設計不同的閃爍頻率，可以用來提示，連上網路，連上mqtt 服務器，網路資料傳達等狀態。不過簡單閃爍功能是利用delay機制，這個做法初期優點可以快速實作，不過隨者整合功能增加，造成一些功能不穩定，舉例來說，在led閃爍期間，如果有按下安鍵動作，會造成失效問題。解決此問題，利用millis()函式，搭配設定的狀態機制，來取代delay函式。

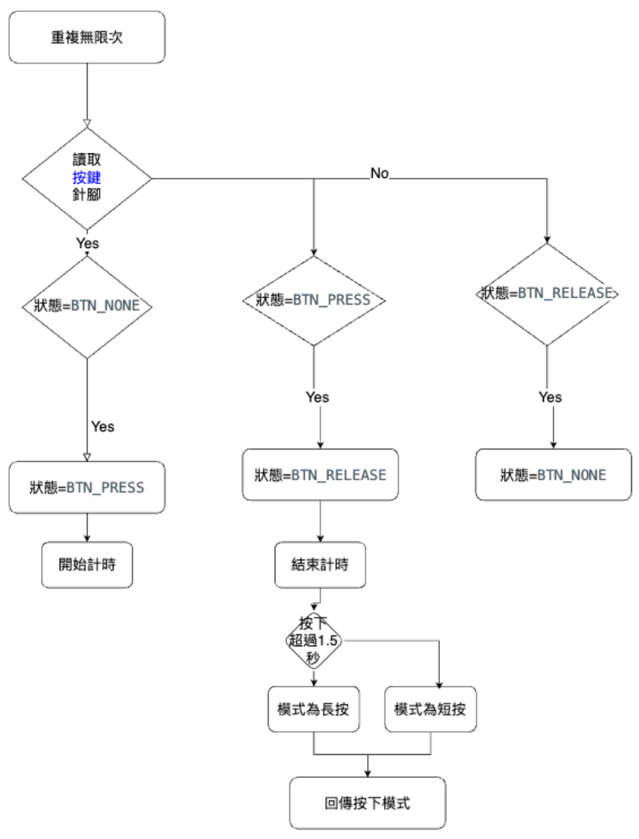
圖 12 led delay修改流程圖

1. 按鍵偵測：

在loop 函式中，利用digitalRead()函式，讀取按鍵針腳狀態，可以輕易做到利用按鍵來控制主板程式流程，不過，這裡同樣面臨一個問題，當按下時間過長，會觸發好幾次，造成功能無法如預被觸發。

設計三種狀態，分別表達，1.無動作（BTN\_NONE），2.按下(BTN\_PRESS)，3.放開(BTN\_RELEASE)。每一次按下，無論按下時間長短，收到BTN\_PRESS的回傳可能非常多，不過，BTN\_RELEASE只有一次，所以程式流程只需要依據這個值來判斷，就可避開重複觸發問題。

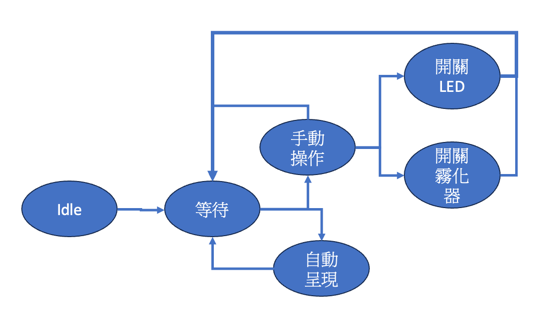
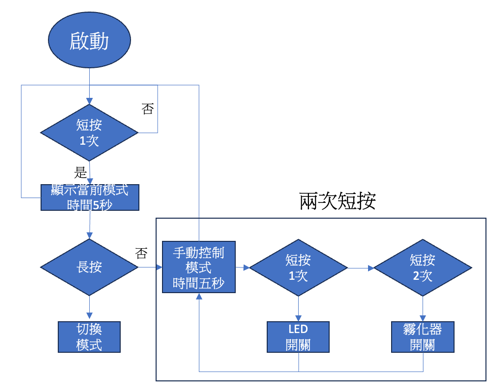
圖 13 按鍵控制流程圖

1. ****長按機制設計：

長按機制可以利用按鍵機制流程稍微修改即可完成。利用millis()函式在按下時，記住當下時間，並在方開時，計算所經歷時經，另用時間差，決定長按，或短按。

圖 14 長按設計流程圖

1. 按鍵功能：

設計三組按鍵實作在單一個uno上，此設計主要方面操作安裝在其他位置的uno，每一個房間的功能由單一顆按鍵控制，主要有三個功能需要實現，1.控制機電，2.抓取溫室度資訊。比較好的體驗是利用長按，與短按行為來區分與控制。

1. **主板串接：**

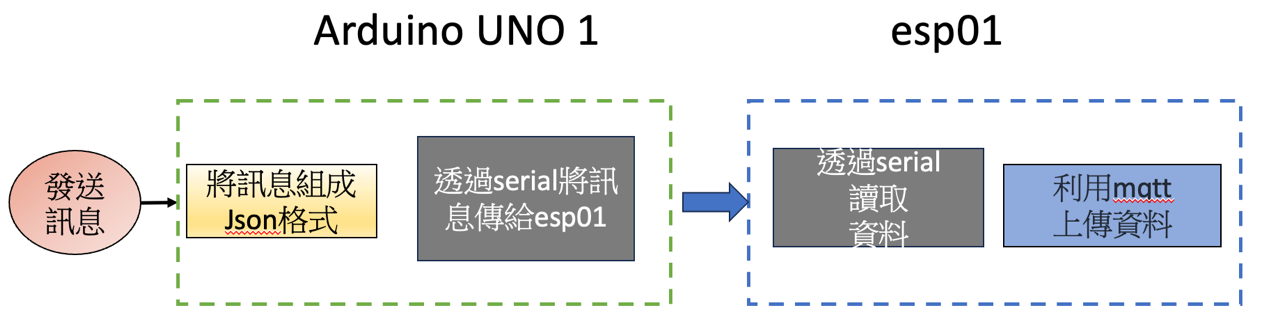
uno與esp01 主板串接是為了讓UNO也能夠連上網路，做到遠程操控uno上的電機。運行機制是利用softwareSerial模擬序列通訊，讓 Arduino 可以透過軟體方式建立額外的 UART (串列通訊) 連接，這樣可以在uno上保留原始除錯模式，也可以直接燒錄程式，如果直接利用硬體序列埠來完成此功能，會有兩個問題:1.燒錄時，必須移除此連接，才能正常上傳程式，2.原來的除錯訊息，也需要移除，。

圖 2 uno 與esp01 串接圖

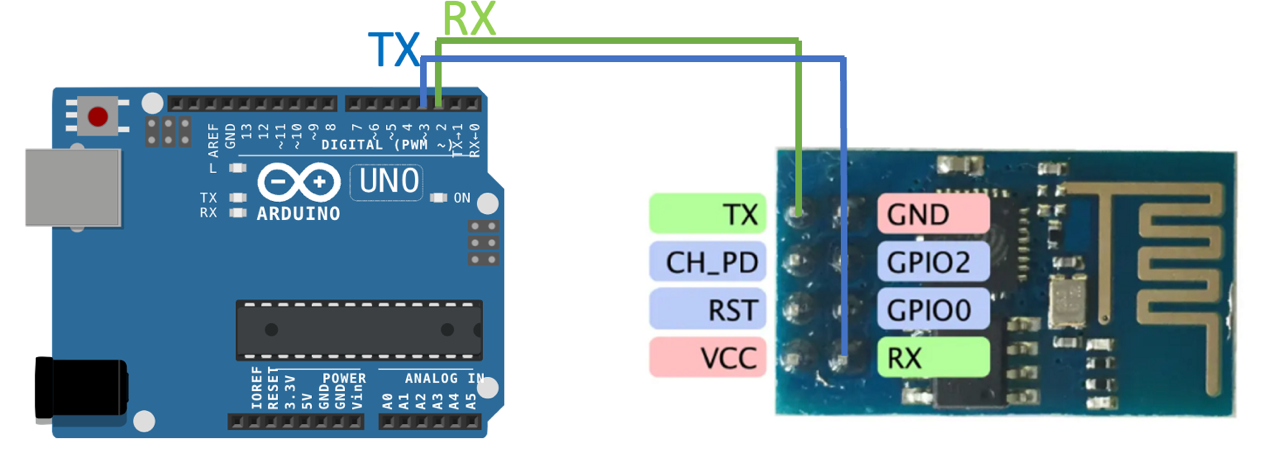
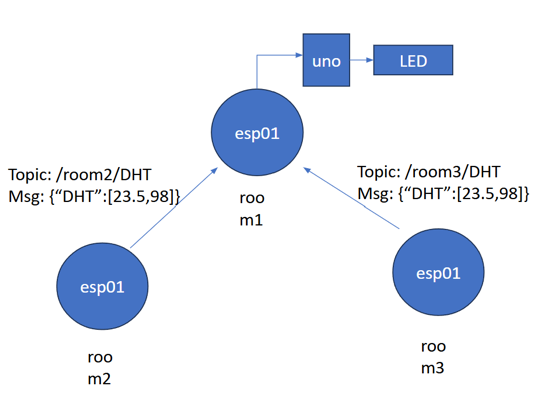


圖 2 uno 與esp01 資料傳遞

1. **將資訊由lcd呈現**

實驗設計理念，是只安裝一個LCD，並且此lcd必須要能

呈現每一個房間溫濕度。因此安裝在room1的esp01需要訂閱來至room2 以及 room3 的主題，這樣就可在此LCD上現示三房間溫濕度資訊

1. 遠程操作。

利用node-red快速製作網頁互動介面

