

桃園市第 64 屆中小學科學展覽會
初賽作品說明書

科 別：生活與應用科學科(一)

組 別：國小組

作品名稱：智能種子孵化器-植物種子發芽的環境監控調節

關 鍵 詞：Arduino 開發板、感測器、程式

智能種子培養孵化器

-植物種子發芽的環境監控調節

摘要

伴隨科技進步，AI 時代的來臨，智能農業發展已成為當今農業領域的一個重要趨勢，而針對傳統農業進行革新和升級，提高農業生產效率、降低生產成本、保護環境和天然資源等方面的技術的提升，勢必是實現農業現代化的重要途徑。

此次研究實驗的主要範圍，在物聯網(IoT)的實驗場域建置，透過 Arduino 開發板，連接傳感器，並將設備部署在種子培養皿中，以達到數據收集，濕度控制，光照調節等功能。此外，為實現遠程控制，縮短開發時程，本實驗透過 MQTT 通訊協議來完成數據發送功能，再搭配 node-red 建置，短時間內便能整合相關功能，建置網頁，達到遠程控制機制。

壹、前言

一、研究動機：

在奶奶家有一塊小小的菜園，每次跟媽媽回奶奶家，都對園裡的植物生長充滿好奇。有一次想起在學校自然課程中曾學習到植物種子發芽的條件，於是想透過現代科技自動化的設備，有效率且立即來觀察監控種子發芽過程。或許可以透過紀錄植物種子生長環境的溫濕度，能即時調節適於種子發芽過程的環境空間。因此自己想藉由 Arduino 開發板，整合相關感測器，建構一個讓植物種子發芽的最佳環境空間。

二、研究目的：

透過本實驗建置過程，學習物聯網的系統架構，並且動手完成各項功能，其中包含：

(一)遠端監控：

利用 node-red 可以即時監控現在的土壤濕度及空氣中的溫濕度，也可以遠端控制讓機器啟動。

(二)建置自動開關：

當機器檢測到空氣中溫度過高，會自動啟動霧化器來降溫；濕度過低，也會啟動霧化器來提升濕度；檢測到天色變暗了，會自動啟動 LED 燈條來增加光照；當檢測到土壤濕度過低，澆水系統會自動啟動來提高土壤濕度以免植物死掉。

(三)高靈敏度：

準確探測空氣中的溫濕度和土壤中濕度，而且當機器感測到水位過低時，機器會自動幫瓶子加水

(四)科學性，實際運用：





透過實驗培養對科學過程的態度，例如：觀察、提問、搜尋資料。

(五)教學性：

使用土壤感測器、溫濕度感測器等，將偵測訊號傳送 node-red，利用現有的通訊協議，收集相關資訊，以便之後方便查看。

貳、 研究設備及器材：

一、 硬體設備：

			
Arduino UNO	ESP8266	MOS 驅動板	霧化器主板
			
人體感測器	杜邦線	LED 燈條	霧化器
			
馬達控制器	電池盒	水汞	溫濕度感測器
			
鋰電池	按鈕	LCD	擴充板
其他器材和材料： 絕緣膠帶、剝線鉗、剪刀、鑽洞器、焊槍、熱熔槍、手機、土壤濕度感測器、USB 傳輸線、螺絲、螺絲起子、三用電表、塑膠盒、老虎鉗、水位感測器、種子培養盆、水管、開關、芹菜種子、青蔥種子， Wifi 分享器，MacBook Air。			

二、軟體開發環境：

（一）服務器端：

Homebrew 套件管理系統，Mosquito MQTT 消息代理，node-red 圖形化程式編輯

（二）客戶端：

Arduino IDE 整合開發環境，Scratch。

參、研究過程或方法:

本實驗包含軟體與硬體的建置，以及初期認識系統架構，後期實作設計兩部分，第一階段，利用 Scratch 來做初步模擬，透過積木程式，了解整個系統框架，伴隨軟體能力的提升後，將開發平台移至 Arduino IDE 環境，在 C++的程式學習中，體驗程式語法，以及網路資源利用。此外，為降低服務器端建構的難度，使用 **mosquitto** 搭配 **node-red** 完成遠程監控機制。

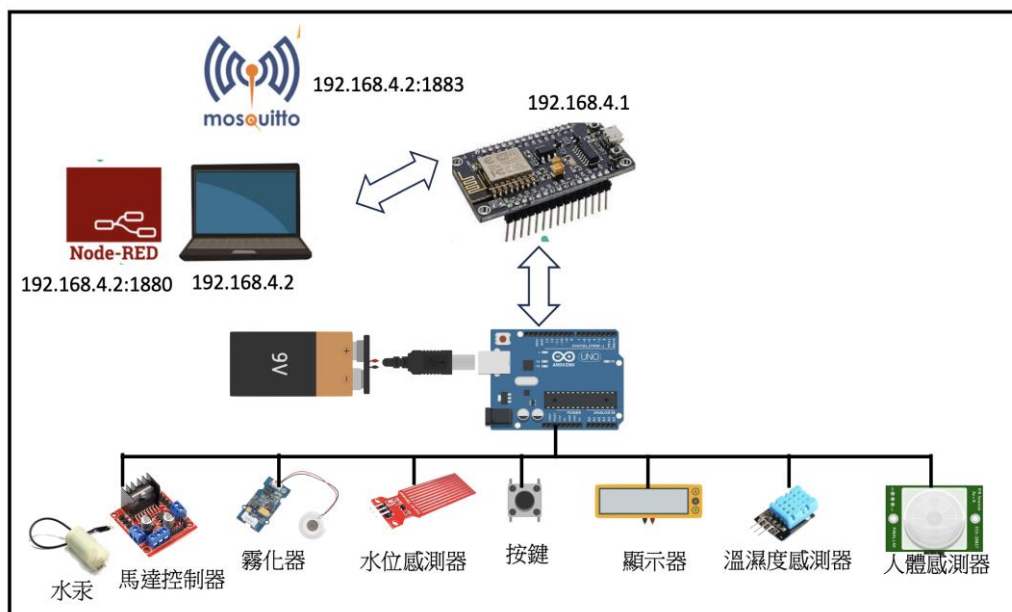


圖 1:系統架構

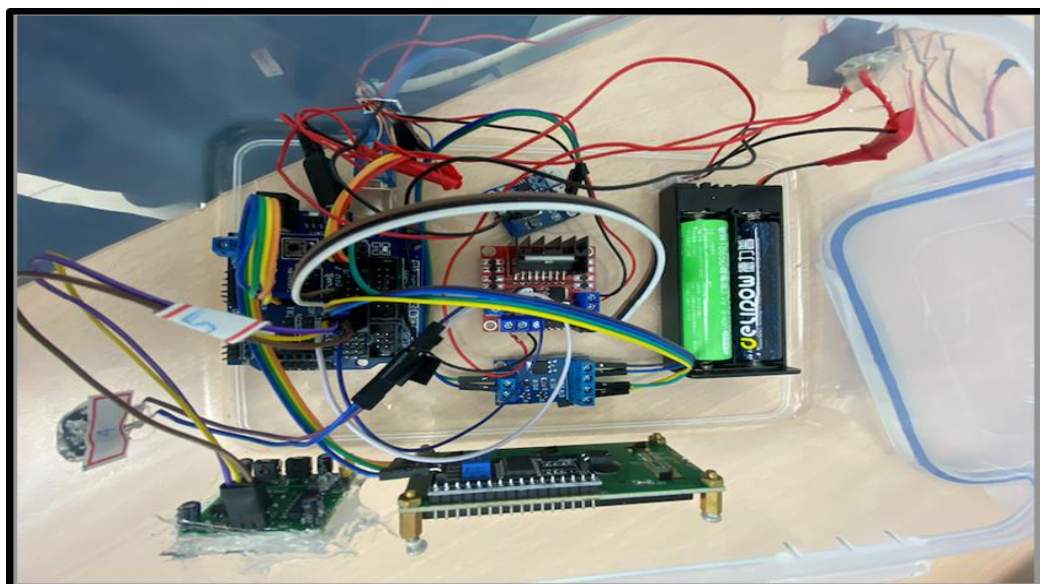


圖 2:裝置整合

一、Scratch 模擬器實作：

利用 Scratch 角色特性，快速模擬各個功能模塊，再利用廣播訊息提供模塊連結，初步串接整個系統。

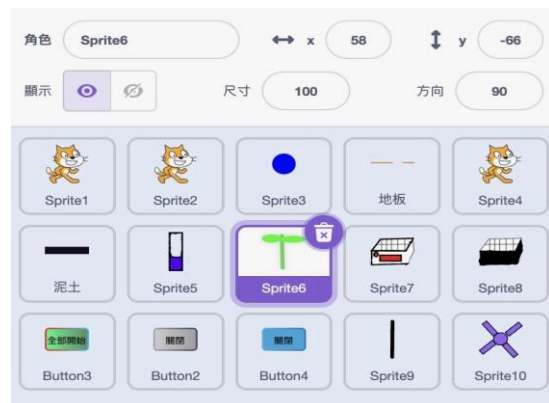
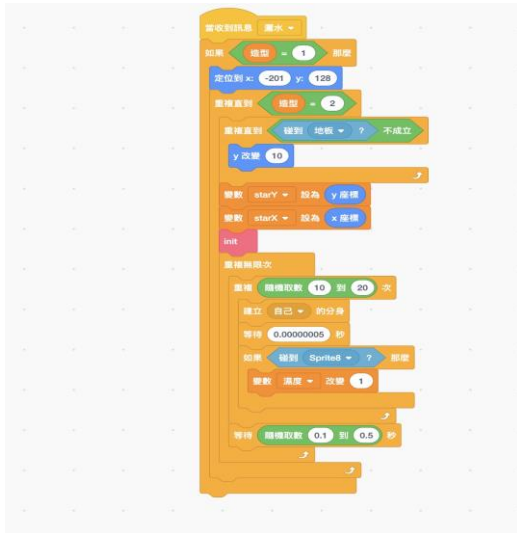


圖 3：訊息廣播

圖 4：角色模擬

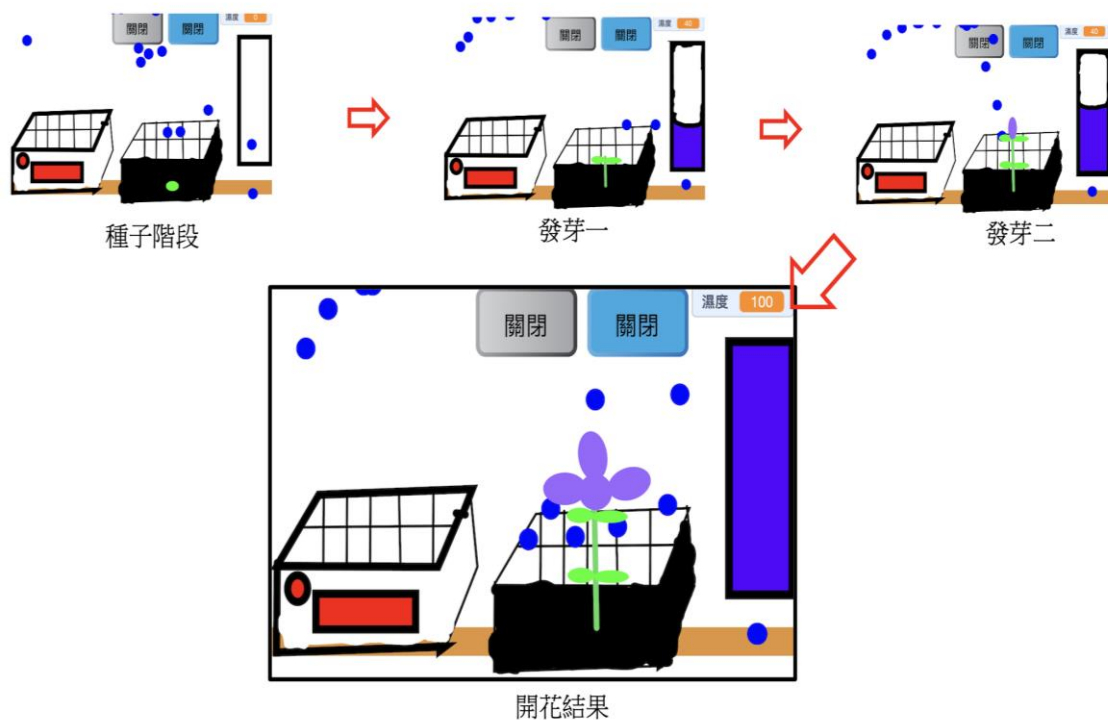


圖 5：Scratch 模擬結果

二、感測器，機電整合，與測試實驗。

本實驗採用 v-模型軟體建置方法，來整合多項感測器與機電，並確保各個階段的功能完整。

(一)主板連接

1.硬體組裝：

採用兩款 Arduino 開發板，分別為 UNO 與 ESP8266。其中 UNO 負責連接控制各個感測器與機電操作，由於 UNO 本身未提供無線傳輸功能，為實現遠程操控，必須再加上 Wifi 模組，讓兩個開發板，能相互傳送訊息。

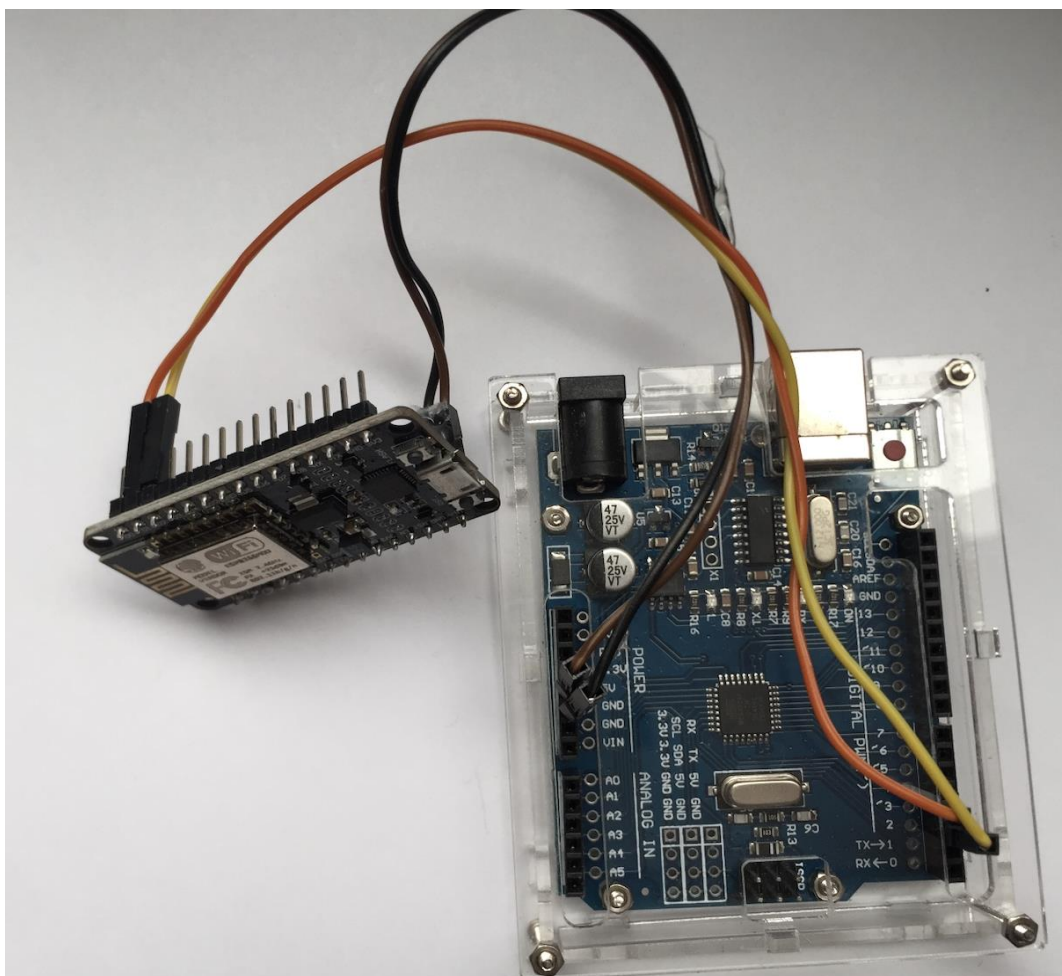
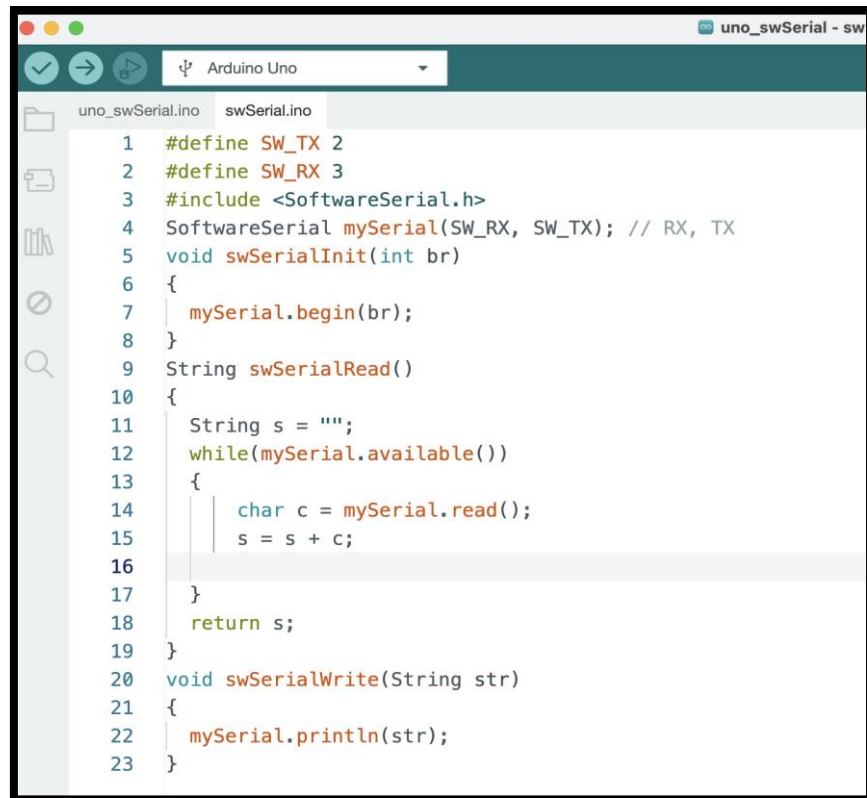


圖 6：Arduino UNO 與 ESP8266 通訊連接

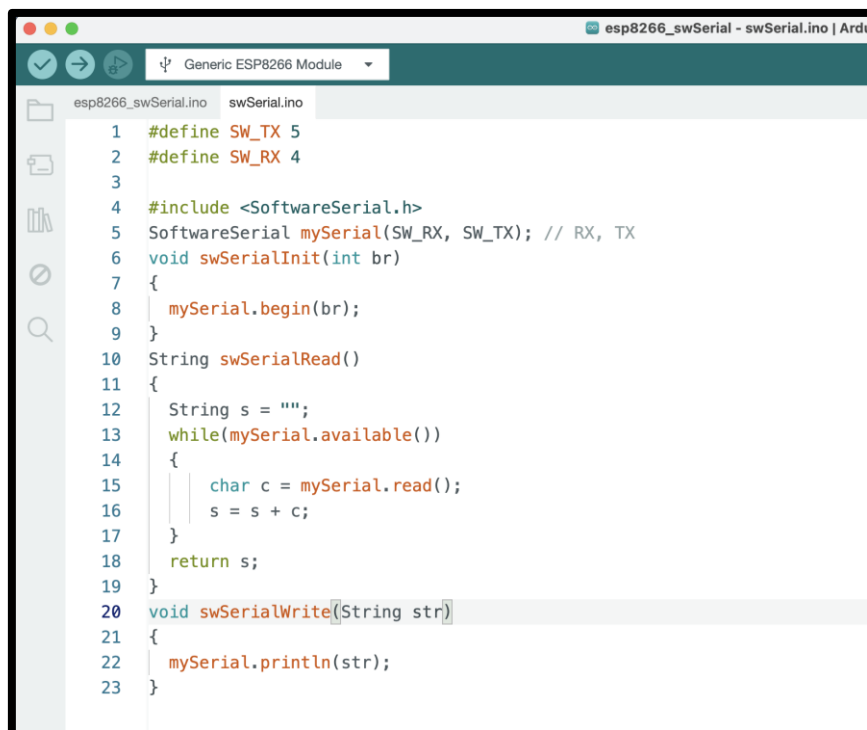
2.軟體設計：

因應不同的主板，需撰寫兩份程式，再分別燒錄到個別板子。



```
1  #define SW_TX 2
2  #define SW_RX 3
3  #include <SoftwareSerial.h>
4  SoftwareSerial mySerial(SW_RX, SW_TX); // RX, TX
5  void swSerialInit(int br)
6  {
7      mySerial.begin(br);
8  }
9  String swSerialRead()
10 {
11     String s = "";
12     while(mySerial.available())
13     {
14         char c = mySerial.read();
15         s = s + c;
16     }
17     return s;
18 }
19
20 void swSerialWrite(String str)
21 {
22     mySerial.println(str);
23 }
```

圖 7：Arduino UNO softwareSerial 連線程式



```
1  #define SW_TX 5
2  #define SW_RX 4
3
4  #include <SoftwareSerial.h>
5  SoftwareSerial mySerial(SW_RX, SW_TX); // RX, TX
6  void swSerialInit(int br)
7  {
8      mySerial.begin(br);
9  }
10 String swSerialRead()
11 {
12     String s = "";
13     while(mySerial.available())
14     {
15         char c = mySerial.read();
16         s = s + c;
17     }
18     return s;
19 }
20 void swSerialWrite(String str)
21 {
22     mySerial.println(str);
23 }
```

圖 8：Arduino **ESP8266** softwareSerial 連線程式

透過軟體接口(SoftwareSerial)完成此功能，之所以不採用硬體接口(HardwareSerial)

來完成對接，是因為每次燒錄程式時，必須拔除連接針腳，否則會導致程式上傳失敗。

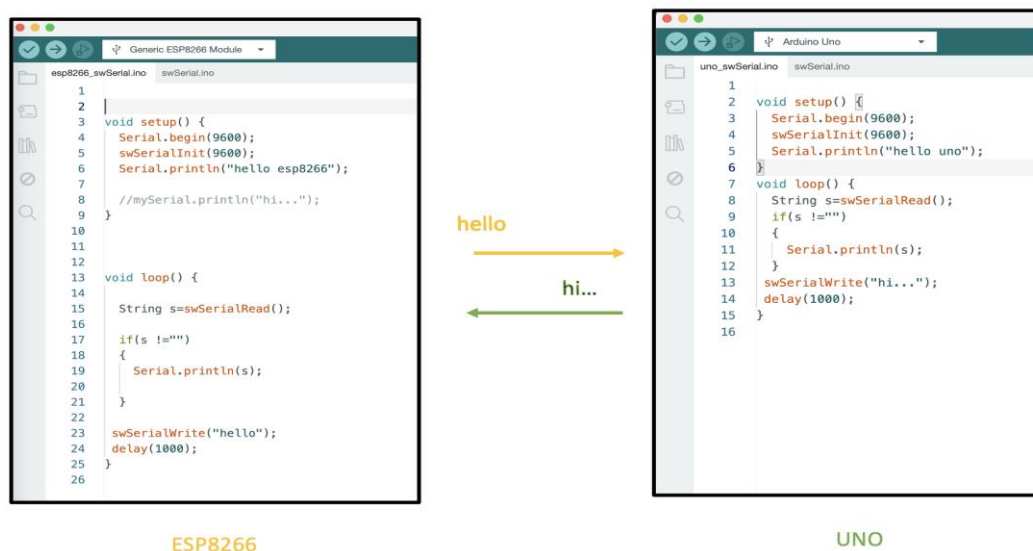


圖 9：UNO 與 ESP8266 訊息傳送程式

(二)感測器連接：

1.溫濕度感測器：

DHT11 是一個可以同時測試溫濕度的感測器。只須接入數位針腳，搭配軟體，即可抓取數據。

(1)硬體組裝：

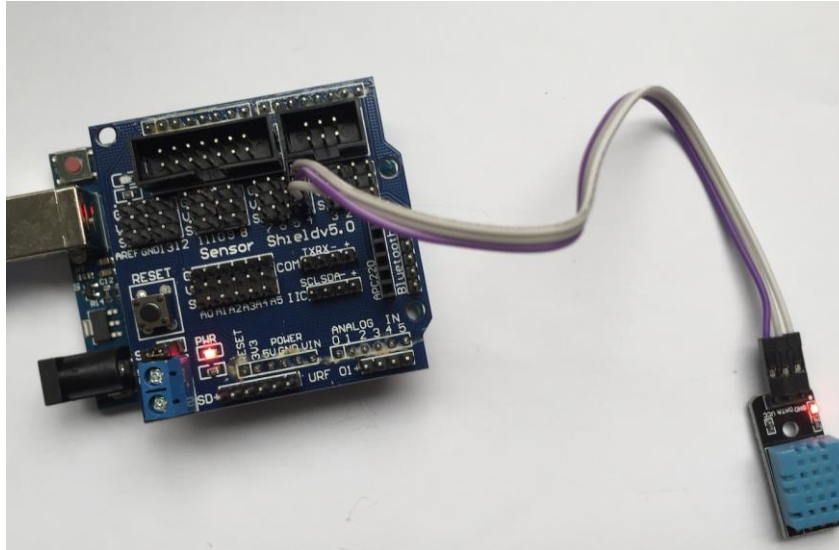


圖 10：Arduino UNO 主板連接溫濕度感測器的圖片

(2)軟體設計：

DHT11 感測器利用了 simpleDHT 這包函式庫。利用的函式庫裡頭的 read()函式即可抓取所需數值。

```

DHT11.ino
1  #include <SimpleDHT.h>
2  int pinDHT11 = 4;
3  SimpleDHT11 dht11;
4  void setup() {
5      // put your setup code here, to run once:
6      Serial.begin(9600);
7  }
8
9  void loop() {
10     // put your main code here, to run repeatedly:
11     byte temperature = 0;
12     byte humidity = 0;
13     int err = SimpleDHT11::read(pinDHT11, &temperature, &humidity, NULL);
14     // start working...
15     Serial.println("=====");
16     if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHT11::SUCCESS) {
17         Serial.print("Read DHT11 failed, err="); Serial.println(err); delay(1000);
18         return;
19     }
20     Serial.print("Humidity = ");
21     Serial.print((int)humidity);
22     Serial.print("% ");
23     Serial.print("Temperature = ");
24     Serial.print((int)temperature);
25     Serial.println("C ");
26
27     delay(3000);
28 }
29

```

圖 11：溫濕度傳感程式

2.LCD 顯示器：

將溫濕度資訊即時呈現在 LCD 顯示器，在實驗初期，並未整合雲端功能，必須要手動記錄每個裝態，有了此項功能，就無須每次帶上電腦，就能完成溫濕度紀錄。

(1)硬體組裝：

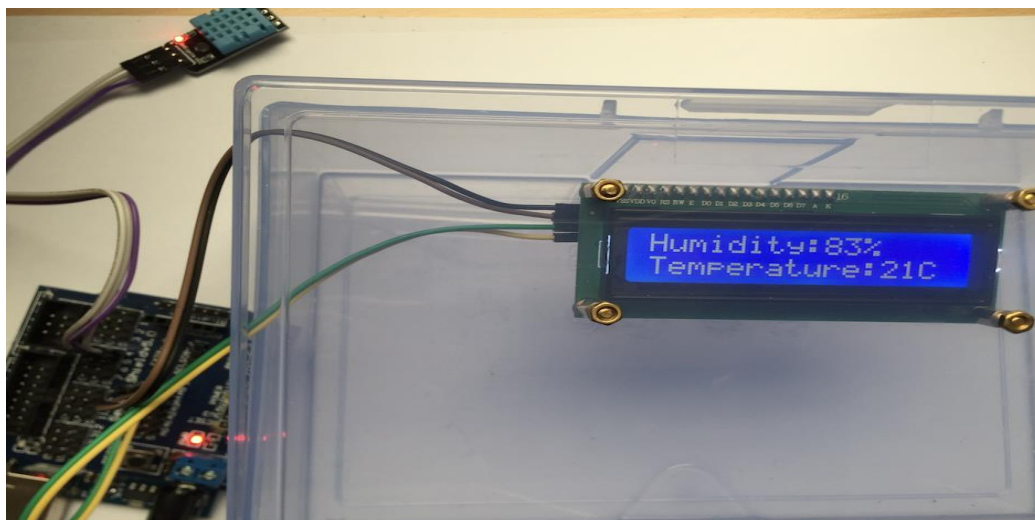


圖 12：LCD 顯現溫濕度的數值

(2)軟體設計：

```
lcd_dht11 | Arduino IDE 2.3.2
Arduino Uno
lcd_dht11.ino
12 void DHTtoLED() {
13   // put your main code here, to run repeatedly:
14   byte temperature = 0;
15   byte humidity = 0;
16   int err = SimpleDHTerrSuccess;
17   // start working...
18   Serial.println("=====");
19   if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTerrSuccess) {
20     Serial.print("Read DHT11 failed, err="); Serial.println(err); delay(1000);
21     return;
22   }
23   String T(temperature);
24   String H(humidity);
25   lcd.setCursor(0, 0);
26   lcd.print("Humidity:");
27   lcd.print(H);
28   lcd.print("%");
29   lcd.setCursor(0, 1);
30   lcd.print("Temperature:");
31   lcd.print(T);
32   lcd.print("C");
33
34   Serial.print("Humidity = ");
35   Serial.print((int)humidity);
36   Serial.print(" , ");
37   Serial.print("Temperature = ");
38   Serial.print((int)temperature);
39   Serial.println("C ");
40 }
```

圖 13：LCD 整合 DHT11 程式

3.人體感測器:

透過人體感測器，偵測當有人進入實驗場域時，才點亮 LCD，一方面延長 LCD 的壽命，另一方面也能更省電。

(1)硬體組裝：

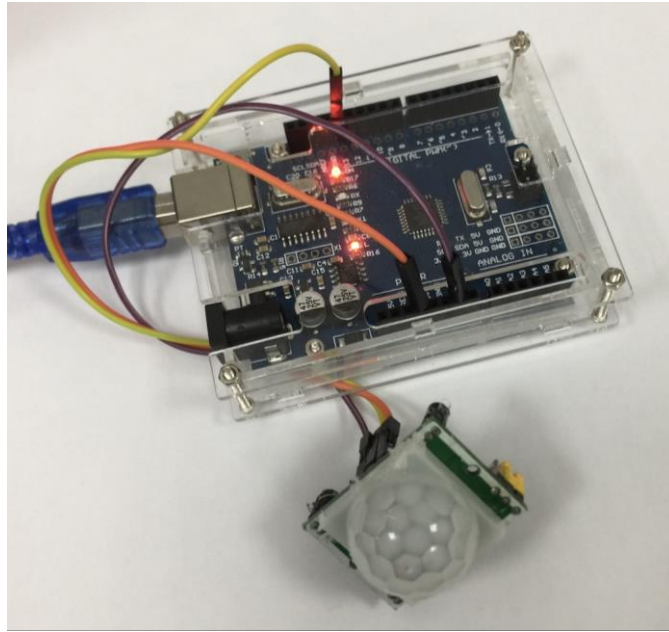


圖 14：人體感測器

(2)軟體設計：

```

1 int PIR_Pin = 3;
2 void setup() {
3   // put your setup code here, to run once:
4   pinMode(PIR_Pin, INPUT);
5   pinMode(13, OUTPUT);
6   Serial.begin(9600);
7
8
9
10 void loop() {
11   // put your main code here, to run repeatedly:
12   int state = digitalRead(PIR_Pin);
13   if(state == 1)
14   {
15     digitalWrite(13, HIGH);
16     Serial.println("----detected on----");
17   }
18   else
19   {
20     digitalWrite(13, LOW);
21     Serial.println("No detect");
22   }
23   delay(100);
24 }
25

```

圖 15：人體感測程式

```

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on '/dev/cu.usbserial-110')

No detect
No detect
----detected on----
----detected on----
----detected on----
----detected on----
----detected on----
----detected on----
----detected on----
----detected on----
No detect
No detect
No detect
No detect
No detect
No detect
No detect
No detect
No detect

```

圖 16：人體感測器測試數據

4.馬達控制

(1)硬體組裝：

單獨使用繼電器控制，只能控制開與關，無法實現水量細微控制，，因而增加馬達控制器，並透過 PWM 來完成此功能。

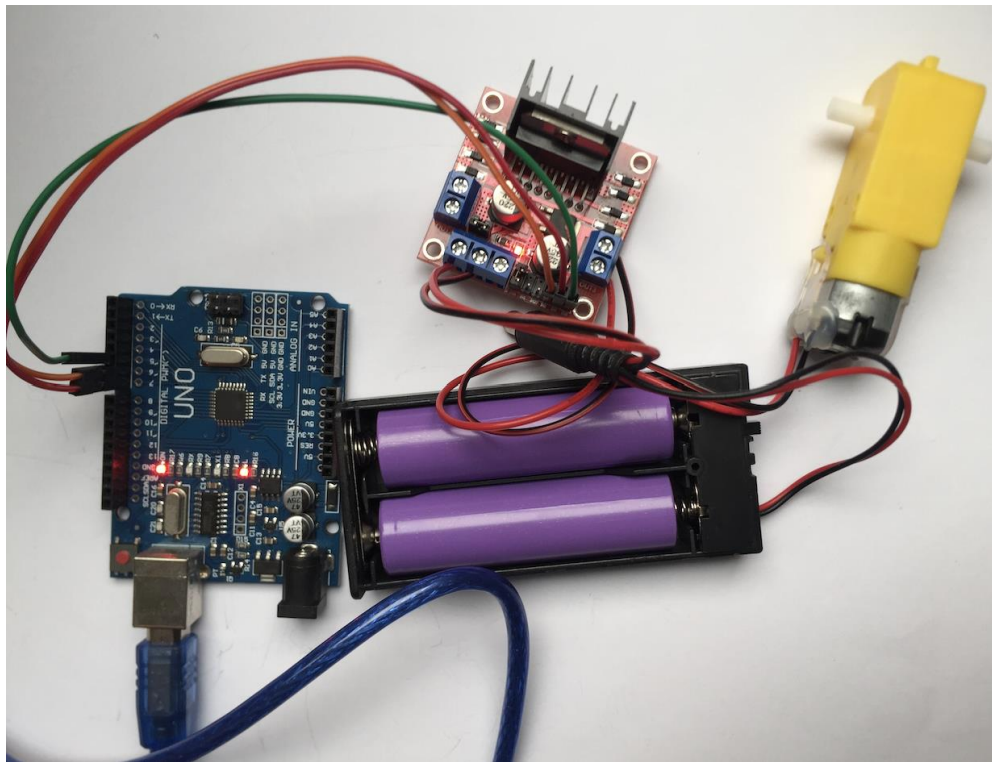


圖 17：UNO 整合馬達控制板

(2)軟體設計：

```
l298n-motor.ino
1  int IN1=8;
2  int IN2=9;
3  int pwm1 = 7;
4
5
6
7  //-----
8  void setup() {
9      Serial.begin(9600);
10     pinMode(IN1, OUTPUT);
11     pinMode(IN2, OUTPUT);
12     pinMode(pwm1, OUTPUT);
13     control2();
14 }
15
16 //-----
17 void loop() {
18
19
20 }
21 //-----
```

圖 18：PWM 控制馬達輸出

5. LED

LED 燈條主要是為一方面提升植物光照時間，另外也是比較省電的裝置。

(1)硬體組裝：

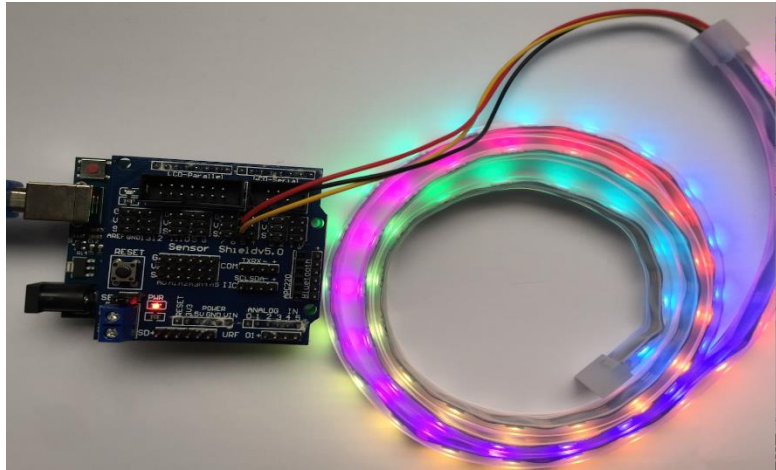


圖 19：UNO 連接 LED

(2)軟體設計：

FastLED 函式庫來實現 led 顏色控制，FastLED 初始化要將 gpio 針腳與 LED 陣列提供給函式庫，LED 陣列所儲存的值代表每一顆 LED 的顏色，關閉 LED 的方法，則直接將 LED 值設為 black。透過 FastLED.show()函式 來實現對狀態改變。



圖 20：LED 燈光控制

6.霧化器與水位偵測器:

除了開起與關閉霧化器功能以外，還希望能做到改變噴出霧量多寡，為實現這項功能，必須在外接一個電機控制板，透過 PWM 來完成此功能。

(1)硬體組裝：

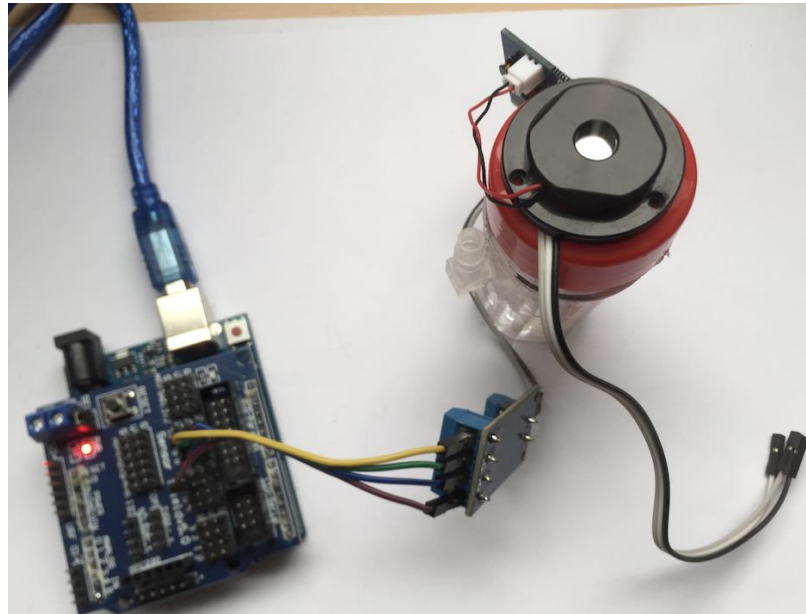


圖 21 霧化器，水位感測器整合

(2)軟體設計：

```

atomizer.ino
1
2 int pwm3 = 6;
3 //-----
4 void setup() {
5     Serial.begin(9600);
6
7     pinMode(pwm3, OUTPUT);
8     analogWrite(pwm3, 250);
9 }
10 //-----
11 void loop() {
12
13 }
14

```

圖 22：PWM 控制霧化器。

```

waterlevelsensor.ino
1 int water_sensor=0;
2 int val=0;
3 void setup()
4 {
5     Serial.begin(9600);
6 }
7 void loop()
8 {
9     val=analogRead(water_sensor);
10    Serial.println(val);
11
12    delay(100);
13 }
14

```

圖 23：讀取水位感測器數值

(三)軟體整合：

硬體功能的銜接都是透過杜邦線來完成，常常會有掉線問題發生，或程式碼整合過程中，出現改錯現象，一時不知是軟體還是硬體問題，因此，當我們將所有功能模塊，整合到同一個 Arduino 專案過程中，必須能夠單獨驗證每一個功能，以提

供方便快速驗證功能機制。

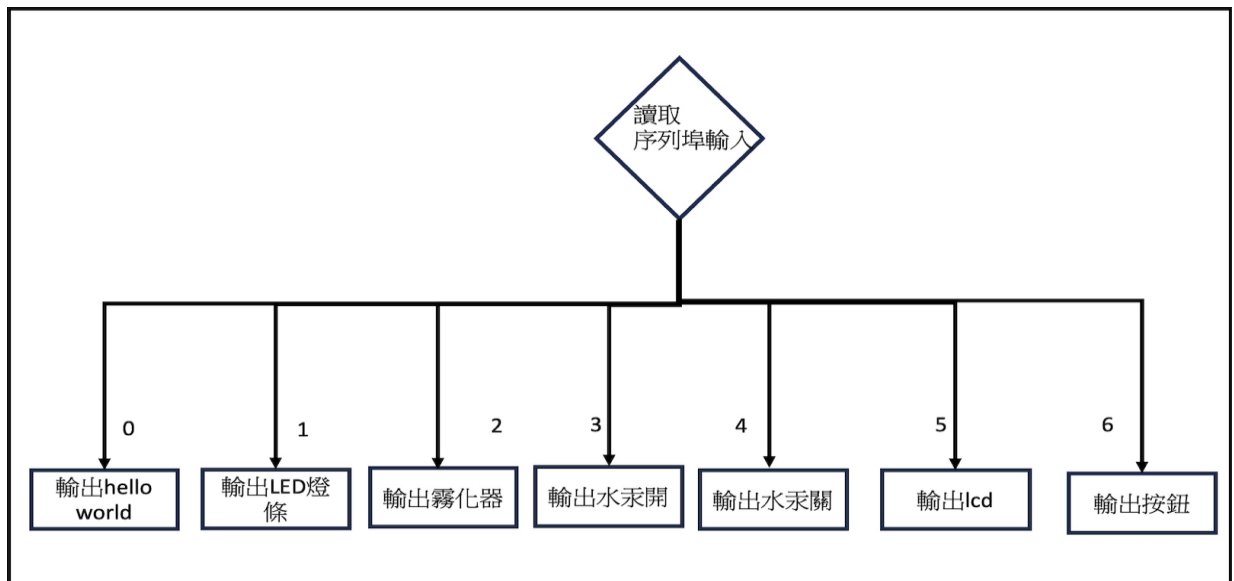


圖 24：利用序列埠輸入，實現單元測試

```
1 String s="-----\n\n\n2 0:hello world\n\n3 1:led\n\n4 2:atomizer\n\n5 3:openmotor\n\n6 4:closemotor\n\n7 5:lcd\n\n8 6:btn\n\n9 7:pir\n\n10 8:wls\n\n11 9:dht\n\n12 -----";\n13 void setup() {\n14   testcase_setup();\n15   Serial.println(s);\n16 }\n17 void loop(){\n18   testcase_loop();\n19 }
```

The screenshot shows the Arduino IDE 2.3.2 interface with the 'main.ino' file open. The code defines a string 's' containing a list of test cases, each with a number and a description. The 'setup' function calls 'testcase_setup()' and prints the string 's'. The 'loop' function calls 'testcase_loop()'.

圖 25：單元測試程式

(四)服務器整合與 node-red 程式串接:

1.設定/燒錄 ESP8266 相關程式：

第一步：設定 MQTT 服務器所在 ip

```
const char* ssid = APSSID;
```

```
const char* password = APPSK;
```

```
const char* mqtt_server = "192.168.4.2";
```

第二步：將 ESP8266 設置為 Wifi 分享模式：

```
WiFi.softAP(ssid,password);
```

第三步：設置 MQTT 廣播/訂閱主題。

```
client.subscribe("inTopic");
```

```
client.publish("outTopic", "hello world");
```

2. 在 node-red 流程介面，可以快速撰寫簡單測試來驗證 mosquitto service 是否正常運行。

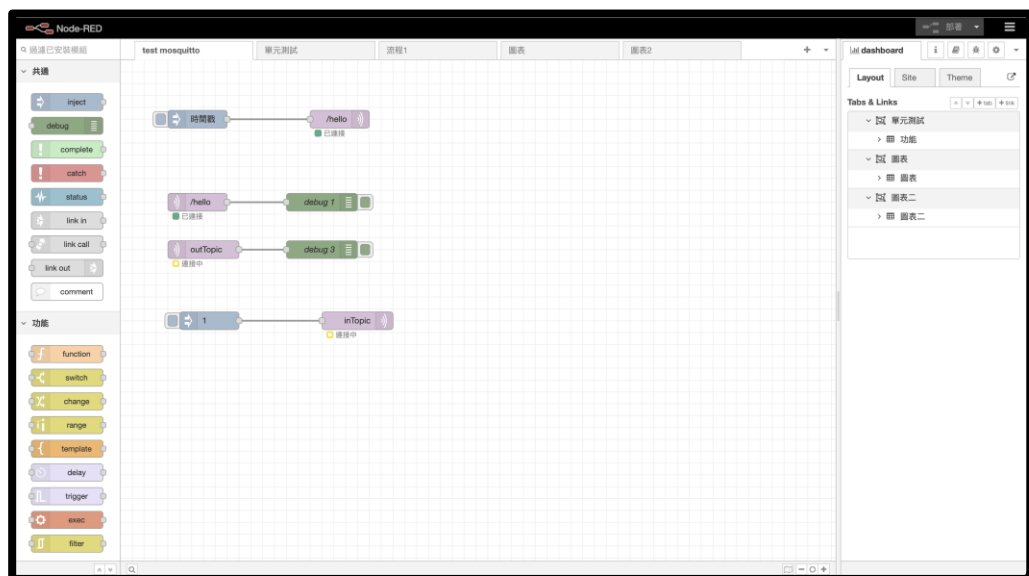


圖 26：利用 node-red 測試連線

3. 在 node-red 撰寫功能模塊測試程式：

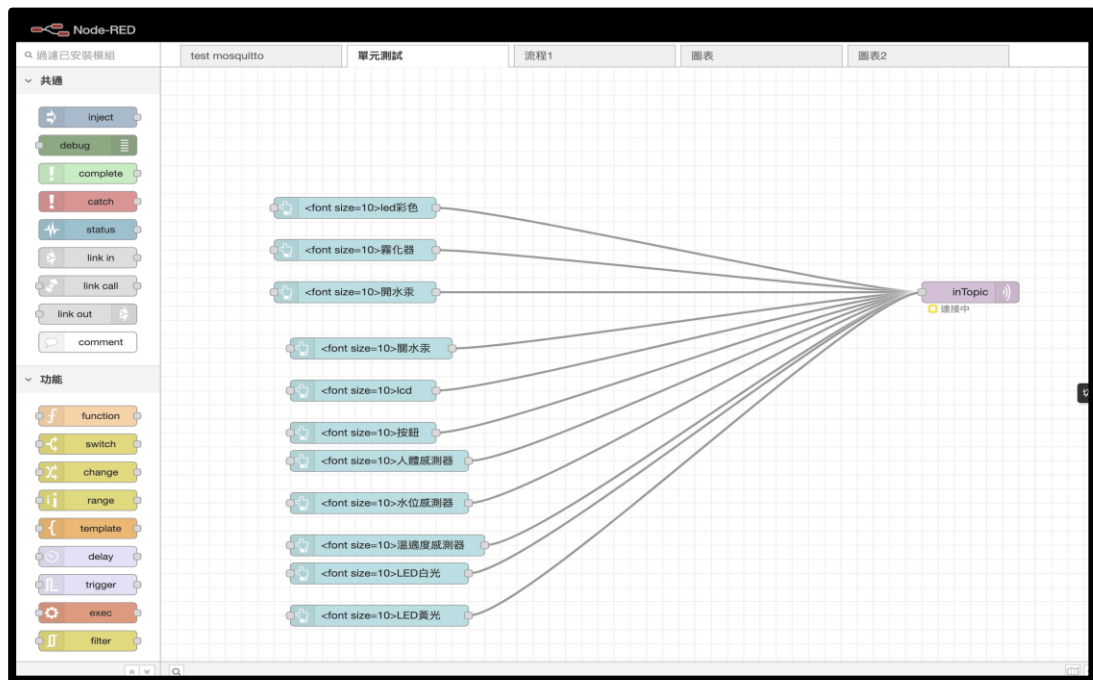


圖 27：利用 node-red 單元測試程式

4.新增網頁操作介面：



圖 28：利用 node-red 網頁操作介面

5.將傳感器資訊呈現在網頁上：

(1)累積數據收集：

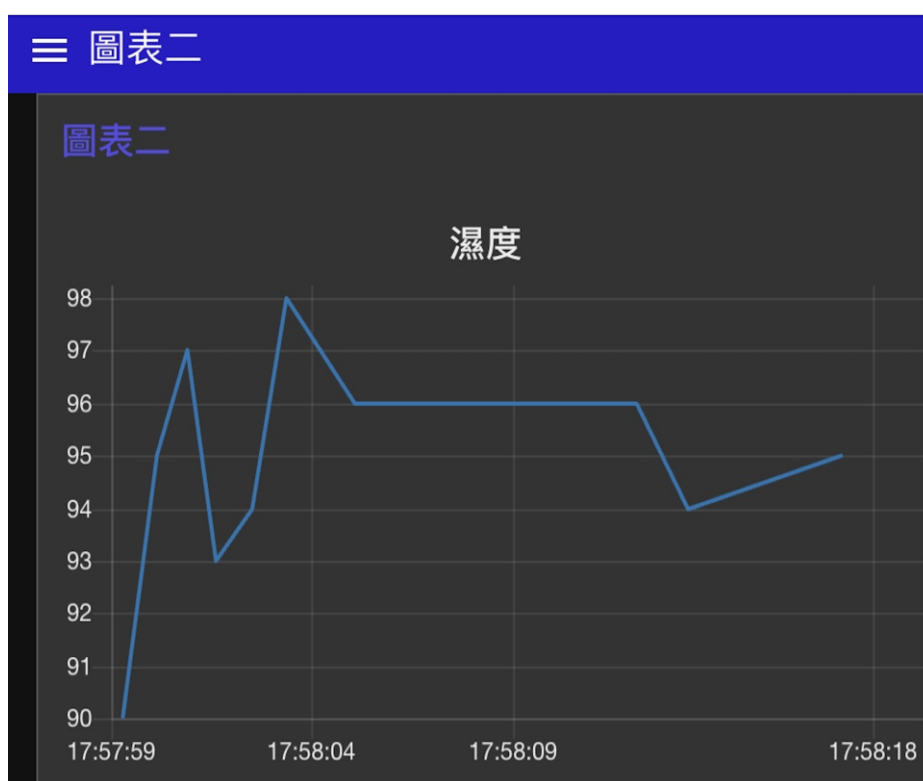


圖 29：利用 node-red 網頁數據呈現介面(一)

(2)即時溫濕度數據呈現：

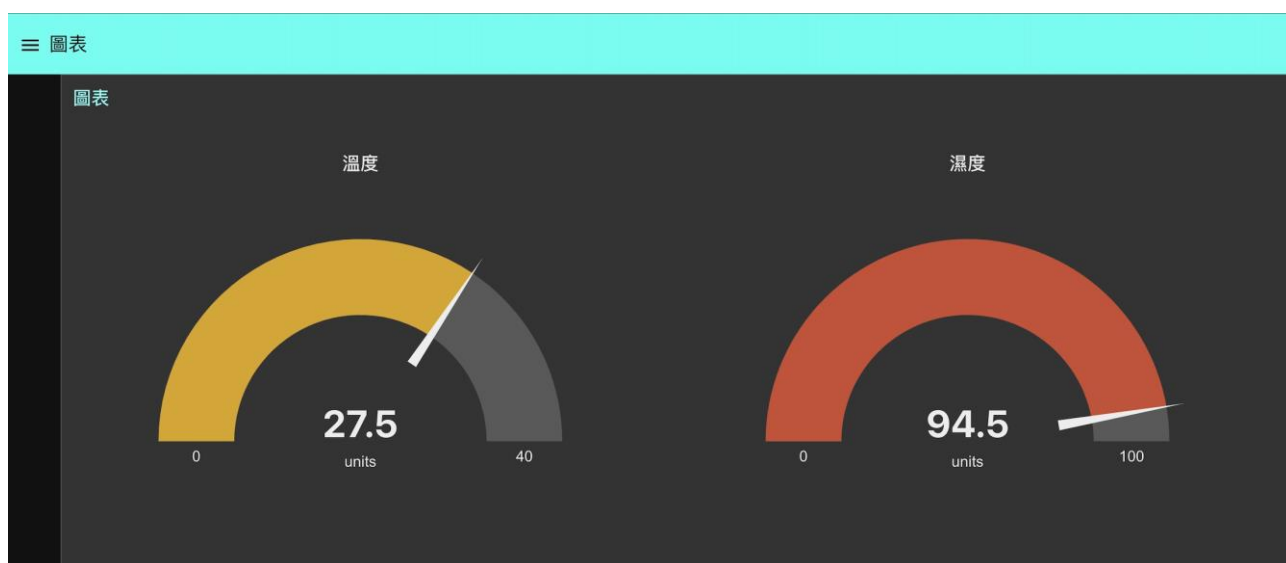


圖 30：node-red 網頁數據呈現介面(二)

肆、 研究結果

一、種子發芽紀錄：

第一階段，觀察並記錄每日種子發芽情況，比較調整濕度與光照時間後，與實驗組做比較。



圖 31：發芽紀錄

二、智能裝置安裝：

為避免短路，將所有跟水相關的電子元件，重新規劃，佈線。



圖 32：霧化器，與灑水器安裝

伍、討論

一、感測器整合問題：

問題 1：UNO 與 ESP8266 資料無法傳達？

答：序列埠傳送，在初期遇到速率不一致，產生亂碼，如果完全沒有訊息，則需確定 UNO 的 TX 針腳，接到 ESP8266 的 RX 針腳。

問題 2：如何將溫度與濕度得數值呈現在 LCD 上？

答：LCD 採用 I2C 的通訊，軟體驗證前，必須要將 SCL 與 SDA 分別接上 A4 與 A5。本實驗利用 LiquidCrystal_I2C 函式庫，本實驗利用 LiquidCrystal_I2C 函式庫，print()函式，可以將資料呈現，不過換行功能則需要使用 setCursor()來完成。

問題 3: 霧化器發出異聲？

答：霧化器透過海綿來吸取水分，達到霧化效果。海綿沒有緊密接觸時，才會發出異聲。

問題 4: 水泵無法正常運作？

答：本實驗使用馬達電壓要求比較高，需高於 5V 電壓才能讓水泵正常運作，另外 PWM 值不能低於 150 否則無法正常抽水，為避免電壓不穩，採用外部電池單獨供電。

問題 5: 土壤濕度偵測器無法正常運作？

答：本實驗，土壤偵測感應器，經過一段時間後，會出現氧化現象，就必須更換新的感測器。

問題 6: 人體感測器不靈敏？

答：人體感測器有兩組可變電阻，調整後問題及獲得改善。

問題 7: Arduino IED 無法複製貼上？

答：更新 IDE 2.3.2 版本即可。

二、服務器設定問題：

問題 1: ESP8266 無法連上 MQTT 服務器？

答：mosquitto 設定檔案需開啟 1883 埠，與予許任意使用者連線。啟動 mosquitto

時，帶入參數-v，可以即時觀看連線裝置是否成功連上。

問題 2: ip 配置問題。

答：ESP8266 的程式，固定了 MQTT server 的 IP，初期 ESP8266 設定為 Wifi 模

式，實驗後期，則切換成 AP 模式，服務器被分配到的 ip 是透過 ESP8266 所

給予，須重新修改，並燒錄新的程式碼。

問題 3: node-red 無法安裝其他工具？

答：安裝 node-red 時，資料夾權限設定錯誤所導致。

陸、 結論

本實驗目的從一開始，串連感測器，確定功能完整，到整合區域網路連線，導入 MQTT 通訊協議，最終實現物聯網(IoT)架構。

第一階段將目標鎖定 Arduino 與機電，傳感器的整合，除了認識基本硬體設備，電子元件，工作特性外，更深刻學習 Arduino IDE 軟體開發，燒錄程式，編譯程式，程式除錯等，此外，在過程中，學習拆解需求，分析問題，解決問題，體驗自己動手做的創客精神。

第二階段，架設服務器平台，串連網路，撰寫簡單互動網頁，達到遠程控制，此過程中，除了瞭解基本網路連線行為，更深刻體驗，封包格式，資料傳輸等知識。

透過此次的學習歷程，相信對於軟硬體相關知識的認識，又有更進一步的提升。

柒、 參考文獻資料

1. Arduino UNO IDE 開發與環境架設：<https://www.arduino.cc/en/software>
2. ESP8266:<https://randomnerdtutorials.com/projects-esp8266/>
3. brew 環境安裝：<https://brew.sh/>
4. node-red 環境安裝: <https://www.iamtie.com/2019/04/NodeJS.html>
5. MQTT 教學：<https://swf.com.tw/?p=1007>

6. 米羅科技文創學院 <https://swf.com.tw/?p=1007>
7. mosquit 教學 <https://jimirobot.tw/esp32-mosquitto-windows-mqtt-tutorial/>