

# 中華民國第 58 屆中小學科學展覽會 作品說明書

---

高級中等學校組 工程學科(一)科

**第三名**

052317

**遠端自動化-農田監控警示系統**

學校名稱：高雄市立高雄女子高級中學

作者：  高二 廖壬瑜  高二 左穎瑩  高二 林佳瑩	指導老師：  呂雲瑞  張志聰
---	-----------------------------

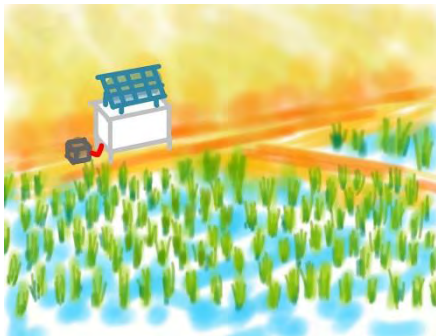
關鍵詞：監控、警示、Arduino 裝置

## 摘要

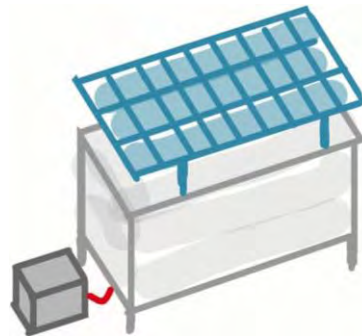
本研究目的在建立農田自動化遠端監控警示系統：

1. 利用感測器測得環境狀況，當數值超出警戒範圍，自動作出應對措施(例如灑水)調節環境
2. 手持裝置遠端顯示測得數值，並作出歷史趨勢圖，當數值超出警戒值時發出警示
3. 雲端同步儲存測得環境數值
4. 測得環境數值進行迴歸分析，預測未來變化趨勢

以「低成本」、「操作簡便」為方向，使用 Arduino 裝置檢測作物環境的「土壤濕度」、「光照度」、「空氣溫濕度」和「pH 值」；Google 表單儲存測得數值；應用程式 Blynk 即時監控警示；使用迴歸分析，做出趨勢預測。預期此系統可提升農夫工作效率，降低成本。



圖(一)本研究裝置農田實際配置示意圖



圖(二)本研究裝置示意圖

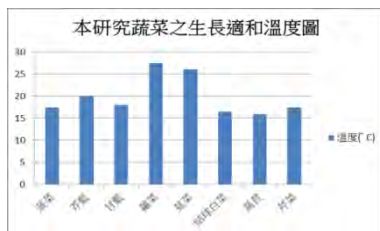
# 壹、 研究動機

我們曾在學校看過一部台灣紀錄片一無米樂，提及農夫辛勤播種、除草、澆水、施肥、噴灑農藥、收割，每天重複著單調繁瑣的體力活、全年無休卻收入微薄，投資報酬率低、工作辛苦。此外，現代社會農村勞動力嚴重不足、人口老化，更加深了農民的負擔。為減輕其壓力，我們設想：結合現代資訊科技，開發一套遠端監控系統，及時且準確監控作物生長環境。

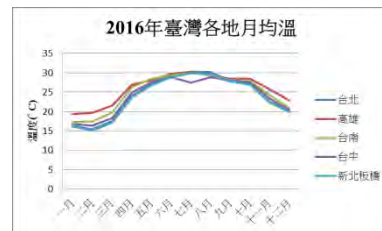
以讓農民能簡單上手為前提，本研究選用單晶片控制器 Arduino 作為控制面板，在農田放置感測器及灑水器，手持裝置應用程式 Blynk 及 Google 表單分別獨立接收測得資料。感測器監控環境，灑水器則藉由灑水調節異常之環境數值，應用程式遠距即時監控農田，Google 表單則作為雲端資料庫儲存歷史資料，供農民提取使用。

## 一、 蔬菜

我們以空氣溫度作為作物生長條件之指標，得出台灣較適合栽種之作物。



圖(三)本研究蔬菜生長適合溫度

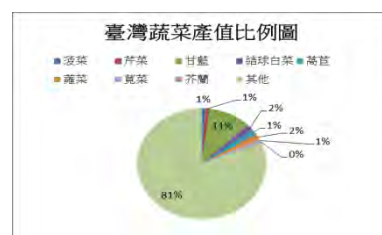


圖(四)臺灣各地月均溫

由圖(三)、(四)可知：本研究選用之八種蔬菜的生長適合溫度與台灣主要縣市溫度相近，且一年四季均可產出。



圖(五)臺灣蔬菜產量比例圖



圖(六)臺灣蔬菜產值比例圖

如圖(五)、(六)所示：本研究所選之八種蔬菜的產量及產值在總產量及產值中占有一定比例(2016年)。另外，將本研究推廣至其他作物，實用價值亦高。

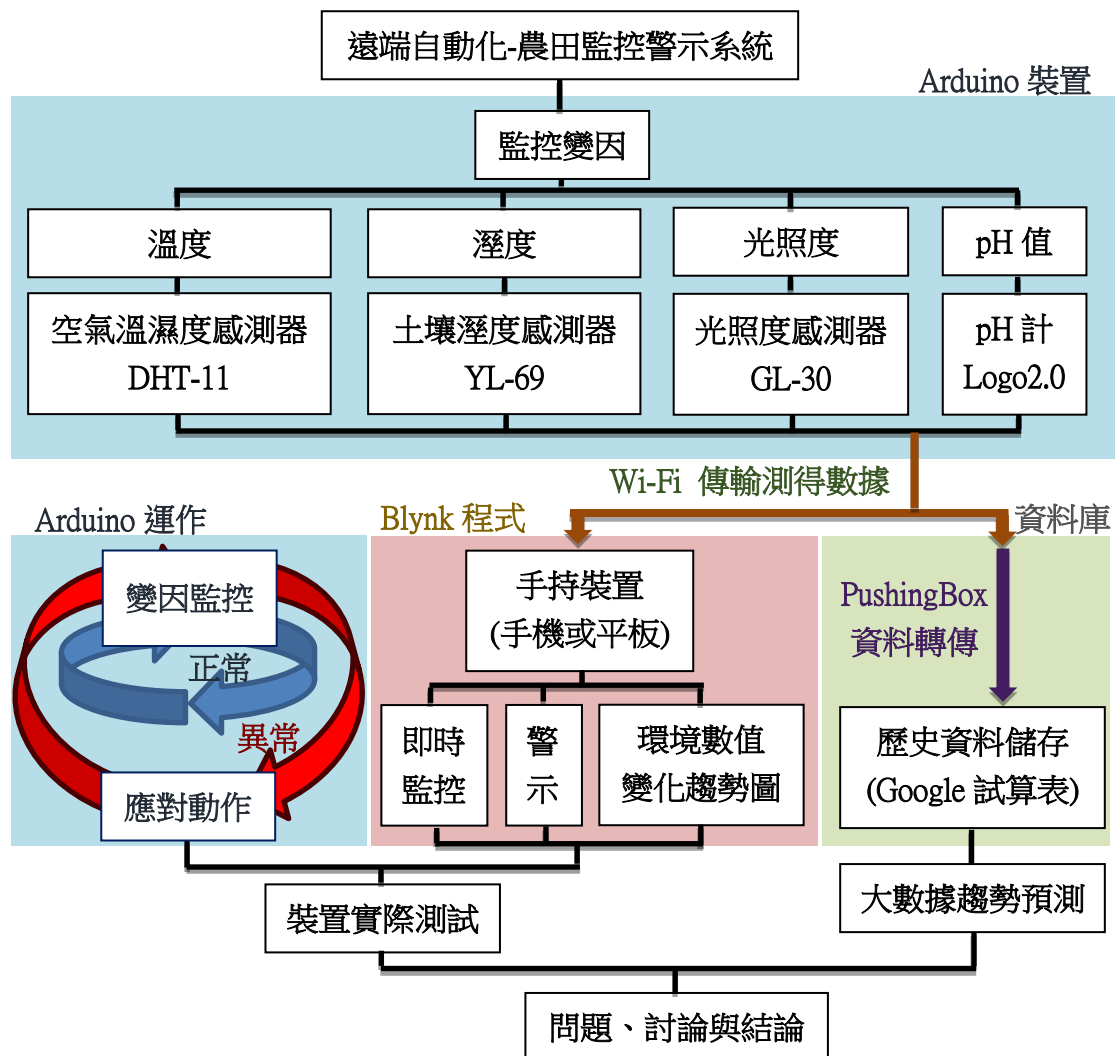
表(一)本研究蔬菜之簡介

菜名	特性	生長適溫(°C)	年產量(公噸)
菠菜	喜冷、涼氣候，耐寒	15~20	29848
芥藍	一年四季均可播種栽培	18~32	19265
甘藍	生長在涼爽地區	7~25	399175
蕓菜	耐澇、耐炎熱	15~40	35859
莧菜	耐熱力強、不耐寒冷	23~27	15606
結球白菜	適合涼爽環境 過高溫會抑制生長	15~26	80087
萵苣	適合生長於涼爽環境	11~25	61232
芹菜	喜濕冷氣候，不耐高溫	5~20	20477

參考文獻得知蔬菜產量、產值以及適合生長溫度後，本研究做出歸納整理，選出台灣常見之八種蔬菜並採用其最適生長範圍。

## 貳、研究目的

### 一、流程圖



## 二、研究目的

我們嘗試結合免費、易取得且操作簡單之網站及應用程式，開發一套遠距自動監控農田系統。首先，選用操作簡單的 Arduino 裝置連接感測器，偵測對植物影響較深的四項環境變因—土壤濕度、空氣溫度、光照度及 pH 值，利用 Wi-Fi 將測得數值傳至免費的 Google 表單，作為雲端資料庫，同時數值經 Arduino 裝置判斷，在手持裝置之應用程式發出警示、作出歷史數值變化趨勢圖且以應對措施調節異常環境變因，有即時監控警示之功能。

### (一) 監控警示系統

#### 1. 感測器控制面板(Arduino Wemos 板)

- (1)連接感測器，測量當下「土壤濕度」、「光照度」、「空氣溫度」、「空氣濕度」及「pH 值」；連接液晶顯示器，顯示當下測得之環境數值。
- (2)測得環境數值不合最適生長範圍時，作出應對動作直到環境變因回歸正常。

#### 2. 手持裝置之應用程式 Blynk

- (1)顯示當下測得之環境數值，並將歷史數值以折線圖呈現。
- (2)依據不同蔬菜給出不同的最適生長範圍，當測得之環境數值超出範圍時，亮紅燈並發出提醒(警示音、提醒欄)。

### (二) 資料庫歷史資料儲存

#### 1. PushingBox 網站、Google 表單

PushingBox 網站接收測得之環境數值並轉傳至 Google 表單儲存。

### (三) 趨勢預測

將測得歷史環境數值作迴歸分析，預測未來環境變化趨勢。

### (四) 裝置實際測試

我們與高雄市南區銀髮族農園合作，實際測試檢測本研究裝置之準確性。



圖(七)銀髮族農園



圖(八)農田一景



圖(九)農田二景

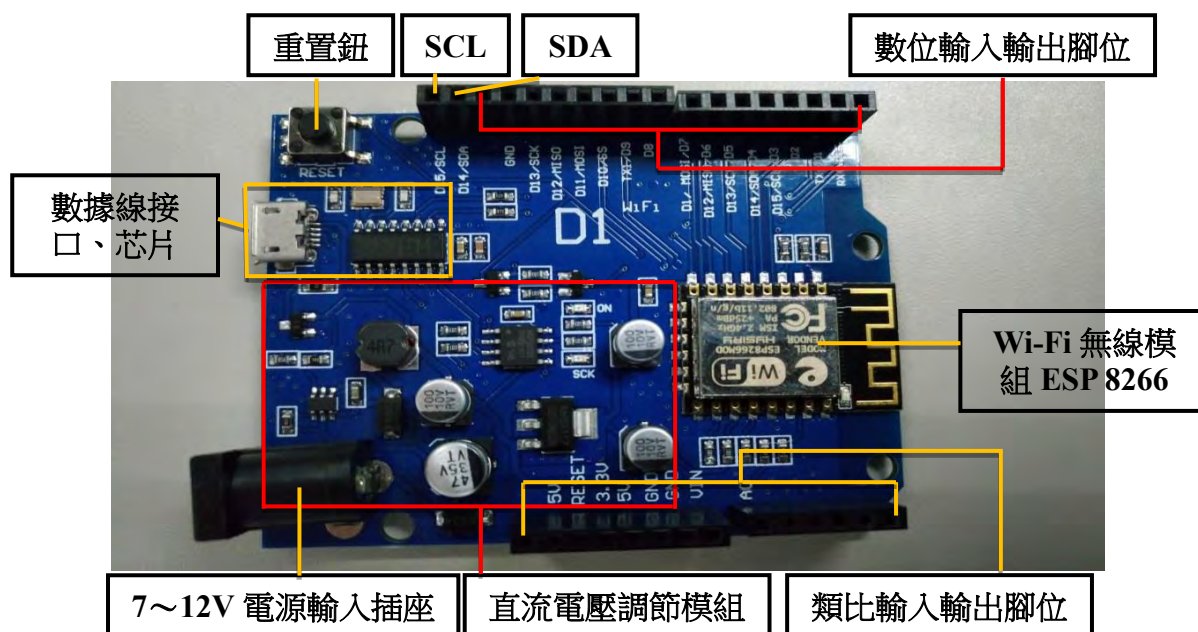
### 參、 研究設備及器材

				
圖(十) Arduino WeMos 板	圖(十一)馬達	圖(十二)光強度感測器 GY-30	圖(十三) Logo pH sensor v2.0	圖(十四)太陽能防水行動電源
				
圖(十五)液晶顯示器 2x16 LCD	圖(十六)市售檢測儀	圖(十七)土壤溼度感測器 YL-69	圖(十八)空氣溫濕度感測器 DHT11	圖(十九)麵包板、杜邦接頭

### 肆、 研究過程或方法

#### 一、 系統介紹

##### (一) Arduino(WeMos 板)介紹



圖(二十)Arduino (WeMos 板)結構圖

Arduino WeMos D1 為一物聯網控制面板，雖然接腳較其它控制面板少，但十分淺顯易懂，且已內建 Wi-Fi 晶片(ESP 8266)；價格相較於其它面板更是便宜許多。

表(二)常見 Arduino 控制面板比價表

Arduino 控制面板種類	價錢 (NT)
Arduino WeMos D1	185
Arduino UNO	180
Arduino UNO Rev3	840
Arduino Yun	2350
Arduino Mega2560 Rev3	1500

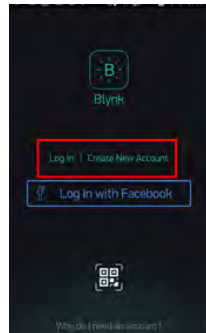
## (二) 手持裝置應用程式 Blynk 介紹

手持裝置應用程式 Blynk 為一免費軟體，連結至 Arduino 系統，易於操作。

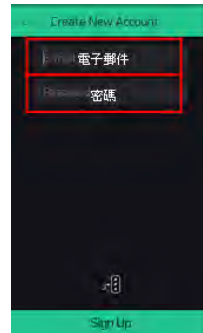
1. 下載手持裝置應用程式 Blynk：由手持裝置進入 Google Play 或 iPhone APP Store 下載（支援 Android、ios 版）。

2. 使用應用程式 Blynk

(1) 進入初始頁面與登入：可選擇建立新帳戶或由 Facebook 登入。

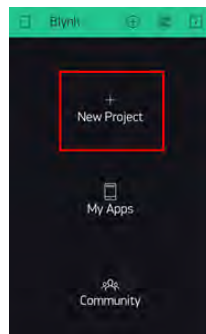


圖(二十一)初始頁面



圖(二十二)登入頁面

(2) 建立新專案：設定「專案」名稱及控制面板，建立新專案後產生認證碼。

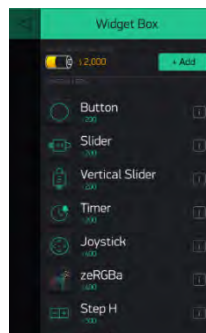


圖(二十三)建立專案



圖(二十四)設定名稱與控制面板

(3) 設置專案內容：圖(二十三)：「新增內容」最上方是個人點數，用以兌換「功能」，每個帳戶贈送 2000 點，超額便需付費買點。圖(二十四)：新增功能後可設定該功能細部項目。

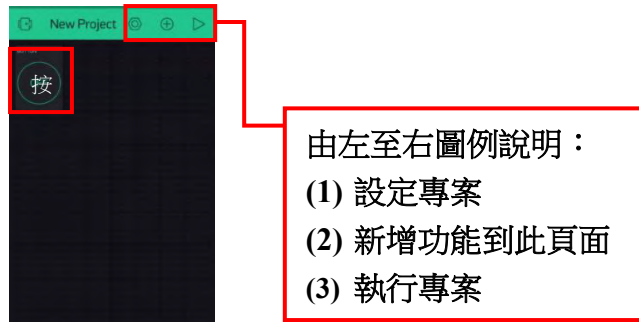


(二十五)功能選擇



圖(二十六)功能細部設定

(4) 最終控制頁面呈現：最終控制頁面如圖(二十六)，「功能」視窗可拖曳移動，也可新增功能。功能實際執行則需與程式碼相互搭配。



圖(二十七)最終控制頁面

### (三) 雲端資訊推送平台 PushingBox 介紹

雲端資訊推送平台 PushingBox 為一免費且少數能與 Google 相容的網站，有 Google 帳號就能註冊使用，十分簡便。

#### 1. 登入雲端資訊推送平台 PushingBox

搜尋 PushingBox，點選頁面右上方進入，以 Google 帳號登入。



圖(二十八)登入雲端資訊推送平台 PushingBox 流程圖

#### 2. 功能介紹

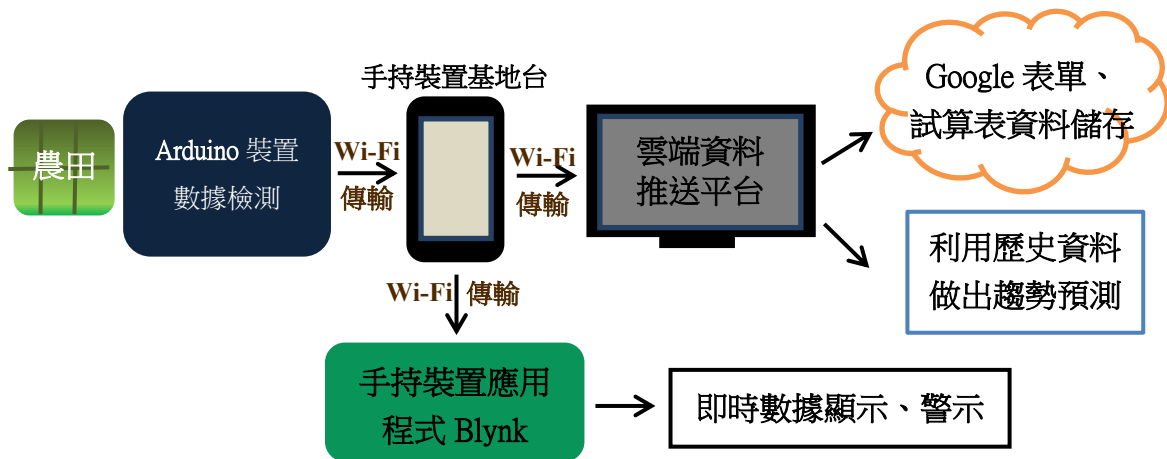


圖(二十九)雲端資訊推送平台 PushingBox 功能簡介

PushingBox 可選擇建立多種類型的專題，且使用介紹清楚易懂，本研究中是作為 Arduino 裝置傳送資料至 Google 雲端的中繼站。

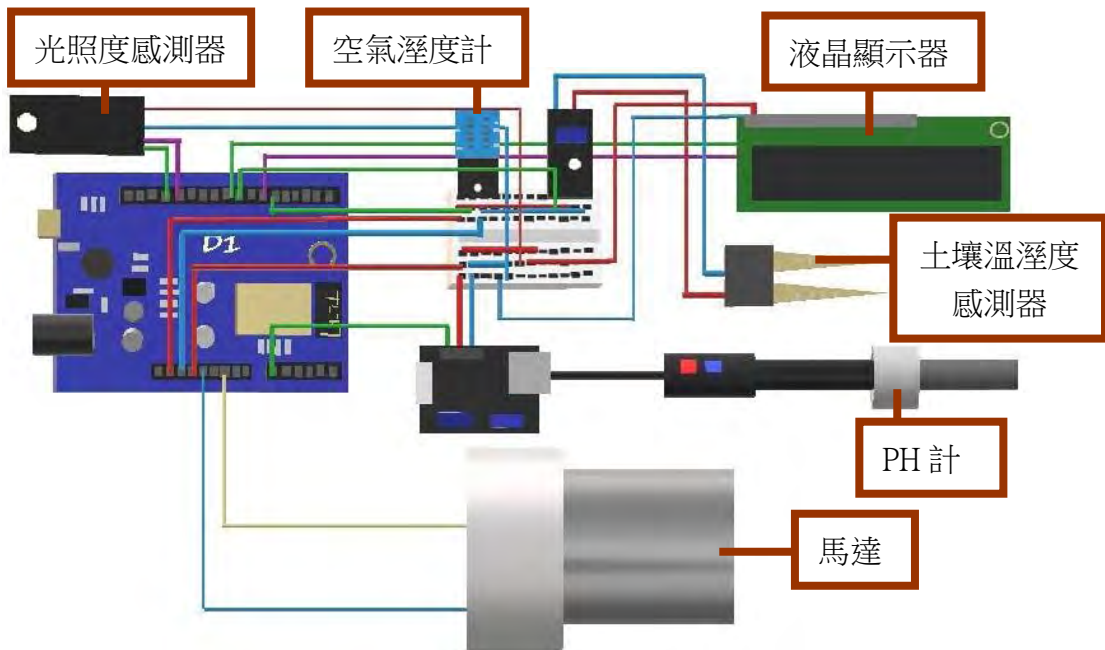


### 三、實驗步驟



圖(三十)本研究資料傳輸流程圖

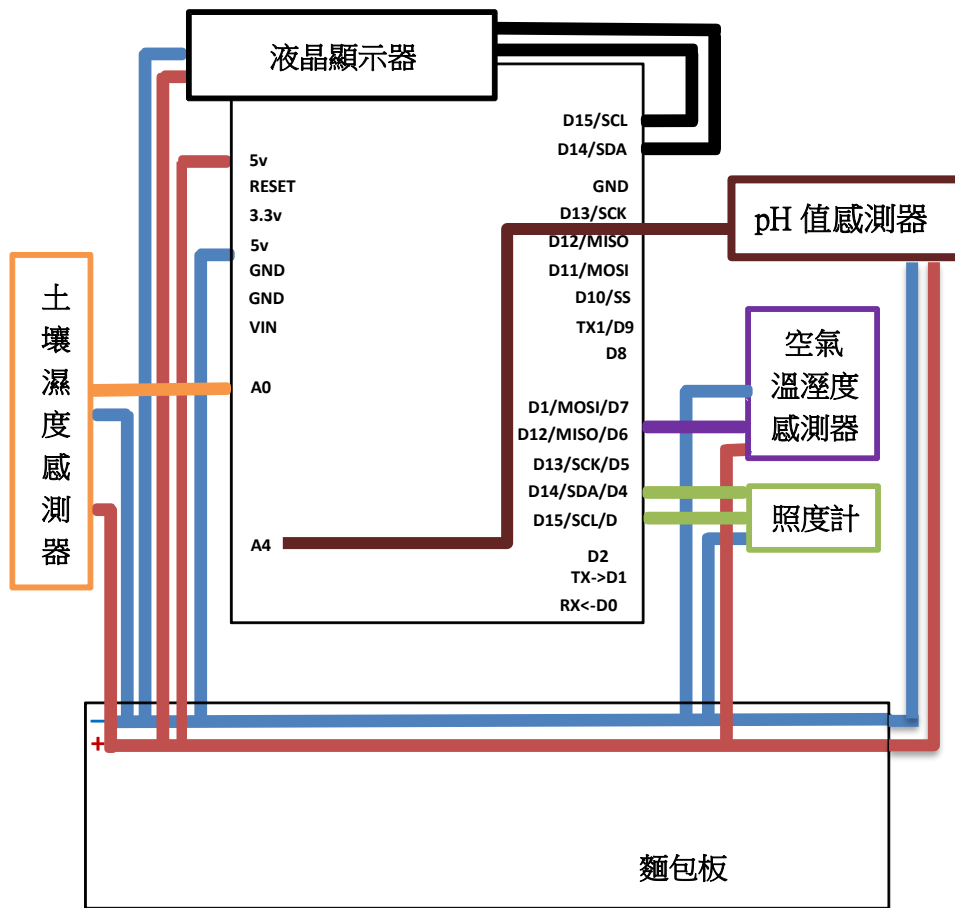
#### (一) Arduino裝置程式監控與警示



圖(三十一)本研究之 Arduino 裝置電路圖

表(三) 本研究之 Arduino 裝置電路腳位表

器材	腳位
土壤溼度感測器 (YL-69)	5V、GND、A0
液晶顯示器 (LCD2x16)	SCL、SDA、VCC、GND
空氣溫溼度感測器 (DHT-11)	D6、5V、GND
光照度感測器 (GY-30)	D3、D4、5V、GND
pH 計 (Logo pH sensor v2.0)	5V、GND、A4



圖(三十二) 本研究之 Arduino 裝置腳位圖

## 1. 環境偵測

### (1) 土壤濕度偵測(YL-69)

連接 Arduino WeMos 板之 5V、GND、A0 腳位。

```

combine
182 //在此從所有感測器接收數值 要存到上面的變數中
183 void readSensor(){
184   Serial.println("timer event");
185   moist = analogRead(MoistPIN); //讀取土壤濕度
186   moist=map(moist,250,800,0,100);
187   lux = lightMeter.readLightLevel();
188   h = dht.readHumidity();
189   t = dht.readTemperature();
190 }

```

圖(三十三)土壤溼度感測器程式

```

COM4
moist has some problems !!
lux OK!
temperature has some problems!!
Light: 357 lux
Humidity: 50.00
Temperature: 26.00 °C

```

圖(三十四)土壤濕度監控視窗

### (2) 空氣溫溼度偵測(DHT-11)

連接 Arduino 板之 D6、5V、GND 腳位，空氣溫度及濕度能同時偵測出。

```

combine
182 //在此從所有感測器接收數值 要存到上面的變數中
183 void readSensor(){
184   Serial.println("timer event");
185   moist = analogRead(MoistPIN);//讀取土壤濕度
186   moist=map(moist,250,800,0,100);
187   lux = lightMeter.readLightLevel();
188   h = dht.readHumidity();
189   t = dht.readTemperature();
190 }

```

圖(三十五)空氣溫溼度感測器程式

```

COM4
moist has some problems !!
lux OK!
temperature has some problems!!
Light: 557 lux
Humidity: 50.00
Temperature: 26.00 °C

```

圖(三十六)空氣濕度監控視窗

### (3) 光照度偵測(GY-30)

連結 Arduino 版的 SCL、SDA、5V、GND 腳位。

```

combine
182 //在此從所有感測器接收數值 要存到上面的變數中
183 void readSensor(){
184   Serial.println("timer event");
185   moist = analogRead(MoistPIN);//讀取土壤濕度
186   moist=map(moist,250,800,0,100);
187   lux = lightMeter.readLightLevel();
188   h = dht.readHumidity();
189   t = dht.readTemperature();
190 }

```

圖(三十七)光照度感測器程式

```

COM4
moist has some problems !!
lux OK!
temperature has some problems!!
Light: 557 lux
Humidity: 50.00
Temperature: 26.00 °C

```

圖(三十八)光照度監控視窗

## 2. 監控與警示

Arduino 裝置程式中分別寫入不同蔬菜適合的環境變因之適當範圍，蔬菜種類選定後，即自動對應到該蔬菜之最適生長範圍，並以此判定是否發出警示。

表(四)本研究匯整之常見八種蔬菜環境狀態警示數值及蔬菜代碼

	空氣溫度 (°C)	土壤濕度 (%)	光照度 (Lux)	pH 值	程式代碼
菠菜	低於 15	低於 70	低於 10000	低於 6	Vegetables[0]
	高於 20	高於 80	高於 30000	高於 7	
甘藍	低於 7	低於 70	低於 10000	低於 5.5	Vegetables[1]
	高於 25	高於 80	高於 40000	高於 6.5	
芥藍	低於 18	低於 80	低於 13000	低於 5.5	Vegetables[2]
	高於 32	高於 90	高於 40000	高於 6.8	
蕹菜	低於 15	低於 75	低於 8000	低於 5.5	Vegetables[3]
	高於 40	高於 90	高於 40000	高於 6.5	

萵苣	低於 11	低於 70	低於 10000	低於 5.3	Vegetables[4]
	高於 25	高於 85	高於 20000	高於 6.8	
結球白菜	低於 15	低於 80	低於 10000	低於 5.5	Vegetables[5]
	高於 26	高於 90	高於 40000	高於 6.8	
芹菜	低於 5	低於 60	低於 10000	低於 5.6	Vegetables[6]
	高於 20	高於 80	高於 25000	高於 6.8	
莧菜	低於 23	低於 60	低於 10000	低於 4.6	Vegetables[7]
	高於 27	高於 80	高於 13000	高於 7.5	

本研究查閱文獻整理出此八種蔬菜之生長環境閾值，作為判斷發出警示的標準，每種蔬菜都有各自的程式代碼。

以甘藍為例，若農夫在手持裝置端選擇種植甘藍，就會送出[1]的訊號，程式便可根據「甘藍」(Vegetables[1])的警示數值運作。

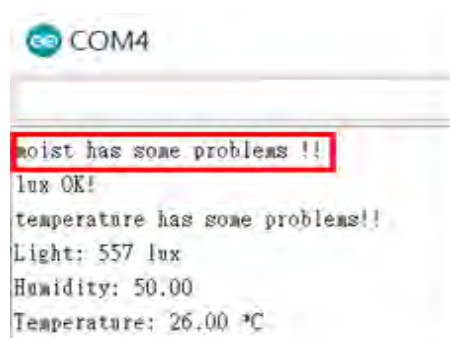
### (1) 土壤濕度監控警示

```

combina
PerfectValue vegetable[0]; //蔬菜生長環境的閾值，目前只一種蔬菜
4:
5: //在此設定每一種蔬菜的適合環境 (目前只一種)
6: void vegetableSetting() {
7:   vegetable[0].minDeg = 15;
8:   vegetable[0].maxDeg = 20;
9:   vegetable[0].minLux = 0;
10:  vegetable[0].maxLux = 10000;
11:  vegetable[0].minMoist = 70;
12:  vegetable[0].maxMoist = 80;
13: }
14:
15: vegetable[1].minDeg = 7;
16: vegetable[1].maxDeg = 23;
17: vegetable[1].minLux = 10000;
18: vegetable[1].maxLux = 40000;
19: vegetable[1].minMoist = 70;
20: vegetable[1].maxMoist = 80;
21: }

```

圖(三十九)土壤濕度警示程式



圖(四十)土壤濕度警示監控視窗

### (2) 空氣溫度警示

```

combina
PerfectValue vegetable[0]; //蔬菜生長環境的閾值，目前只一種蔬菜
4:
5: //在此設定每一種蔬菜的適合環境 (目前只一種)
6: void vegetableSetting() {
7:   vegetable[0].minDeg = 15;
8:   vegetable[0].maxDeg = 20;
9:   vegetable[0].minLux = 0;
10:  vegetable[0].maxLux = 10000;
11:  vegetable[0].minMoist = 70;
12:  vegetable[0].maxMoist = 80;
13: }
14:
15: vegetable[1].minDeg = 7;
16: vegetable[1].maxDeg = 23;
17: vegetable[1].minLux = 10000;
18: vegetable[1].maxLux = 40000;
19: vegetable[1].minMoist = 70;
20: vegetable[1].maxMoist = 80;
21: }

```

圖(四十一)空氣溫度警示程式



圖(四十二)空氣溫度警戒監控視窗

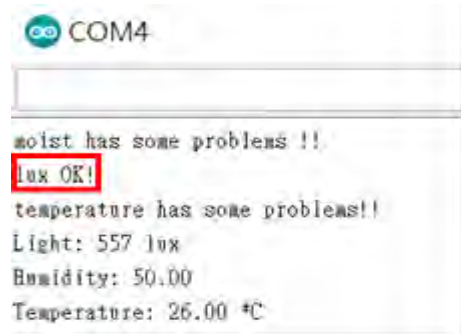
### (3) 光照度警示

若光照強度過大，會使植物的葉綠素遭破壞而失去其功能；反之，若光照

強度不足則會造成植物發育不良。

```
combine
// PerfectValue vegetables[]; //蔬菜最佳環境的陣列 目前有3種蔬菜
4) //在此設定每一種蔬菜的適合環境 只寫一次
5) void vegetableSetting() {
6)   vegetables[0].minDeg = 15;
7)   vegetables[0].maxDeg = 30;
8)   vegetables[0].minLux = 0;
9)   vegetables[0].maxLux = 10000;
10)  vegetables[0].minMoist = 30;
11)  vegetables[0].maxMoist = 80;
12)
13)  vegetables[1].minDeg = 7;
14)  vegetables[1].maxDeg = 25;
15)  vegetables[1].minLux = 10000;
16)  vegetables[1].maxLux = 40000;
17)  vegetables[1].minMoist = 70;
18)  vegetables[1].maxMoist = 80;
19}
```

圖(四十三)光照度警示程式

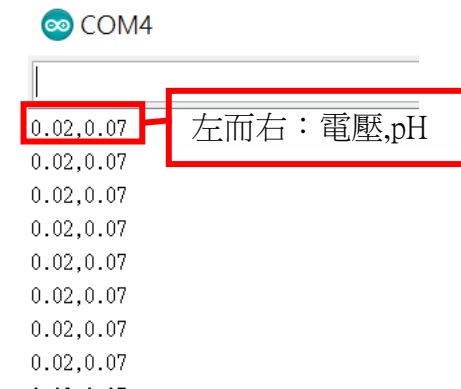


圖(四十四)光照度警示監控視窗

#### (4) pH 值警示

```
37 void loop(void)
38 {
39   static unsigned long samplingTime = millis();
40   static unsigned long printTime = millis();
41   static float pHValue, voltage;
42   if (millis() - samplingTime > samplingInterval)
43   {
44     pHArray[pHArrayIndex++] = analogRead(A0);
45     if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
46     voltage = averagarray(pHArray, ArrayLenth) * 5.0 / 1024;
47     pHValue = 3.5 * voltage + Offset;
48     samplingTime = millis();
49   }
50 }
```

圖(四十五) pH 值警示程式



圖(四十六) pH 值警示監控視窗

#### (5) 應對措施—灑水

經程式自動判斷，若環境的「溼度不足」或「過酸過鹼」，我們採取之應對措施皆為「灑水」。(其他環境數值各有其應對方式，我們以灑水為例)

```
49 //lcd.setCursor(0,1);
50
51 if(SensorValue >= 1000) {
52   //lcd.println("No soil Haha");//LCD印出訊息
53   Serial.println("seror not ready!"); //感測器有錯誤
54   Serial.println(SensorValue);//印出目前的SensorValue值
55   digitalWrite(motorPin, HIGH); //關閉馬達
56   delay(500); // 等待500ms後重新開始檢測
57   Serial.println("restart!"); //重新開始
58 }
```

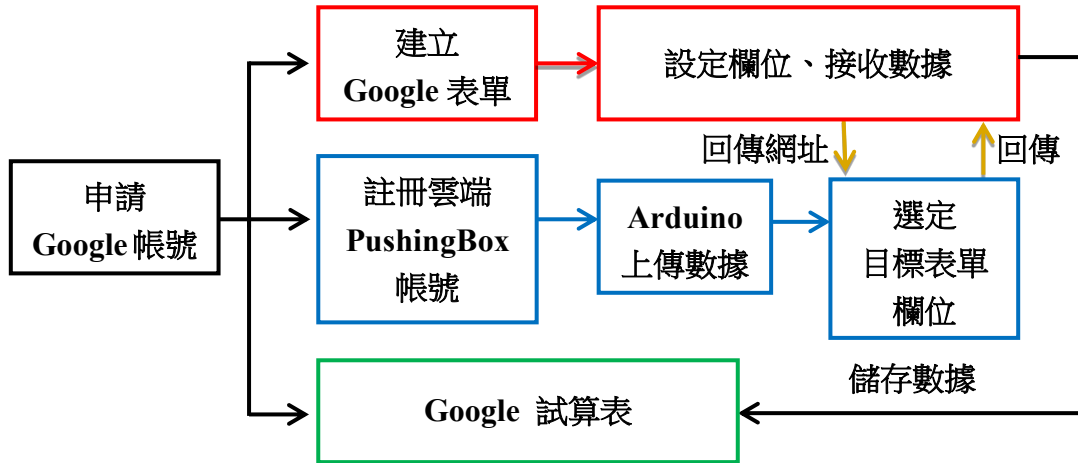
圖(四十七)抽水馬達程式

### 3. 液晶顯示器環境數值之監控

測得之環境數值於液晶顯示器(2x16LCD)螢幕上顯示。



## (二) 雲端歷史資料儲存



圖(五十五)雲端資料儲存流程圖

### 1. Google 表單建立

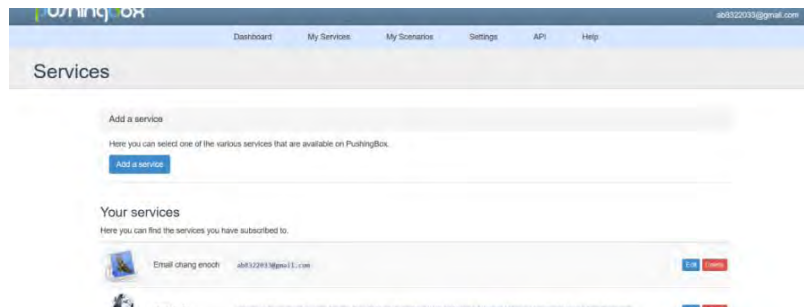


圖(五十六) Google 表單頁面

設立 Google 表單，標示出欄位名稱，並於設定中選擇身分為「使用人」，點擊「回覆/取得預先填入之網址」，此網址即是數據接收的目的地。

### 2. 雲端資訊推送平台 PushingBox 的應用

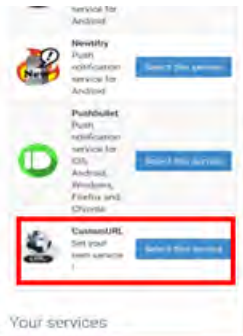
PushingBox 為少數與 Google 相容的網站，由於 Arduino 所偵測之數據無法直接傳送至 Google 表單，於是我們藉由 PushingBox 作為兩者之資訊傳輸媒介。



圖(五十七)PushingBox 頁面

#### (1) 新增專案與選定欄位

在 PushingBox 建立 CustomURL 檔專案，前述取得網址為數據接收位置。



圖(五十八)專案選擇



圖(五十九)欄位網址設定

## (2) 設定變數

由於 Arduino 裝置偵測的數據都是數字串，無法分辨數據的種類，因此需在「我的場景(My Scenarios)」新增變數，以「\$」前後包住變數，宣告數據來源以標明每筆資料的種類(例如土壤濕度)。

表(五)資料種類與程式變數對照表

資料種類	變數名稱
土壤濕度	\$moist\$
空氣溫度	\$t\$
空氣濕度	\$hum\$
光照度	\$Lux\$
酸鹼值	\$pHValue\$

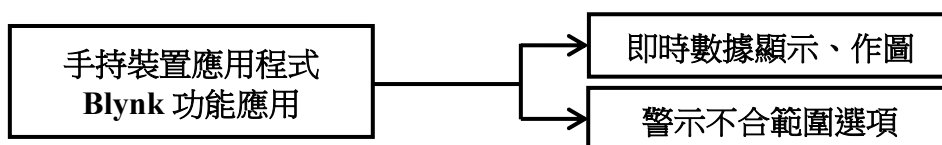
## 3. 測得數值之呈現方式

時間戳記	空氣濕度	照度	溫度	土壤濕度	酸鹼度(pH)
1825 2018/6/6 上午 2:19:44		56	235	29	7.1
1826 2018/6/6 上午 11:20:55		59	54306	29	7.3
1827 2018/6/7 上午 11:15:22		58	55393	29	7.8
1828 2018/6/8 上午 11:10:46		49	54035	30	6.6
1829 2018/6/9 上午 11:27:23		48	53197	31	6.8
1830 2018/6/10 上午 11:22:44		48	55467	30	7
1831 2018/6/11 上午 11:25:31		49	42396	26	6.5
1832 2018/6/12 上午 11:25:46		48	54887	27	7
1833 2018/6/13 上午 8:05:23		59	55382	29	6.9

圖(六十)Google 試算表頁面呈現

測得數據以 Google 表單儲存，並以 Google 試算表將所得的數據以表格的樣式呈現，能將繁複的資料簡潔化，供日後大數據的應用。

## (三) 手持裝置應用程式 Blynk 監控與警示



圖(六十一)Blynk 運作流程圖



### 1. 選擇蔬菜與種類(共 8 種)

應用程式 Blynk 中蔬菜種類選擇的菜單有 8 種蔬菜供農夫挑選，不同的選項對應各自的最適生長環境數值範圍。圖(五十八)

### 2. 即時監控

本研究設置了空氣溫溼度、光照度、土壤濕度及 pH 值之監控視窗，可即時掌握農田環境變化。圖(五十九)



圖(六十二)蔬菜選擇細部設定



圖(六十三)監控視窗細部設定

表(六) 監控視窗與細部設定

	空氣溫度	光照度	土壤濕度	空氣濕度	pH 值
監控 視窗 細部 設定					
視窗	目前氣溫 24.00°C	目前光照 146lux	目前土壤濕度 29.00%	目前空氣濕度 37.00%	目前PH值 7.2

### 3. 測得數據作圖與警示

#### (1) 測得數據作圖

利用應用程式 Blynk 將測得數據繪成折線圖，由折線圖可知農田環境的歷史變化趨勢。時間軸可調整，最長可呈現三個月間之環境變化。



圖(六十四)測得數據作圖視窗



圖(六十五)測得數據作圖細部設定

## (2) 警示

在 Arduino 裝置程式中預設好警示數值，判斷測得土壤溼度數值是否不合最適生長範圍，若超出或低於範圍再將警示指令傳至手持裝置應用程式 Blynk。應用程式 Blynk 有警示功能，讀取警示指令後發出提示(警示欄、警示音且警示燈由綠轉紅)。



圖(六十六)提示音細部設定



圖(六十七)警示燈細部設定

## (四) 趨勢分析

將測得的歷史資料由雲端資料庫擷取下來，匯入 Excel 中繪製趨勢預測圖

2018/6/6 上午 11:20:55	59	54306	29	54	7.3
2018/6/7 上午 11:15:22	58	55393	29	63	7.8
2018/6/8 上午 11:10:46	49	54035	30	61	6.6
2018/6/9 上午 11:27:23	48	53197	31	64	6.8
2018/6/10 上午 11:22:44	48	55467	30	53	7
2018/6/11 上午 11:25:31	49	42396	26	72	6.5
2018/6/12 上午 11:25:46	48	54887	27	56	7
2018/6/13 上午 11:05:23	59	55382	29	60	6.9

圖(六十八)資料表格

由左而右：  
 (1)空氣濕度  
 (2)照度  
 (3)溫度  
 (4)土壤濕度  
 (5)酸鹼度

## (五) 裝置實際測試

表(七)裝置實測對照



### 1. 實作

我們實際到農田內以市售檢測儀檢測並紀錄測得環境數值，再與本實驗之手持裝置應用程式監控視窗顯示之數值做比較；另外實際測式感測器與灑水器對環境之調節狀況。



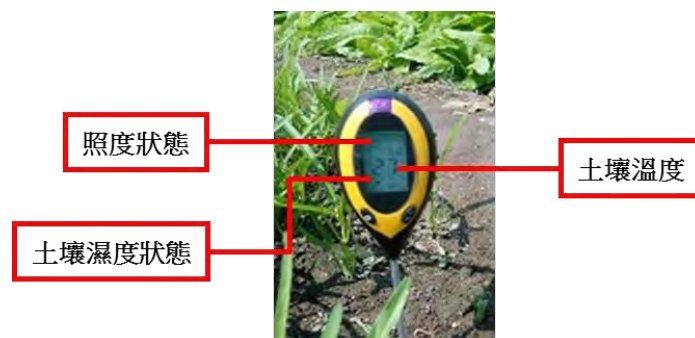
圖(六十九)監控畫面與裝置



圖(七十)市售檢測儀數值紀錄

### 2. 市售檢測儀使用說明

市售檢測儀僅能顯示土壤溫度及土壤酸鹼度兩項環境變因之實際數值，而土壤濕度及光照度則只能顯示狀態(例：土壤濕度為 DRY+)，還需由狀態數據對照表對照，才能得知測得環境變因之實際數值落在哪個範圍內。



圖(七十一)功能說明

表(八)四合一數位式土壤檢測儀數值狀態對照表

土壤	DRY+	<5%	DRY	5%~10%	NOR	10%~20%
濕度	WET	20%~30%	WET+	>30%		
光照度	LOW-	<500Lux	LOW	500~5000Lux	LOW+	5000~10000Lux
	NOR-	10000~25000Lux	NOR	25000~60000Lux	NOR+	60000~80000Lux
	HGH	80000~100000Lux	HGH	100000~120000Lux	HGH+	>120000Lux

## 伍、研究結果

一、感測器與灑水裝置兩相配合，將環境變因控制在一定範圍內。

(一) 感測器定時傳回測得環境變因之數值。

(二) 當農田端感測器測得之環境數值經 Arduino 裝置程式判斷「已超出作物最適生長範圍」後，程式自動發出指令，進行灑水動作：馬達轉動→抽水→灑水。



圖(七十二)灑水器調節概念圖

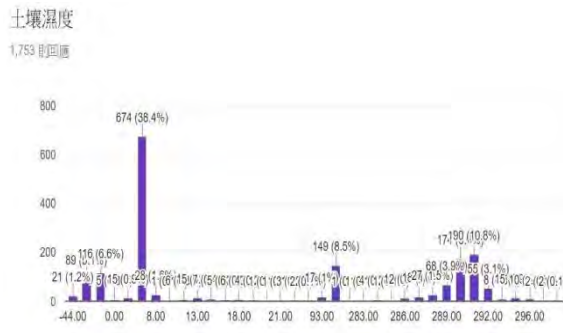
(三) 至感測器測得之環境數值回歸作物之最適生長範圍內，程式判定後發出指令，停止灑水：馬達停止運轉→結束抽水→結束灑水。

二、數據經 Arduino 裝置之 Wi-Fi 無線模組傳送至 Google 雲端儲存，於遠端進行資料檢視，提升農作物管理效能。

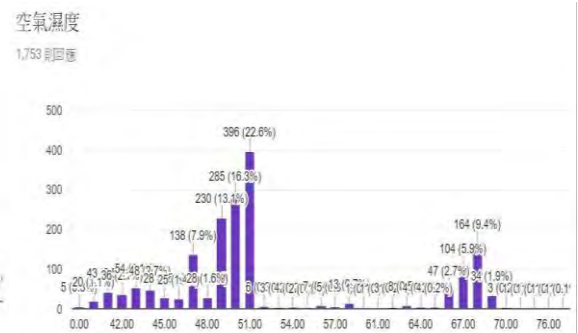
作法：將 Arduino 裝置由 UNO 板換成 WeMos 板(已整合 Wi-Fi 無線模組)。

(一) 將 Arduino 裝置測得之數值經 Wi-Fi 無線模組傳送至雲端資訊推送平台 PushingBox，再由 PushingBox 轉傳至 Google 表單及 Google 試算表。

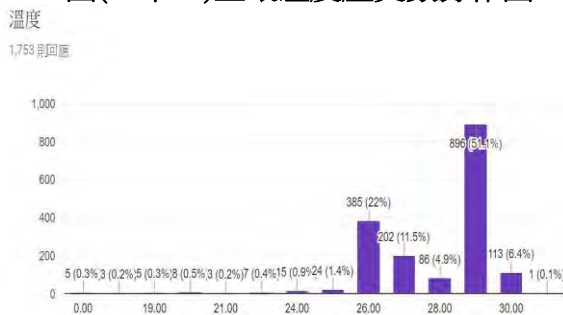
(二) 儲存至 Google 表單之歷史資料可輕易於網頁檢視或下載，進行資料分析。



圖(七十三)土壤溫度歷史數據作圖



圖(七十四)空氣濕度歷史數據作圖



圖(七十五)空氣溫度歷史數據作圖



圖(七十九)即時數據監控視窗

五、手持裝置端 Blynk 介面可選擇不同蔬菜，判斷目前環境是否適合生長並進行警示。

(一) 手持裝置應用程式 Blynk 端設置作物選擇菜單，不同蔬菜可切換不同頁面檢視即時環境資訊，Arduino 裝置端根據不同作物各自最適生長範圍判斷最是否發出警示，將警示指令傳至應用程式 Blynk。



圖(八十)蔬菜種類選擇視窗



圖(八十一)蔬菜種類選擇菜單

(二) 在應用程式 Blynk 選擇作物，可檢視作物之土壤溫濕度、空氣溫度、光照度及酸鹼度等即時資訊和最適環境條件，若環境變因不符合最適值，則於手持裝置應用程式 Blynk 頁面顯示紅色燈號進行警示，提醒欄跳出、警示音響起。



圖(八十二)警示提醒視窗

六、將已測得歷史數值繪成趨勢圖，可根據數值變化趨勢進行對應措施。

作法：於手持裝置應用程式 Blynk 將「空氣溫度」、「光照度」、「土壤溼度」、「空氣濕度」及「酸鹼度」即時繪製成折線圖，並可切換一小時、一天、一周、一個月、三個月等不同時間軸之頁面，檢視作物生長環境變化趨勢。



圖(八十三)趨勢圖視窗

七、以「硬體價格便宜」、「軟硬體規格通用」、「建置及維護簡易」為前提，選用「Google 表單」作為資料庫、「PushingBox 網路平台」作為資料推送中繼介面。

作法：為避免同一表單填寫者，因在短時間內傳送多筆資料，而被 Google 表單視為入侵動作、啟動自我保護機制進而拒絕接收傳輸資料，導致資料遺失，我們選擇 PushingBox 網站填寫 Google 表單，此網站因被 Google 表單信任而不會啟動表單的自我保護機制(也稱作「相容」)。缺點為每日只可傳送 2000 筆資料，但是，農園環境變化少有短時間劇烈變化，約每 30 分鐘傳送一筆檢測資料即可，針對小型農田來說，PushingBox 提供的流量已是相當足夠。

八、利用 Google 表單儲存的歷史資料推估未來環境趨勢變化

藉由 Google 表單儲存的歷史資料，以迴歸分析作出未來 3 日的環境變化預測。我們以溫度為例：



圖(八十四)溫度趨勢預測圖

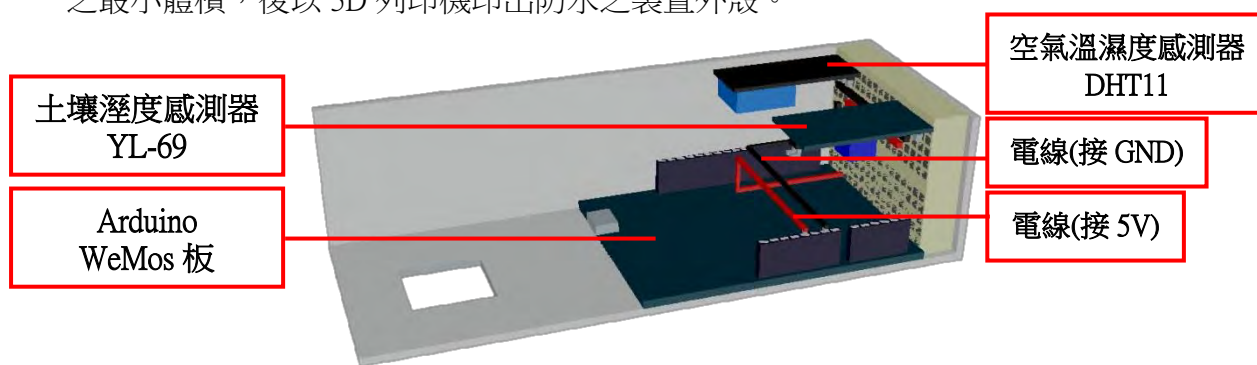
表(九)趨勢預測數值表

時間表	值	趨勢預測	較低的信賴繫結	較高的信賴繫結
6月6日	29			
6月7日	29			
6月8日	30			
6月9日	31			
6月10日	30			
6月11日	28			
6月12日	28			
6月13日	28	28	28.00	28.00
6月14日		27.75142492	25.88	29.62
6月15日		27.52584878	25.01	30.04
6月16日		27.30027264	24.27	30.33

由上之圖表，虛線部分為趨勢預測，兩旁的橙色實線表示可能的變動範圍，如此即可及早作出防範措施。

#### 八、設計出最小體積裝置，精確計算並以 3D 列印製出防水外殼

以繪圖程式設計感測器、電路等裝置之最合適位置配置，計算並整合至不影響運作之最小體積，後以 3D 列印機印出防水之裝置外殼。



圖(八十五)3D 裝置配置示意圖



圖(八十六)3D 手繪裝置底殼



圖(八十七)3D 手繪裝置頂殼

#### 九、裝置實際測試

以蘆菜為例生長期約 15 天。我們以本研究裝置測得之 15 天環境數值與氣象局、市



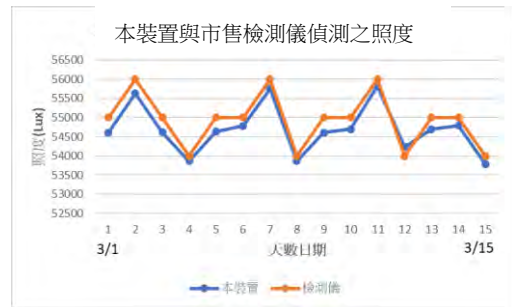
售檢測儀的數據比對，檢驗**準確性**；令接水管至馬達實際測試其調節環境之狀況。

(一) 本裝置與氣象局偵測之溫度比較

本系統溫度之偵測相較於氣象局資料誤差為 $\pm 2^{\circ}\text{C}$ ，有一定準確性。

(二) 本裝置與市售檢測儀偵測之照度比較

本系統照度之偵測與市售檢測儀之數據十分相近，精準度高。



圖(八十八)本裝置溫度準確性比較折線圖 圖(八十九)本裝置照度準確性比較折線圖

## 陸、 討論

### 一、 問題探討

(一) 偵測環境數值之單位轉換：

原先土壤濕度感測器(YL-69)測得的數值單位較少被使用，所以我們撰寫程式轉換單位，以百分比的形式呈現資料，即可方便估算資料**變化趨勢**。

(二) 利用**免費、普遍**的網站自動傳輸及儲存歷史數值：

我們利用:Google 表單填寫完畢後會自動將填寫內容儲存至 Google 試算表的特性，填寫測得環境數值至表單，以 Google 試算表作為歷史資料庫。

而由於 Google 有自我防禦機制，只有和它「相容」的網站才能使用表單，而 Arduino 不在此列。而 PushingBox 則是和 Google 相容的、可免費接收與傳輸訊息的網站，在 Arduino 程式中將環境數值傳輸至 PushingBox 後，由 PushingBox 自動填寫表單，後 Google 表單的 Google 試算表將歷史數值作出折線圖。

因此得以儲存數據至歷史資料庫，並可作得變化趨勢圖，供農夫作更多應用。

(三) 供電設備選用**太陽能防水行動電源**：

最初以行動電源供電，具**可充電、體積小**之優點，缺點為須定時充電。後來我們以太陽能防水行動電源作為供電設備，除兼具前述之優點，**裝置更可自力運作**。

## 二、未來展望

### (一) 手持裝置加入控制灑水功能(除自動以灑水調節環境變因，亦可手動控制灑水)：

期望本裝置除了能在遠端手持裝置接收農田端感測器測得資訊，做出監控及警示，自動在判斷環境數值達危險值後，發出指令控制農田端 Arduino 裝置進行灑水外；另外更可在手持裝置端發出灑水與否之指令：

當農田端感測器測得之環境數值經 Arduino 裝置判斷「已超出作物最適生長範圍」後，程式自動發出指令進行灑水。另在手持裝置端進行警示，農夫可控制執行或停止灑水，由 wi-fi 將指令回傳至農田端 Arduino 裝置，進行或停止灑水。

### (二) 農田環境與地區狀況即時比對(得知農田異常狀況)

除了現有感測器測得之農田環境數值作出歷史趨勢圖，期望未來加入將氣象局測得之地區環境狀況，一同繪製成趨勢圖、兩線對比，了解單一農田環境相對於大範圍地區之異常變化。

### (三) 自行撰寫程式以更貼近需求

本研究整合了數種資源加以延伸應用，但還是有相當程度的應用限制，期望未來能自撰程式，更迎合實際需求。

## 柒、結論

本研究成功建立一套農田遠端監控警示系統，幫助農夫節省時間、有效管理農田。

### 一、監控警示系統

農夫原本需定時親自至農田了解作物情況，藉由環境數值感測器全面監控農田。

#### (一) 感測器控制面板(Arduino Wemos 板)

1. 藉此控制面板之 Wi-Fi 傳輸，可自農田之 Arduino 裝置以 Wi-Fi 大範圍傳送測得數值至遠距監控裝置，節省時間與體力。
2. 感測器與灑水裝置配合，調節環境變因，將之控制在適合植物生長的範圍內。

#### (二) 手持裝置之應用程式(Blynk 應用程式)

1. 顯示當下之作物環境資訊供農夫判讀以進行下一步工作，助農夫更精準判斷。
2. 根據不同蔬菜給出各自最適生長範圍。

3. 隨時比對當下測得之環境數值是否超出最適生長範圍，自動發出警示。
4. 將環境數值 3 個月內之測得歷史數據(來自 Arduino)繪製成折線圖，以得知過去環境數值變化趨勢。

## 二、資料庫歷史資料儲存

### (一) PushingBox 網站、Google 表單

1. 以免費、普遍、簡單為前提，以 Google 表單作為雲端資料庫，將資料分享範圍擴展至無線網路。未來可參考比對，找出環境數值變化週期規律以做應對。
2. 以和 Google 相容的雲端資料推送平台 PushingBox 作為資料傳輸中繼站，將 Arduino 裝置測得環境數值填入 Google 表單儲存。

## 三、未來趨勢預測

本研究以 Aruino 裝置測得數據作迴歸分析，推估未來三日環境數值可能變化之範圍，供農夫提早作出適切的應對措施。

## 四、裝置實際測試

本研究裝置經實際測試後，得知其具有一定之準確性，具實用性。

# 捌、參考資料及其他

## 一、參考文獻

- (一) 廖文淵等人，2014，「以 Arduino 發展平台為基礎之智慧生活監控」，德霖學報，第 27 期，1 月。
- (二) 文淵閣工作室，2014，「手機應用程式設計 APP Inventor 專題 特訓班」，鄧文淵，碁峯資訊股份有限公司。
- (三) 梅克工作室，2014，「Arduino 微電腦控制實習(OZONE 適用)」，郭靜文，台科大圖書股份有限公司。
- (四) Swarup S. Mathurkar, D. S. Chaudhari, “A Review on Smart Sensors Based Monitoring System for Agriculture,” International Journal of Innovative Technology and Exploring Engineering(IJITEE), vol. 2, pp. 76-78, March. 2013.
- (五) Izzat Din Abdul Aziz, Mohd Hilmi Hasan, Mohd Jimmy Ismail, Mazlina Mehat and

Nazleeni Samiha, “Remote Monitoring in Agricultural Greenhouse Using Wireless Sensor and Short Message Service(SMS),” International Journal of Engineering & Technology IJET - IJENS, vol.9,no.9,pp. 35-43, Oct. 2012.

(六) 行政院農業委員會, “農政與農情,” 農委會出版品,第 242 期, Aug. 2011.

(七) Tom Igoe, 2011, 「互動裝置專題製作：運用感測器、網路及 Arduino 建置互動電子裝置第二版」, 蔣大偉, 碁峯資訊股份有限公司。

(八) David Hest, “Farmstead Wi-Fi hotspots,” Farm Industry News, April. 2010

## 【評語】 052317

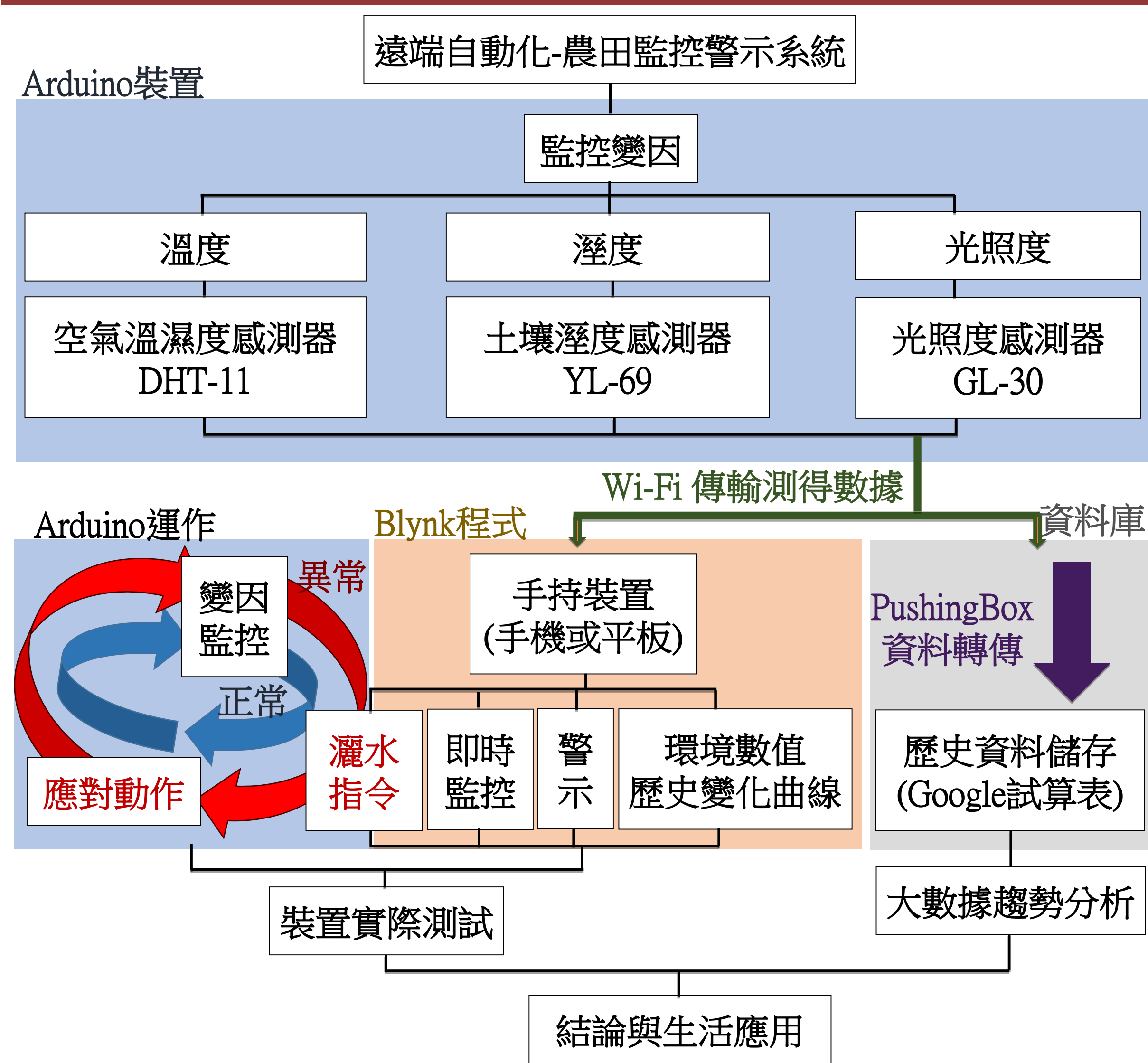
1. 本作品設計遠端農田監控警示系統，有實際場域之調查資料。
2. 本作品利用 IoT 觀念，並結合雲端功能，功能相當實用，有應用潛力。
3. 偵測之照度與市售檢測儀之數據十分相近，偵測之溫度相較於氣象局資料誤差小。
4. 計畫未來可再深入研究，用大數據及 AI 來提高農田生產管理。

# 壹、研究動機

我們曾在學校看過一部台灣紀錄片—無米樂，提及農夫辛勤播種、除草、澆水、施肥、噴灑農藥、收割，每天重複著單調繁瑣的體力活、全年無休卻收入微薄，投資報酬率低、工作辛苦。此外，現代社會農村勞動力嚴重不足、人口老化，更加深了農民的負擔。為減輕其壓力，我們設想：結合現代資訊科技，開發一套遠端監控警報系統，及時且準確監控作物生長環境。

以讓農民能簡單上手為前提，本研究選用單晶片控制器Arduino作為控制面板，在農田放置感測器及灑水器，手持裝置應用程式Blynk及Google表單分別獨立接收測得資料。感測器監控環境，另藉由灑水調節異常之環境數值，應用程式Blynk遠距即時監控農田，Google表單則作為雲端資料庫儲存歷史資料，供農民提取使用。

# 貳、研究目的



## 本研究主要方向

### (一) 監控警報系統

- 感測器控制面板(Arduino Mega板)
  - 連接感測器，測量當下「土壤溼度」、「光照度」、「空氣溫度」及「空氣濕度」；連接液晶顯示器，顯示當下測得之環境數值
  - 測得環境數值不合最適生長範圍，接收手持裝置傳回之指令時，作出應對之灑水動作(開始or停止)

### 2. 手持裝置之應用程式(Blynk應用程式)

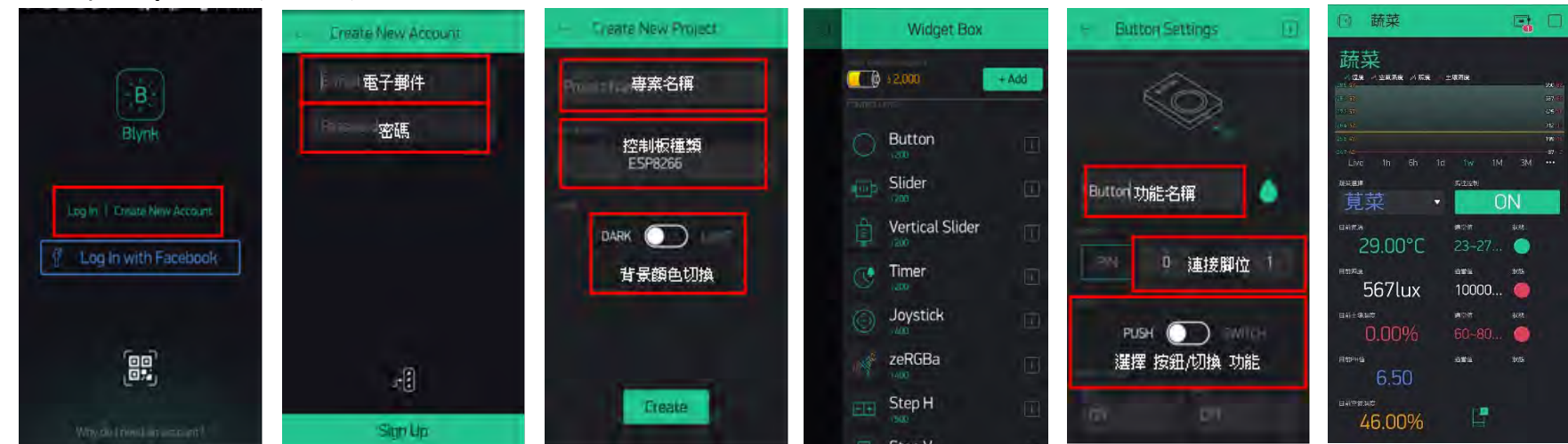
- 顯示當下測得之環境數值、將歷史數值以折線圖呈現
- 依據不同蔬菜給出不同最適生長範圍，當測得之環境數值超出範圍時，亮紅燈並發出提醒(警示音、提醒欄)

### (二) 資料庫歷史資料儲存

PushingBox網站、Google表單  
PushingBox網站接收測得之環境數值並轉傳至Google表單儲存

### (三) 趨勢分析

測得環境數值整合氣象預報、作出趨勢圖；另分析測得資料做出未來趨勢預測



# 參、研究設備及器材

馬達、馬達電機驅動模組	Arduino Mega板	麵包板、杜邦接頭	太陽能防水行動電源	液晶顯示器 2x16 LCD	溫濕度感測器 DHT11	土壤溼度計 YL-69	光強度感測器 GY-30	市售檢測儀	wi-fi晶片 ESP-01

# 肆、研究方法

## 一、設定Arduino程式

利用Arduino 1.8.2開發環境，設定好「偵測數據」、「判斷數值是否超過警戒範圍」、「手持裝置應用程式Blynk數值顯示」、「上傳雲端平台Pushing Box」及「控制馬達開關」的程式。

偵測數據	判斷與警示	手持裝置數值顯示	上傳雲端平台	馬達控制
<pre>combine 182 //在此從所有感測器接收數據 要存到上面的變數中 183 void readSensor() 184   Serial.println("timer event"); 185   moist = analogRead(MoistPIN); //讀取土壤濕度 186   moist=map(moist,250,800,0,100); 187   lux = lightMeter.readLightLevel(); 188   h = dht.readHumidity(); 189   t = dht.readTemperature(); 190 }</pre>	<pre>combine 112 if((lux &lt; vegetable[currentVegetable].minLux    lux &gt; vegetable[currentVegetable].maxLux)    113    (moist &lt; vegetable[currentVegetable].minMoist    moist &gt; vegetable[currentVegetable].maxMoist)    114    (t &lt; vegetable[currentVegetable].minTemp    t &gt; vegetable[currentVegetable].maxTemp)) 115   digitalWrite(LED_PIN, HIGH); 116   Serial.println("Warning!"); 117   digitalWrite(LED_PIN, LOW); 118   delay(1000); 119 }</pre>	<pre>combine 107 void sendData() { 108   Blynk.virtualWrite(V11, t); 109   Blynk.virtualWrite(V12, h); 110   Blynk.virtualWrite(V13, lux); 111   Blynk.virtualWrite(V14, moist); 112 }</pre>	<pre>combine 131 //-----PushingBox Start 132 WiFiClient client; //Instantiate WiFi object 133 if (client.connect(WEBSITE, 80)) 134   client.print("GET /pushingbox?device=" + device 135              + "&amp;name=" + (String) h 136              + "&amp;name" + (String) lux 137              + "&amp;t" + (String) t 138              + "&amp;moist" + (String) moist 139              ); 140   client.println(" HTTP/1.1"); 141   client.println("Host: "); 142   client.println(WEBSITE); 143   client.println("User-Agent: ESP8266/1.0"); 144   client.println("Connection: close"); 145   client.println(); 146 }</pre>	<pre>//在手機上選擇蔬菜時會在這裡收到訊息 147 digitalWrite(LED_PIN, HIGH); 148 if (value == 1) { 149   digitalWrite(LED1, HIGH); 150   //digitalWrite(LED2, LOW); 151   digitalWrite(PWMA, 100); 152   Serial.println("Turn motor ON."); 153 } else { 154   digitalWrite(LED1, LOW); 155   //digitalWrite(LED2, LOW); 156   digitalWrite(PWMA, 0); 157   Serial.println("Turn motor OFF."); 158 }</pre>
讀取土壤濕度(moist)、光照度(lux)、空氣濕度(h)、氣溫(t)	判斷偵測到的環境數值：若數值大於或小於作物最適生長範圍便發出警示	(上) 偵測數據 手持裝置端顯示  (下) 警戒範圍 手持裝置端顯示	將數據上傳至雲端平台Pushing Box	按下「on」/「off」，做出「灑水」之應對措施

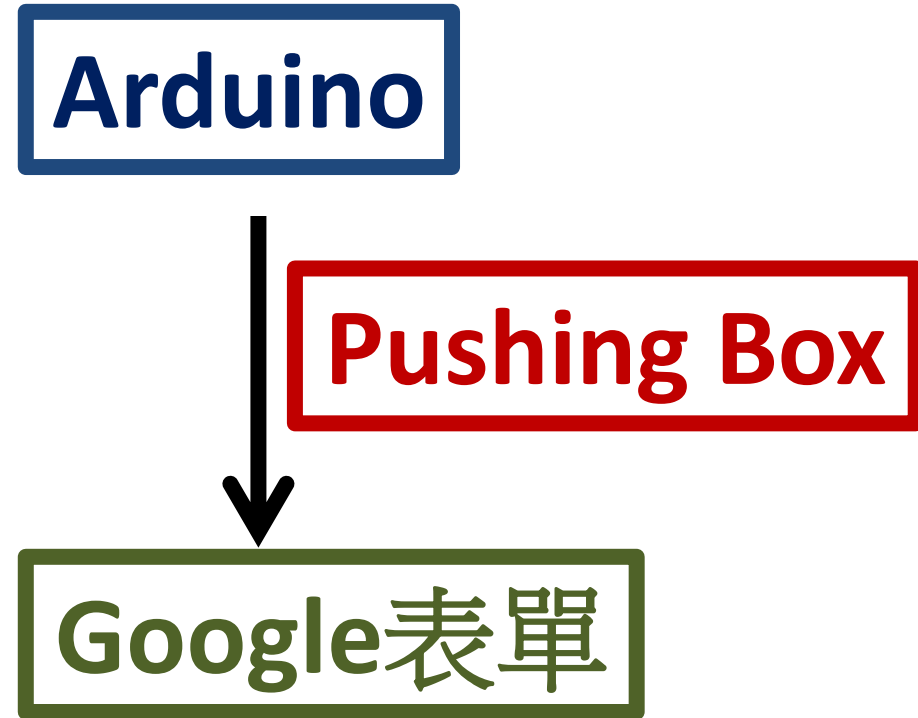
## 二、雲端歷史資料儲存

### (一) 雲端平台PushingBox的應用

PushingBox為少數能與Google相容的網站，由於Arduino所偵測之數據無法直接傳送至Google表單，於是我們藉由PushingBox作為兩者之間的橋樑，將資料完整傳輸。

### (二) 建立Google表單

標示欄位名稱，為了明確接收數據，須到「回覆/取得預先填入之網址」，此網址即是數據接收的目的地。



## 三、手持裝置應用程式Blynk監控與警示

為了在手機端能清楚明瞭地即時監控，需先建置蔬菜選單、監控視窗、趨勢圖表、提醒、狀態顯示以及控制灑水的功能。

選單設置	數據監控視窗	圖表製作設定	手持裝置提示音	狀態顯示	控制灑水
設定8種蔬菜供農夫選擇	接收Arduino測得的數據	建立趨勢圖，設定作圖的細項	當超出生長適當範圍，發出提醒至手持裝置	當數值未在生長安全範圍內，亮紅燈；反之亮綠燈	控制灑水與否

## 四、裝置實際測試

本研究裝置與監控畫面	市售檢測儀數值(比對用)

我們實際到農田內以市售檢測儀檢測並紀錄測得環境數值，再與本實驗之手持裝置應用程式監控視窗顯示之數值做比較。



# 伍、研究結果

## 一、Arduino裝置成果圖

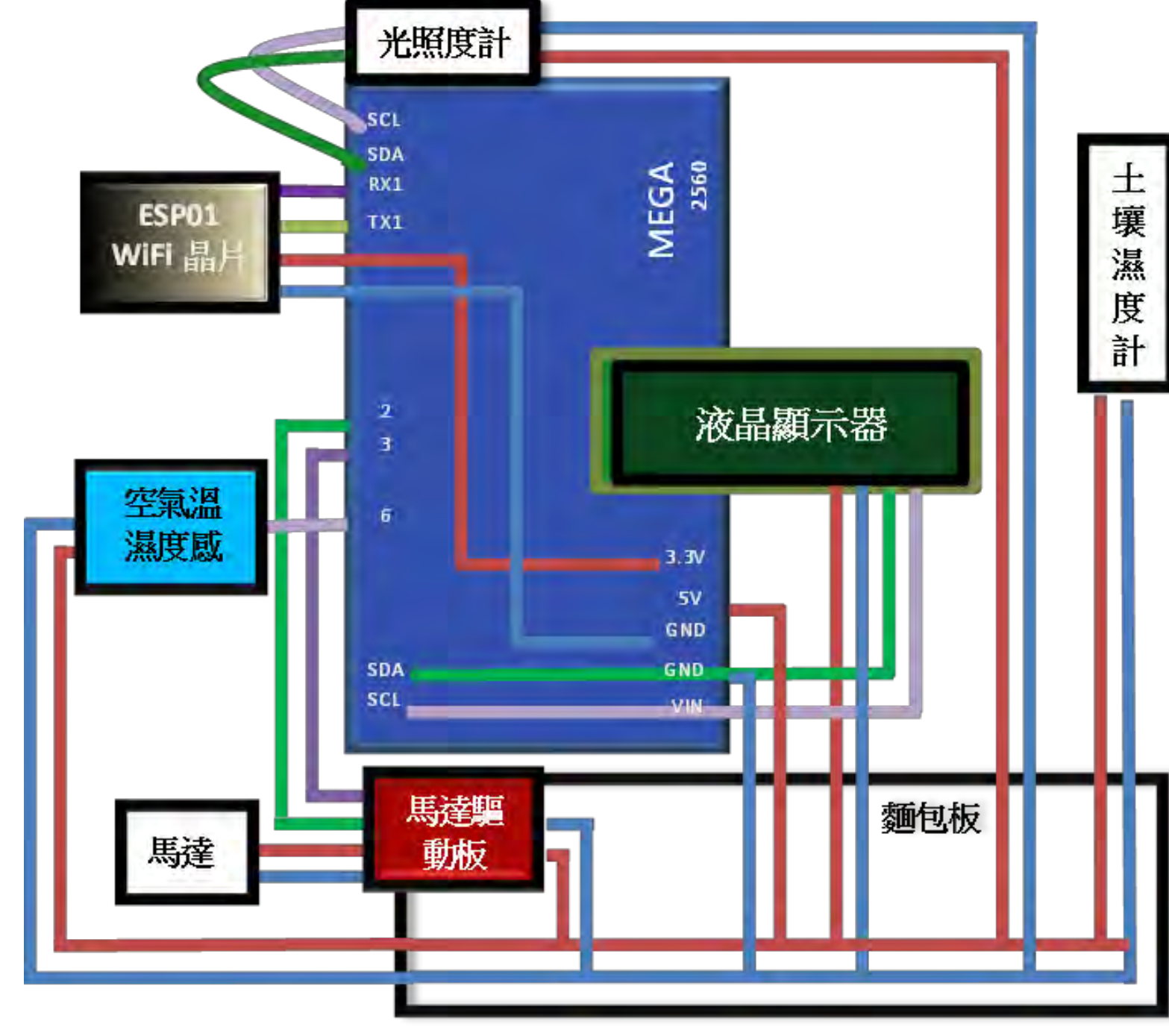
本研究系統示意圖及成品圖，感測器連接腳位配置如下表所示：

器材	腳位
土壤溼度感測器 (YL-69)	5V、GND
液晶顯示器 (LCD2x16)	SCL、SDA、VCC、GND
空氣溼度感測器 (DHT-11)	D6、5V、GND
光照度感測器 (GY-30)	SCL、SDA、VCC、GND
馬達電機驅動模組 (TB6612FNG)	D2、D3、GND、5V

(一)使用面板：Arduino Mega 2560

(二)液晶顯示器：

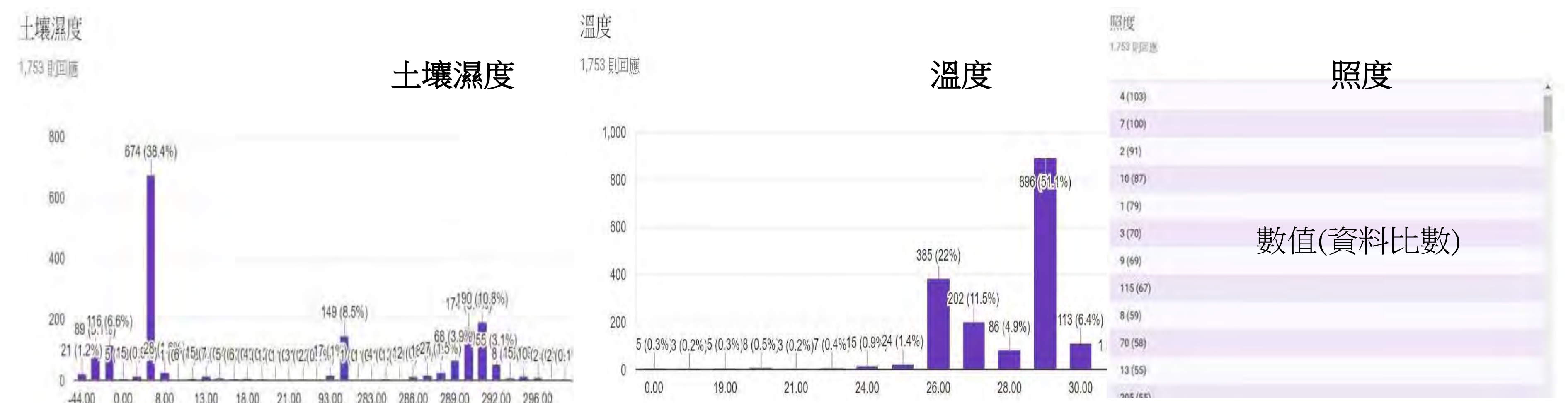
1. 土壤濕度(M)
2. 照度(Lux)
3. 空氣溫度(T)



## 二、數據經WiFi無線模組傳送至雲端儲存，於遠端電腦進行資料檢視，提升農作物管理效能

(一)儲存Google表單之歷史資料可輕易於網頁檢視或下載，進行趨勢分析或分享農作物環境資料給農民或專家分析。

縱軸	時間
橫軸	偵測資料
圖說文字	(前)資料個數實際值
	(後)所占百分比(%)



(二)試算表中，可將各資料繪製成簡化的統計圖表，以供大數據統計的應用。

1	時間戳記	空氣溼度	照度	溫度	土壤溼度
1121	2018/6/19 上午 11:27:23	46	59462	31	64
1122	2018/6/20 上午 11:22:44	48	53633	30	53
1123	2018/6/21 上午 11:25:31	47	49863	31	72
1124	2018/6/22 上午 11:25:46	48	53249	28	56
1125	2018/6/23 上午 8:05:23	56	53589	27	60
1126	2018/6/24 上午 8:05:27	57	46593	26	40
1127	2018/6/25 上午 2:19:35	36	59313	29	76
1128	2018/6/26 上午 2:19:44	46	42655	29	29
1129	2018/6/27 上午 11:15:22	46	51654	29	63
1130	2018/6/28 上午 11:10:46	49	53561	30	61



## 三、手持裝置應用程式Blynk之應用

- (一)顯示作物生長環境之即時監控數據。
- (二)選擇視窗可選擇不同蔬菜，根據所選擇的不同蔬菜，其最適生長範圍會隨之變動。
- (三)目前環境狀態若超出該蔬菜最適生長之範圍則進行警示。
- (四)將已測得數值繪成圖表，供農夫根據環境變化作好應對措施。
- (五)遠端操控馬達(灑水)。

**選擇蔬菜種類 (8種)**

**即時數據監控**  
(1)空氣溫度  
(2)光照度  
(3)土壤濕度  
(5)空氣溼度

**蔬菜選擇欄**

**歷史變化曲線**

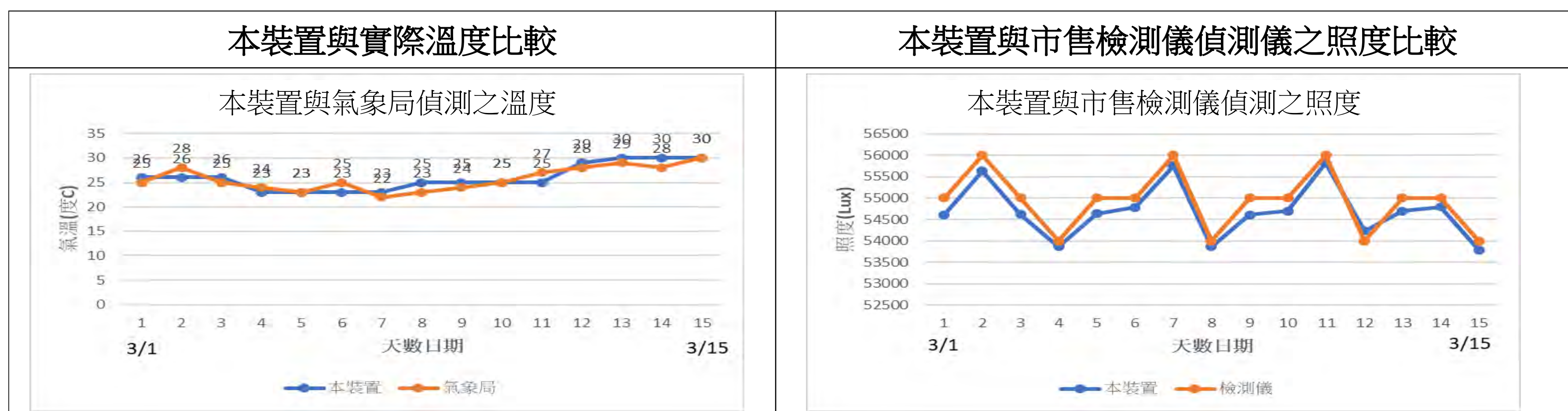
**控制灑水**

**數值警戒：**  
(1)空氣溫度  
(2)光照度  
(3)土壤濕度  
若超出最適生長範圍，亮紅燈

**手持裝置提醒：**  
超出土壤濕度警戒範圍時，發出提醒至手持裝置、警示音響起

## 四、裝置實際測試與數據比對

以薺菜(空心菜)生長期約15天為例，由兩張折線圖可發現，兩者的實驗組及對照組趨勢都十分相近，因此本研究裝置具一定準確性。



## 五、趨勢預測

本研究結合中央氣象局與自行測得的數值，匯入試算表，將過去及未來氣溫資料整合，作出未來趨勢的變化曲線。

**函式(資料擷取)**

**中央氣象局未來一周氣溫**

**過去3日氣溫及未來7日氣溫整合之趨勢線**

## 六、裝置設計

設計出最小體積裝置，精確計算並以3D列印製出外盒。

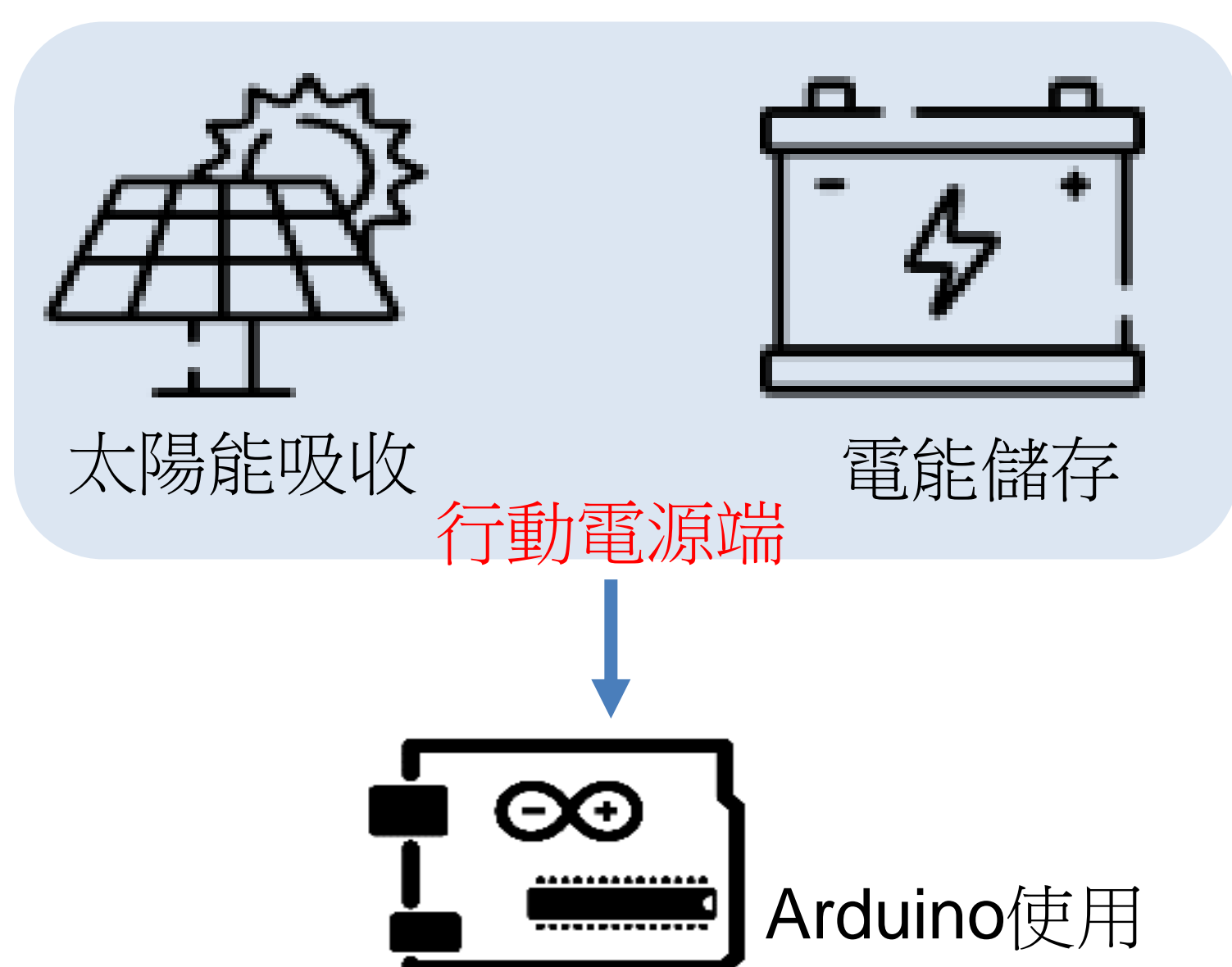


# 陸、討論

## 一、問題探討

### (一) 供電設備選用

最初以行動電源供電，具可充電、體積小之優點，缺點為須定時充電。後來我們以太陽能防水行動電源作為供電設備，除兼具前述之優點外，裝置更可自力運作，經過計算後，一個充飽電的電池，約可供應**15.37**個小時。



器材	電量(mAh)
土壤濕度感應器模組 (YL-69)	3
液晶顯示器 (LCD)	85
溫濕度傳感器 (DHT-11)	1.5
光照檢測傳感器 (GY-30)	1.7
Arduino裝置 (Mega板)	50
WiFi晶片 (Esp01)	50
<b>總電量</b>	<b>221.2</b>
<b>太陽能行動電源</b>	<b>3400</b>

$$3 + 85 + 1.5 + 1.7 + 50 + 50 = 221.2$$

↑ 土壤 LCD 溫度 光照度 Mega WiFi 晶片

$$3400 \div 221.2 = 15.37$$

↑ 太陽能 裝置使用 電源

### (二) 偵測環境數值之單位

原先土壤濕度感測器(YL-69)感測的數值單位較少被使用，所以我們撰寫程式轉換單位，以百分比的形式呈現資料，方便估算變化趨勢。

### (三) 利用免費、普遍的網站自動傳輸及儲存歷史數值

我們利用：Google表單填寫完畢後會自動將填寫內容儲存至Google試算表的特性，填寫測得環境數值至表單，以Google試算表作為歷史資料庫。

而由於Google有自我防禦機制，只有和它「相容」的網站才能使用表單，而Arduino不在此列。而PushingBox則是和Google相容的、可免費接收與傳輸訊息的網站，在Arduino程式中將環境數值傳輸至PushingBox後，由PushingBox自動填寫表單，後Google表單的Google試算表將歷史數值作出折線圖。

因此得以儲存數據至歷史資料庫，並可作得變化趨勢圖，供農夫作更多應用。

## 二、生活應用

### (一) 農田環境與地區狀況即時比對(得知農田異常狀況)

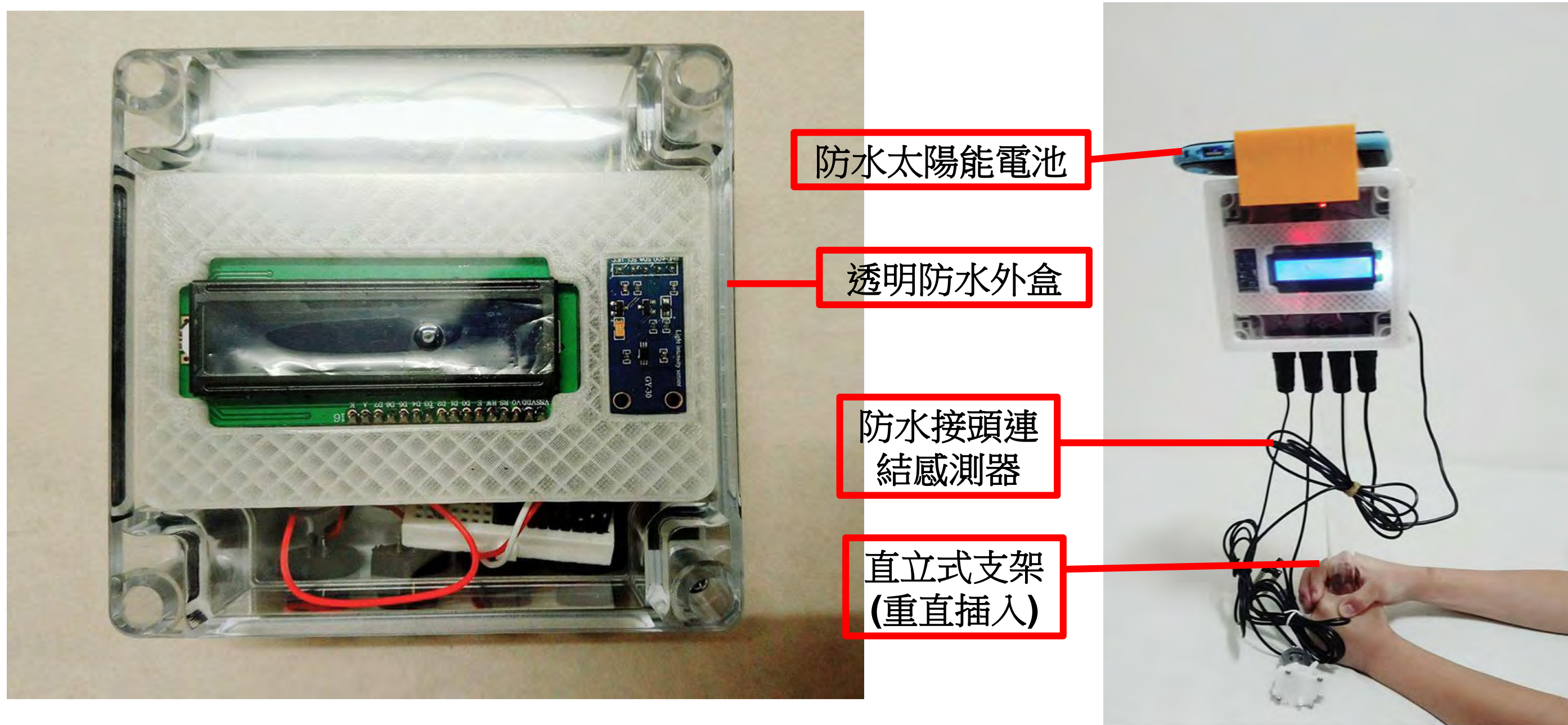
除了現有感測器測得之農田環境數值作出歷史趨勢圖，加入將氣象局測得之地區環境狀況，一同繪製成趨勢圖兩線對比，了解單一農田環境相對於大範圍地區之異常變化。

### (二) 大數據預測圖表

藉由物聯網串連多個農田之環境數值，藉由函式分析變化趨勢。加入預測環境變化的功能，整合天氣預報資料，將兩者的空氣溫溼度做成圖表，達到比對、預防天然災害的功效。

### (三) 遠端操作指令(灑水)

在遠端即時操控Arduino，直接控制馬達的開關(灑水與否)，不但可省去親自澆水的時間，當環境數值有狀況時，也能即時作出應對



器材	價錢(NT)
液晶顯示器 (LCD)	85
土壤濕度感應器模組 (YL-69)	75
光照檢測傳感器 (GY-30)	110
溫濕度傳感器 (DHT-11)	50
Arduino裝置 (Mega板)	250
程式模組 (Blynk)	130
太陽能行動電源	750
馬達及馬達驅動器	120
Wifi晶片 (esp01)	150
<b>總價</b>	<b>1570</b>

# 柒、結論

## 一、監控警示系統

農夫原本需定時親自至農田了解作物情況，本研究藉由環境數值感測器全面監控農田。

### (一) 感測器控制面板(Mega板)

- 藉ESP-01之Wi-Fi傳輸，可自農田之Arduino裝置以Wi-Fi大範圍傳送測得數值至遠距監控裝置，節省時間與體力。
- 感測器與灑水裝置配合，調節環境變因，將之控制在適合植物生長的範圍內。

### (二) 手持裝置之應用程式(Blynk應用程式)

- 顯示當下之作物環境資訊供農夫判讀以進行下一步工作，助農夫更精準判斷。
- 根據不同蔬菜給出各自最適生長範圍。
- 隨時比對當下測得之環境數值是否超出最適生長範圍，自動發出警示。
- 將環境數值3個月內之測得歷史數據(來自Arduino)繪製成折線圖，以得知過去環境數值變化趨勢



## 二、資料庫歷史資料儲存

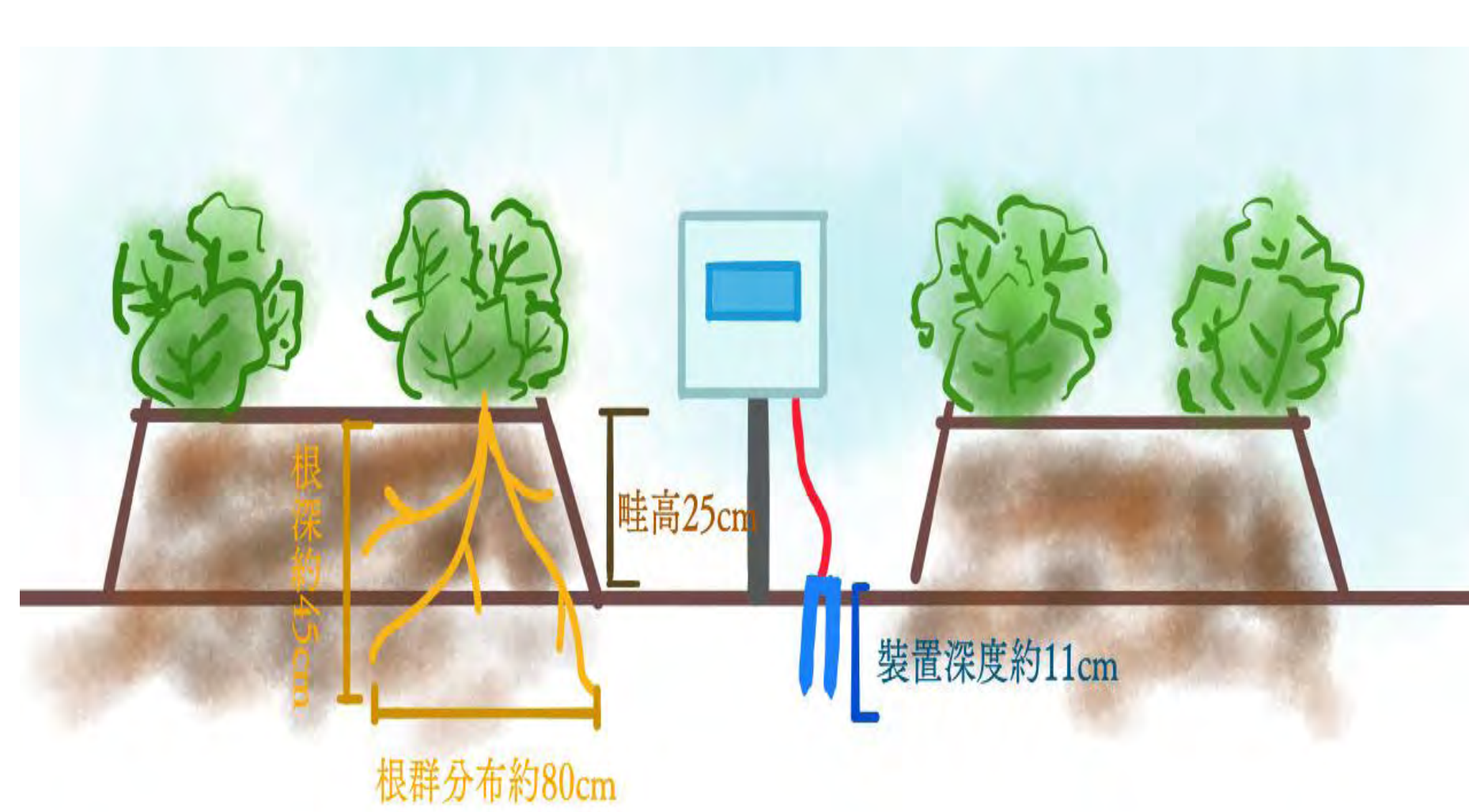
### (一) PushingBox網站、Google表單

- 以免費、普遍、簡單為前提，以Google表單作為雲端資料庫，將資料分享範圍擴展至無線網路。未來可參考比對，找出環境數值變化週期規律以做應對。
- 以和Google相容的雲端資料推送平台PushingBox作為資料傳輸中繼站，將Arduino裝置測得環境數值填入Google表單儲存。

## 三、未來趨勢預測

本研究分析Aruiino裝置測得數據，匯入Excel預測未來三日環境數值可能變化之範圍，供農夫提早作出適切的應對措施。

**本研究建立一套農田遠端監控警示系統，可使農夫節省時間並有效率的管理農田**



# 捌、參考資料

一、廖文淵等人 (2014)。「以Arduino發展平台為基礎之智慧生活監控」。德霖學報, (27)。  
二、Swarup S. Mathurkar, & D. S. Chaudhari, ( March. 2013)。「A Review on Smart Sensors Based Monitoring System for Agriculture, "International Journal of Innovative Technology and Exploring Engineering(IJITEE), 2, 76-78。」