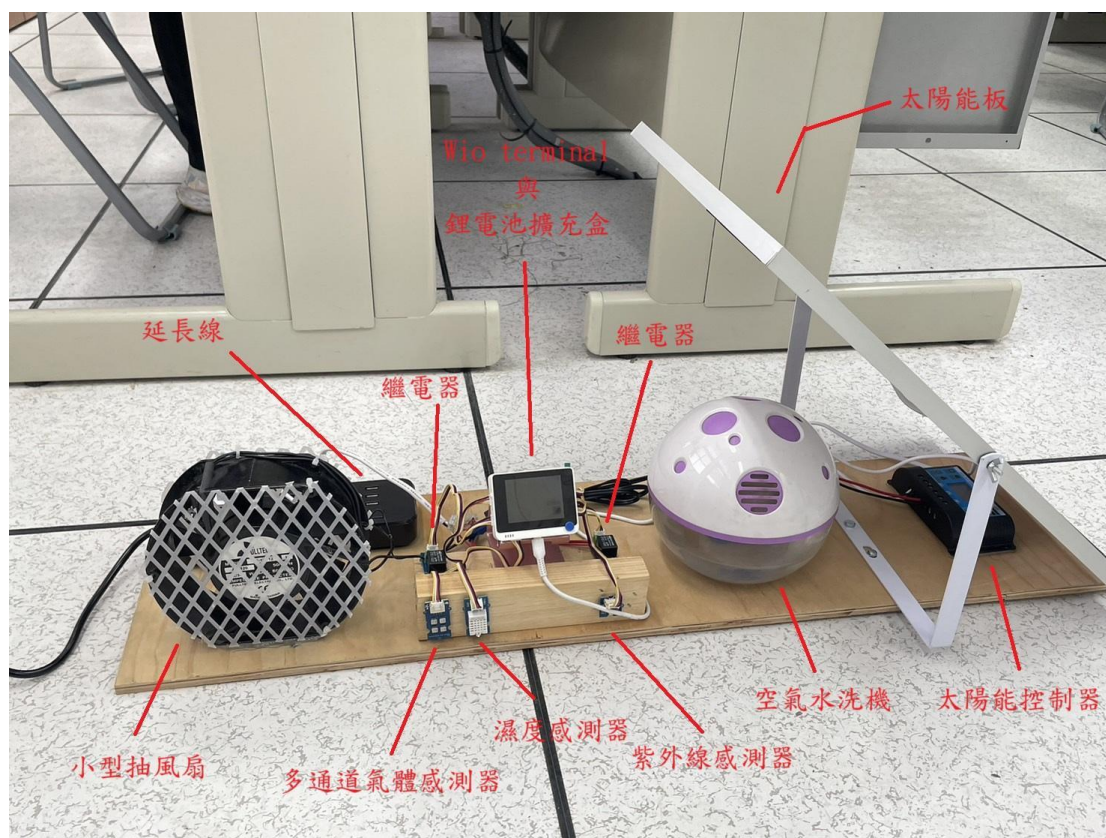


全國高級中等學校專業群科 113 年專題實作及創意競賽
「創意組」作品說明書



群 別：電機與電子群

參賽作品名稱：新時代農舍 AI 幫手

關鍵詞：農藥、人工智慧、太陽能

目錄

摘要：	1
壹、創意動機及目的	1
貳、作品特色及創意特質	2
參、創意發想與研究過程	2
肆、設計相關原理	3
一、氣味訓練方法	3
二、使用材料及程式	5
伍、作品功用與操作方式	10
一、作品功用	10
二、操作方式	10
陸、製作歷程說明	12
一、多通道空氣感測器的 AI 資料集	12
二、安裝 UV 與 DHT 模組	13
三、撰寫程式	14
四、測試結果並撰寫報告	16
柒、附錄：作品分工表	17
捌、附錄：競賽日誌	17

圖目錄

圖 1	熱傷害介紹.....	1
圖 2	紫外線等級比較.....	2
圖 3	搭配 edge impulse，收集農藥味道	3
圖 4	進行分類.....	3
圖 5	特徵轉換：神經網路(Neural Network)與深度學習(Deep Learning)	3
圖 6	產生特徵值.....	4
圖 7	設定神經網路層.....	4
圖 8	放入測試資料，驗證此模型可用程度.....	4
圖 9	打包上傳硬體，並進行 AI 判讀，檢測吸入農藥是否超標	4
圖 10	Wio Terminal.....	5
圖 11	鋰電池擴充盒.....	5
圖 12	多通道氣體感測器.....	6
圖 13	溫溼度感測器.....	6
圖 14	紫外線感測器.....	6
圖 15	繼電器模組.....	7
圖 16	水洗機.....	7
圖 17	小型抽風扇.....	7
圖 18	太陽能控制器.....	8
圖 19	太陽能版.....	8
圖 20	延長線.....	8
圖 21	ArduinoIDE.....	9
圖 22	Node-red	9
圖 23	MQTT	9
圖 24	LINE Notify	10
圖 25	UV 值讀取顯示	10
圖 26	溫溼度讀取顯示.....	10
圖 27	四種氣體讀取顯示.....	11
圖 28	判讀農藥狀態顯示.....	11

圖 29	水洗機與風扇按鈕.....	11
圖 30	顯示水洗機與風扇啟動.....	11
圖 31	LINE 通知顯示.....	12
圖 32	農藥一.....	12
圖 33	農藥二.....	12
圖 34	農藥三.....	12
圖 35	上傳 Ede impulse 網站訓練.....	13
圖 36	取得資料下載到 wio terminal	13
圖 37	安裝 DHT 模組	13
圖 38	安裝 UV 模組	13
圖 39	MQTT 定時連線	14
圖 40	判斷 Node-red dashboard 水洗機、風扇開關按鈕是否按下	14
圖 41	判斷溫度過高風扇自動啟動.....	15
圖 42	判斷紫外線照度（UV 值）過高自動啟動警報.....	15
圖 43	農藥數據推論程式段.....	15
圖 44	顯示多通道感測器數值部分程式.....	16
圖 45	數據查看.....	16
圖 46	傳送通知.....	16

摘要：

新時代農舍 AI 幫手主要在應對農藥與除草劑對人體健康的危害。透過紫外線與溫度檢測功能，我們能夠評估農地的紫外線強度，同時監測空氣中的溫度，提醒農民合理安排工作時間。搭配人工（AI）智慧技術，機器即時檢測農藥、除草劑含量，結果顯示於螢幕上，並啟動警報以通知農民。同時，繼電器啟動風扇與空氣水洗機，避免農藥吸入並淨化空氣，提供全方位的農業安全保障。

壹、創意動機及目的

當農民使用農藥噴灑農作物時，殘留在空氣中的農藥可能會隨著風向，吹往住宅，然而我們卻無法感知到。這些殘留在空氣中的農藥，如果長期由皮膚接觸或者吸入到體內，將會有中毒的危險，甚至對人體健康會造成長期的危害，如癌症、生殖系統以及肝、腦和身體其他部位損害。

且由於農民多在無遮蔽物的農田工作，不僅需要面對炎熱潮濕的天氣，還會受到紫外線的照射。長期待在高溫狀態，可能導致心肌梗塞或腦中風，或是多重器官衰竭死亡，而中暑的死亡率則超過 30%。當紫外線數值超過 6 後，將會導致人體的傷害，如曬傷、皮膚癌等等。

因此在這樣的情境下，我們需要創新的解決方案，以保障農民的健康和提高農業生產效率。



圖 1 熱傷害介紹

	UVA	UVB
波 長	短波UVA:320-340nm 長波UVA:340-400nm	290-320nm
到達地表的比例	95%	5%
傷害力	能量可穿透至皮膚的真皮層	到達皮膚的表皮層
後遺症	<ul style="list-style-type: none"> 引起光老化現象：細紋、皺紋產生及皮膚鬆弛 造成皮膚曬黑，形成斑點，色素沉著 造成皮膚敏感，耐受力下降 	<ul style="list-style-type: none"> 造成皮膚曬傷、曬紅 造成皮膚曬黑
特 點	<ul style="list-style-type: none"> 傷害皮膚時不會有灼熱感，因此很容易被忽略 可穿透雲層及玻璃，即使陰天或室內仍會受到傷害 從早到晚UVA的能量都很高 	<ul style="list-style-type: none"> UVB對皮膚的傷害集中於早上10點到下午2點

圖 2 紫外線等級比較

貳、作品特色及創意特質

一、人工智慧技術

結合了人工智慧技術，能夠即時檢測農藥、除草劑含量並將檢測結果即時顯示於螢幕上。

二、農藥警報提醒

當機器檢測到農藥超標，可以立即發起警報，警示周遭有超標農藥摻雜在空氣中的危險，讓農民可以即時做出反應，預防農藥與除草劑的潛在危害。

三、紫外線與溫度提醒

透過紫外線與溫度檢測功能，裝置評估農地的紫外線強度，同時監測空氣中的溫度。

四、螢幕顯示

監測的同時顯示在機器的 TFT 螢幕與電腦 dashboard，能讓農舍得成員不用出門便能知道附近狀況，減少危險因子。

參、創意發想與研究過程

當初想要製作這個作品的原因是，因為發現住在農田附近的居民日常生活中，不管是身體還是呼吸都是直接暴露在帶有超標農藥的空氣中，長期以來對身體是一中極大的負擔。溫度與紫外線也是危險因子，所以我們增加了溫度、紫外線的偵測與報警功能，超標時會啟動風扇與水洗機，減少對於身體的危害。太陽能板提供綠能充電，提升能源的使用效率。

肆、設計相關原理

一、氣味訓練方法

- (一) Wio terminal 上傳單筆資料到 Edge Impulse，並把資料整理到不同類別，反覆操作。

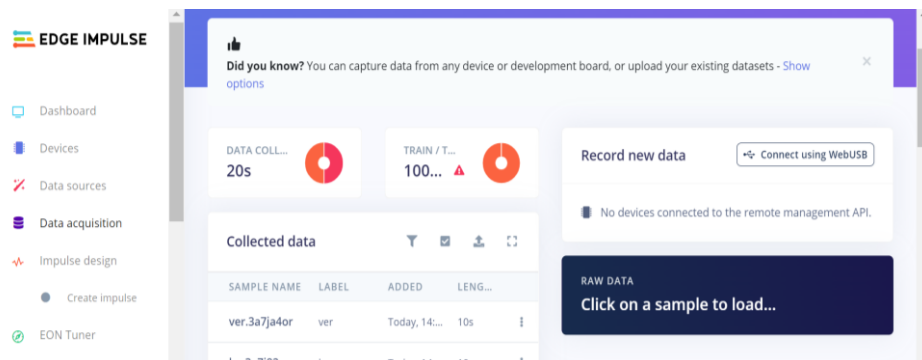


圖 3 搭配 edge impulse，收集農藥味道

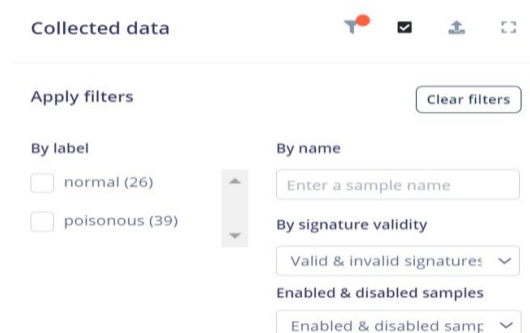


圖 4 進行分類

- (二) 設計 Impulse：資料處理、定義模型。測試神經網路效能、修改、重複。

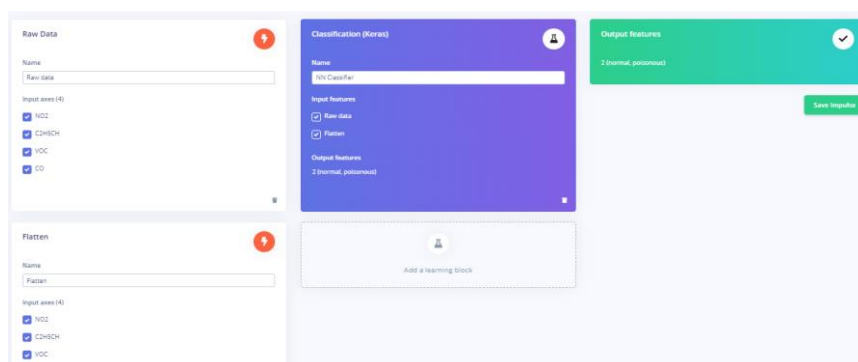


圖 5 特徵轉換：神經網路(Neural Network)與深度學習(Deep Learning)

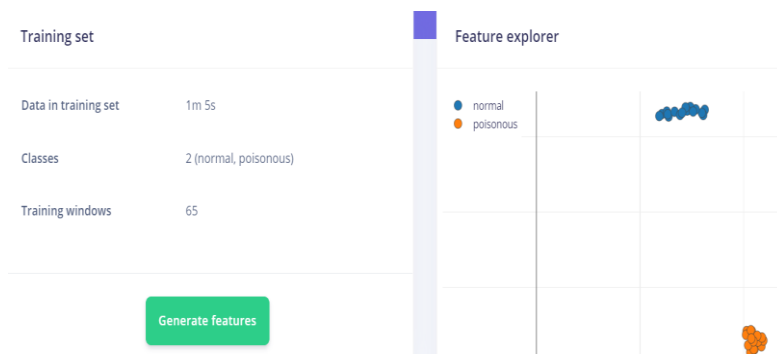


圖 6 產生特徵值

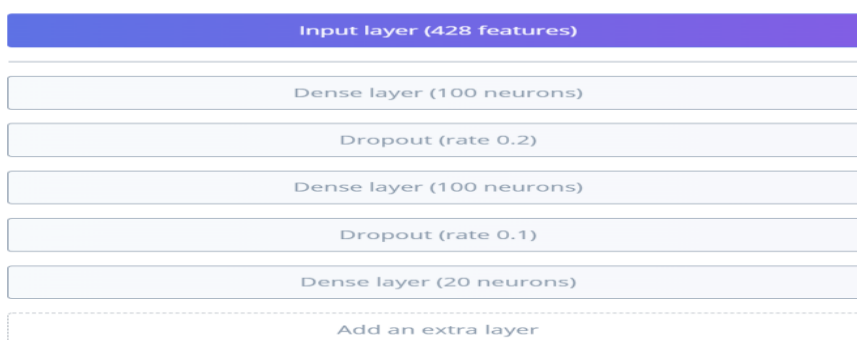


圖 7 設定神經網路層

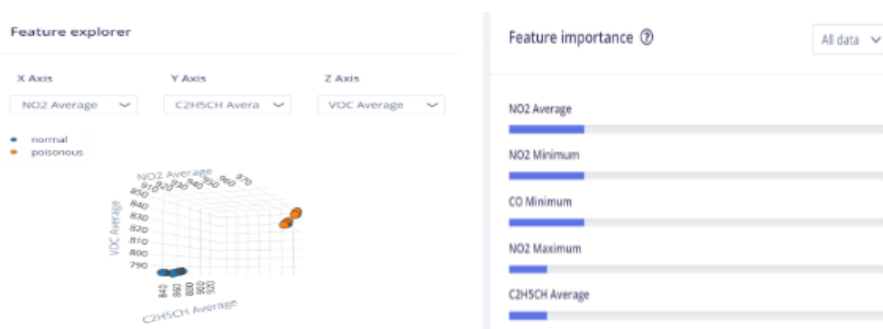


圖 8 放入測試資料，驗證此模型可用程度

(三) 接著匯出神經網路檔案，再透過 Arduino IDE 燒錄至 Wio terminal，最後讓 Wio terminal 執行離線推論。

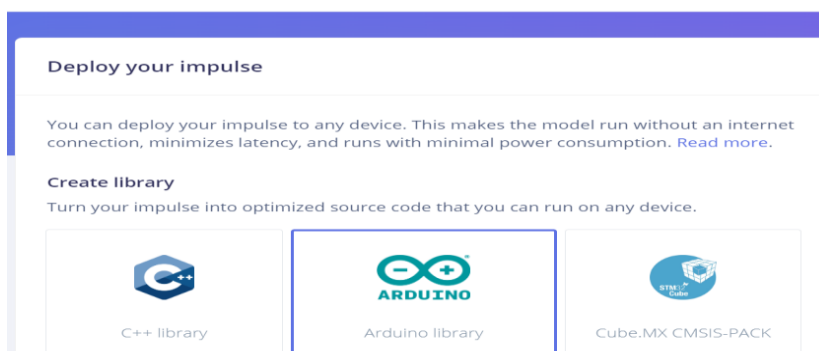


圖 9 打包上傳硬體，並進行 AI 判讀，檢測吸入農藥是否超標

二、使用材料及程式

(一) 硬體

- 1、Wio Terminal：Wio Terminal 開發板 與 Arduino 和 MicroPython 兼容，是基於 ATSAM51 的微控制器，具有通過 RealtekRTL8720DN 支持的藍牙和 Wi-Fi 連接。集成了 2.4 英寸液晶顯示屏，板載 IMU，麥克風，蜂鳴器，microSD 卡插槽，光傳感器，紅外發射器（IR 940nm）。Wio Terminal 是一款專門用於 IoT 與 TinyML 的多功能開發板包含了 ATSAM51P19 晶片並以 ARM Cortex-M4F 為核心 (20MHz)，已支援多種針對微控制器的 ML 推論框架。

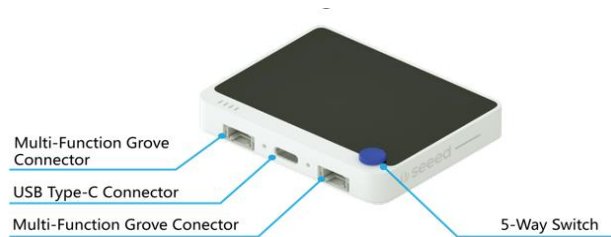


圖 10 Wio Terminal

- 2、鋰電池擴充盒：內置 650mAH 鋰電池，和 6 個 Grove 接口，是 Wio Terminal 的必備擴展板，以便增加更多附件。USB Type-C 充電，4 個 Grove 模擬/數字端口，1 個 Grove I2C 端口，1 個 Grove UART 端口，磁鐵隱藏在外殼內，因此可以將其粘貼在白板上。

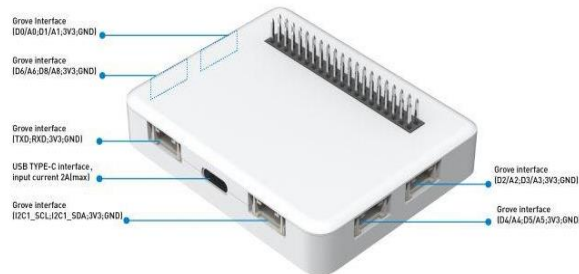


圖 11 鋰電池擴充盒

- 3、多通道氣體感測器：用於檢測空氣中的有毒物質，能夠同時檢測除乙醇（C₂H₅CH₃）、一氧化碳（CO）、二氧化氮（NO₂）、揮發性有機化合物（VOC）等氣體。每個測量單元對各種氣體均敏感，因此可以在同時間獲取到四組氣味數據。並判斷出不同種類的氣體。此模塊中使用的氣體傳感器基於 MEMS 技術。不僅體積小，還能穩定測量，比定性測量更適合定性測量。

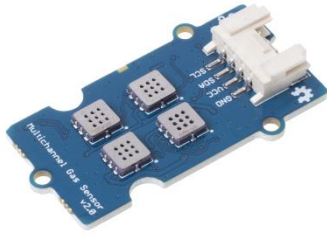


圖 12 多通道氣體感測器

- 4、溫溼度感測器：用於測量周圍環境的溫度和濕度水平。傳感器採用 DHT11，濕度測量範圍：20%-95%（0 度-50 度範圍）濕度測量誤差： $\pm 5\%$ ，溫度測量範圍：0 度- 50 度溫度測量誤差： ± 2 度，工作電壓 3.3V-5V，輸出形式數字輸出，設有固定螺栓孔，方便安裝，小板 PCB 尺寸：3.2cm * 1.4cm，每套重量約為 8g。

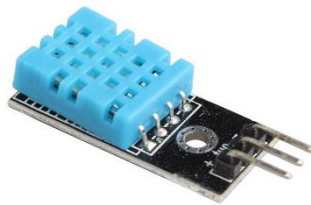


圖 13 溫溼度感測器

- 5、紫外線感測器：監視 UV 光量的應用，輸出校準的模擬電壓，基於反射率，紅外接近，紫外線指數和環境光感測器具有 I2C 數位介面和可程式設計事件中斷輸出。提供優良的寬動態範圍和各種光源包括陽光直射。隨 UV 光強度而變化，能夠檢測周圍環境中的紫外線輻射水平，通常用於監測日照強度、警示皮膚曝曬過度等。

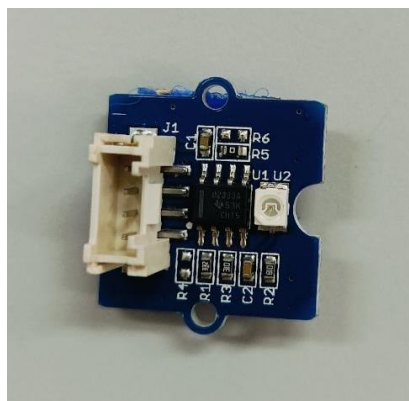


圖 14 紫外線感測器

- 6、繼電器模組：用較小的電流去控制較大電流的一種「自動開關」。由鐵芯、線圈、銜鐵、接點簧片等組成的。只要在線圈兩端加上一定的電壓，線圈中就會流過一定的電流，從而產生電磁效應。



圖 15 繼電器模組

- 7、水洗機：如偵測到空氣中的農藥濃度超標，水洗機則會自動開啟並且吹散空氣中的有毒物質的濃度，防止附近人群吸入過多有毒物質危害身體。



圖 16 水洗機

- 8、小型抽風扇：如偵測到空氣中的農藥濃度超標，風扇則會自動開啟並且吹散空氣中的有毒物質，達到降低濃度，防止吸入過多危害身體。



圖 17 小型抽風扇

- 9、太陽能充電控制器：與 12V 和 24V 系統相容。完全 4 級 PWM 充電(增強、ABS、均衡、浮動)、12V (780W) / 24V (1560W) 輸入，雙 USB 能為電子產品提供 5V 2A。直觀的 LCD 顯示屏，能清楚地顯示狀態和數據，能夠切換模式和參數配置，適用於家庭、工業、商業等。具有短路保護、開路保護、反向保護、過載保護，提高系統效率並延長電池的使用壽命。



圖 18 太陽能控制器

10、太陽能板：轉換效率可達到 20.9 %，最高輸出功率可達 380/385 瓦，9 柵線技術，可有效地限制電池隱裂範圍，提高可靠度，擁有抗高濕、抗鹽霧的高耐候模組，優異抗強風能力，IP-68 頂級封膠接線盒及 MC4 EVO2 原廠連接器。



圖 19 太陽能版

11、延長線：提供裝置所需要的電力。



圖 20 延長線

(二)、軟體

1、Arduino IDE：是一個開源嵌入式硬體平臺，用來供使用者製作可互動式的嵌入式專案。此外也是一個開源硬體和開源軟體的公司，同時兼有專案和使用者社群。該公司負責設計和製造 Arduino 電路板及相關附件。



圖 21 ArduinoIDE

- 2、Node-red：Node-RED 是一個物聯網程式設計工具，它以新穎和有趣的方式，將硬體設備、應用介面和線上服務連接到一起。提供了一個基於瀏覽器的程式設計環境和豐富的節點類型，使流程的創建變得非常容易，而且還為流程的運行提供了運行環境，實現了一鍵部署的能力

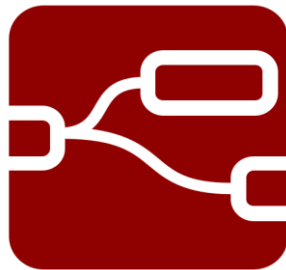


圖 22 Node-red

- 3、MQTT：是一種以標準為基礎的訊息傳遞協定或規則集，用於機器對機器的通訊。智慧型感應器、可穿戴裝置和其他物聯網 (IoT) 裝置，通常須透過頻寬有限的資源受限網路來傳輸以及接收資料。IoT 裝置使用 MQTT 進行資料傳輸，因為它易於實作並且能有效地傳輸 IoT 資料。MQTT 支援裝置到雲端和雲端到裝置間的訊息傳遞。



圖 23 MQTT

- 4、LINE Notify：是一個能與網站服務連動完成後，透過 LINE 所提供的官方帳號「LINE Notify」傳送通知。不僅可與 GitHub、IFTTT、Mackerel 等多個服務連動，也可透過 LINE 群組接收通知。只要與 LINE Notify 連動，LINE 用戶就能輕鬆地接收來自其他服務的通知。



LINE Notify

圖 24 LINE Notify

伍、作品功用與操作方式

一、作品功用

新時代農舍 AI 幫手主要是用於檢測農藥、氣溫、紫外線。觀察數據的方式也很簡單明瞭，並且太陽能板也能夠提供機台再生能源。當周遭環境有危害時，會發出警報通知農民，讓農民可以即時得知，並開啟風扇和水洗機，也可以手動或使用語音開啟風扇及水洗機。

二、操作方式

(一) 1、UV 值讀取顯示

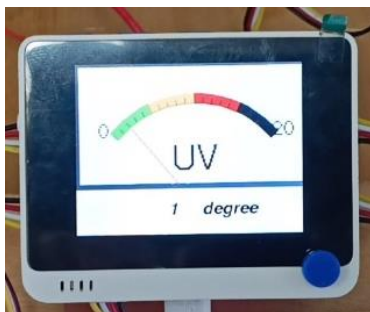


圖 25 UV 值讀取顯示

(二) 溫溼度讀取顯示

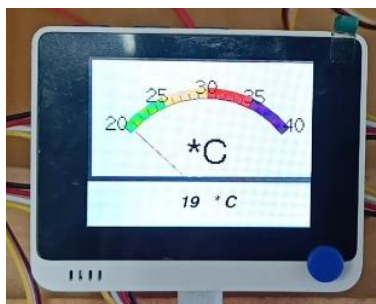


圖 26 溫溼度讀取顯示

(三) 四種氣體讀取顯示

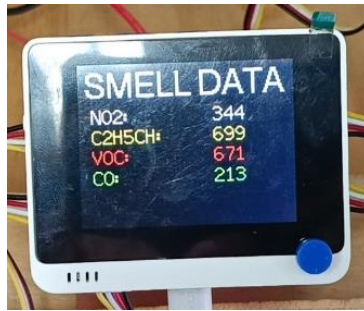


圖 27 四種氣體讀取顯示

(四) 判讀農藥狀態顯示



圖 28 判讀農藥狀態顯示

(五) 顯示水洗機與風扇啟動



圖 29 水洗機與風扇按鈕



圖 30 顯示水洗機與風扇啟動

（六）LINE 通知顯示



圖 31 LINE 通知顯示

陸、製作歷程說明

一、多通道空氣感測器的 AI 資料集

（一）手動收集三種農藥數據

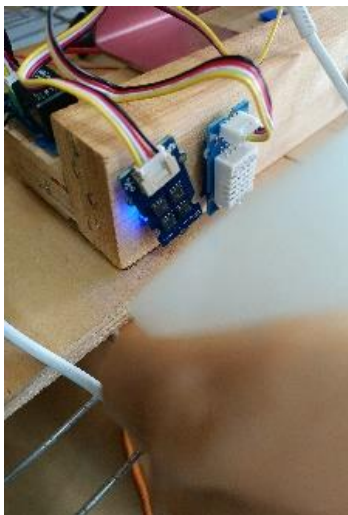


圖 32 農藥一



圖 33 農藥二



圖 34 農藥三

（二）上傳 Ede impulse 網站訓練

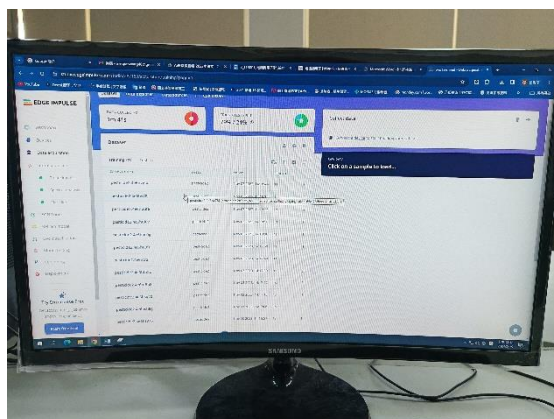


圖 35 上傳 Ede impulse 網站訓練

(三) 取得資料下載到 wio terminal

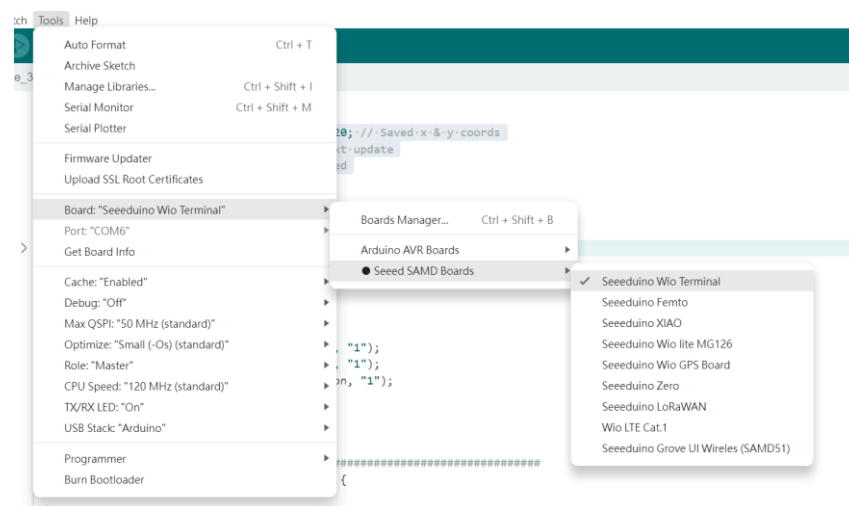


圖 36 取得資料下載到 wio terminal

二、安裝 UV 與 DHT 模組



圖 37 安裝 DHT 模組



圖 38 安裝 UV 模組

三、撰寫程式

```
1 #include "config.h"
2 #include "testmeter.h"
3
4 // ----- mqtt_reconnect() 訂閱主題 -----
5
6 boolean mqtt_nonblock_reconnect() {
7     boolean doConn = false;
8
9     if (!mqttClient.connected()) {
10         // attempts to reconnect every [MQTT_RECONNECT_INTERVAL] milliseconds without blocking the main loop.
11         long currTime = millis();
12         if (currTime - lastTime > MQTT_RECONNECT_INTERVAL) {
13             lastTime = currTime;
14             doConn = true;
15             boolean isConn = mqttClient.connect(clientId);
16             //boolean isConn = mqttClient.connect(clientId, MQTT_USER, MQTT_PASSWORD);
17             char logConnected[100];
18             sprintf(logConnected, "[%s] Connect %s !", clientId, (isConn ? "Successful" : "Failed"));
19             Serial.println(logConnected);
20
21             if (isConn) {
22                 mqttClient.subscribe(MQTT_SUBSCRIBE_D1_Button);
23                 mqttClient.subscribe(MQTT_SUBSCRIBE_D2_Button);
24             }
25         }
26     }
27
28     return doConn;
29 }
```

圖 39 MQTT 定時連線

```
67 if (strcmp(MQTT_SUBSCRIBE_D2_Button, topic) == 0) //主題 PUMPIN
68 {
69     MQTT2_Setup = "";
70     for (int i = 0; i < msgLength; i++) {
71         MQTT2_Setup += (char)payload[i];
72     } // Payload: [0n]
73     if (MQTT2_Setup == "1")
74     {
75         if (D2FLAG == 1)
76         {
77             mqttClient.publish(MQTT_PUBLISH_D2_Button, "0");
78             D2FLAG = 0 ;
79         }
80         else if (D2FLAG == 0)
81         {
82             D2FLAG = 1 ;
83             mqttClient.publish(MQTT_PUBLISH_D2_Button, "1");
84             PUMPout = 1;
85             Serial.println("Publish PUMP2 On");
86         }
87     }
88     else if (MQTT2_Setup == "0")
89     {
90         mqttClient.publish(MQTT_PUBLISH_D2_Button, "0"); //送出 Off 狀態
91         Serial.println("Publish PUMP2 Off");
92         PUMPout = 0;
93     }
94 }
95 }
```

圖 40 判斷 Node-red dashboard 水洗機、風扇開關按鈕是否按下

```
176 if (second() % 23 == 0)
177 {
178     mqttClient.publish(MQTT_PUBLISH_D1_TEMP, itoa(int(Temp), buff1, 10)); //
179     mqttClient.publish(MQTT_PUBLISH_D1_Humi, itoa(int(Humi), buff2, 10));
180     mqttClient.publish(MQTT_PUBLISH_D1_UV, itoa(int(uv_Value), buff2, 10));
181     mqttClient.publish(MQTT_PUBLISH_D1_Harm, itoa(int(guessnum), buff2, 10));
182     mqttClient.publish(MQTT_PUBLISH_TEMP_Button, "0");
183     mqttClient.publish(MQTT_PUBLISH_UV_Button, "0");
184     //mqttClient.publish(MQTT_PUBLISH_HARM_Button, "0");
185     mqttClient.publish(MQTT_PUBLISH_D1_Button, "0");
186     mqttClient.publish(MQTT_PUBLISH_D2_Button, "0");
187 }
188 if (second() % 7 == 0)
189 {
190     if (Temp >= TEMPMAX)
191     {
192         if (TEMPFLAG == 1)
193         {
194             mqttClient.publish(MQTT_PUBLISH_TEMP_Button, "0");
195             TEMPFLAG = 0 ;
196         }
197         else if (TEMPFLAG == 0)
198         {
199             TEMPFLAG = 1 ;
200             mqttClient.publish(MQTT_PUBLISH_TEMP_Button, "1");
201             buzzerdanger ();
202         }
203     }
204 }
```

圖 41 判斷溫度過高風扇自動啟動

```

210         if (uv_Value >= UVMAX )
211         {
212             if (UVFLAG == 1)
213             {
214                 mqttClient.publish(MQTT_PUBLISH_UV_Button, "0");
215                 UVFLAG = 0 ;
216             }
217             else if (UVFLAG == 0)
218             {
219                 UVFLAG = 1 ;
220                 mqttClient.publish(MQTT_PUBLISH_UV_Button, "1");
221                 buzzerdanger ();
222             }
223             else if (uv_Value < (UVMIN) )
224             {
225                 UVFLAG = 0 ;
226                 mqttClient.publish(MQTT_PUBLISH_UV_Button, "0") ;
227             }
228         }
229         if (HARMFLAG == 1)
230         {
231             if (HARMSTATUS == 1)
232             {
233                 mqttClient.publish(MQTT_PUBLISH_HARM_Button, "0");
234             }
235             else if (HARMSTATUS == 0)
236             {
237                 HARMSTATUS = 1 ;
238                 mqttClient.publish(MQTT_PUBLISH_HARM_Button, "1");
239                 buzzerdanger ();
240             }
241             else if (HARMFLAG == 0)
242             {
243                 mqttClient.publish(MQTT_PUBLISH_HARM_Button, "0");
244                 HARMSTATUS = 0 ;
245             }
246         }
247     }
248 }
249
250

```

圖 42 判斷紫外線照度（UV 值）過高自動啟動警報

```

void eiinfer()
{
    ei_printf("Edge Impulse standalone inferencing (Arduino)\n");
    ei_impulse_result_t result = { 0 };
    signal_t features_signal;
    features_signal.total_length = sizeof(features) / sizeof(features[0]);
    features_signal.get_data = &raw_feature_get_data;
    EI_IMPULSE_ERROR res = run_classifier(&features_signal, &result, false /* debug */);
    ei_printf("run_classifier returned: %d\n", res);
    if (res != 0) return;
    ei_printf("Predictions ");
    ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",
        result.timing.dsp, result.timing.classification, result.timing.anomaly);
    ei_printf("\n");
    ei_printf("[");
    for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
        ei_printf("%5f", result.classification[ix].value);
    }
    #if EI_CLASSIFIER_HAS_ANOMALY == 1
        ei_printf(", ");
    #else
        if (ix != EI_CLASSIFIER_LABEL_COUNT - 1) {
            ei_printf(", ");
        }
    #endif
    }
    #if EI_CLASSIFIER_HAS_ANOMALY == 1
        ei_printf("%.3f", result.anomaly);
    #endif
    ei_printf("]\n");

    // human-readable predictions
    for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
        ei_printf("  %s: %5f\n", result.classification[ix].label, result.classification[ix].value);
        v[ix] = result.classification[ix].value;
        strcpy(S[ix], result.classification[ix].label);
        if (result.classification[ix].value > Infer)
        {
            Infer = result.classification[ix].value ;
            guessnum = Infer;
            Rank = ix ;
        }
    }

    void ei_printf(const char *format, ...) {
        static char print_buf[1024] = { 0 };

        va_list args;
        va_start(args, format);
        int r = vsnprintf(print_buf, sizeof(print_buf), format, args);
        va_end(args);

        if (r > 0) {
            Serial.write(print_buf);
        }
    }
}

```

圖 43 農藥數據推論程式段

```

399 tft.fillRect(0, 0, TFT_HEIGHT, TFT_WIDTH, TFT_BLACK);
400 //spr.fillRect(TFT_WHITE);
401 tft.setTextColor(TFT_WHITE);
402 tft.drawString("SMELL DATA", 200, 60, 2);
403 tft.setTextColor(TFT_WHITE);
404 tft.drawString("NO2:", 20, 60, 2);
405 tft.drawNumber(features[0], 200, 60, 2);
406 tft.setTextColor(TFT_YELLOW);
407 tft.drawString("C2H5CH:", 20, 90, 2);
408 tft.drawNumber(features[1], 200, 90, 2);
409 tft.setTextColor(TFT_RED);
410 tft.drawString("VOC:", 20, 120, 2);
411 tft.drawNumber(features[2], 200, 120, 2);
412 tft.setTextColor(TFT_GREEN);
413 tft.drawString("CO:", 20, 150, 2);
414 tft.drawNumber(features[3], 200, 150, 2);

```

圖 44 顯示多通道感測器數值部分程式

四、測試結果並撰寫報告

在 Wio Terminal 上傳檢測的資料後，我們可以很清楚的察看數據變化，當紫外線超標時，會發出警報提醒，當開啟電風扇與水洗機時，也會傳送通知至 LINE 群。



圖 45 數據查看



圖 46 傳送通知

柒、附錄：作品分工表

參賽學生	工作任務
A	收集資料、設計外型、影片拍攝、撰寫報告
B	收集資料、設計外型、程式設計、影片拍攝、撰寫報告
C	收集資料、設計外型、程式設計、影片剪輯

捌、附錄：競賽日誌

年	月	日	進度	紀錄	工作分配
112	11	25~31	討論方案、收集資料	地點：學校 器材：電腦 時間：14 小時	同學 A：查資料、討論 同學 B：查資料、討論 同學 C：查資料、討論
112	12	01~06	討論方案、收集資料、購買材料	地點：學校 器材：電腦 時間：8 小時	同學 A：查資料、討論、 購買材料 同學 B：查資料、討論、 購買材料 同學 C：查資料、討論、 購買材料
112	12	07~31	討論方案、收集資料、設計外型、設計外型、組裝機台	地點：學校 器材：電腦 時間：42 小時	同學 A：查資料、討論 同學 B：設計外型、討論 同學 C：設計外型、討論
113	01	01~19	討論內容、收集資料、設計外型、組裝機台、撰寫程式	地點：學校 器材：電腦 時間：40 小時	同學 A：討論內容、收集資料 同學 B：討論內容、收集資料、組裝機台 同學 C：設計外型、組裝機台、撰寫程式
113	01	20~31	討論內容、收集資料、組裝機台、撰寫程式、撰寫報告	地點：學校 器材：電腦 時間：52 小時	同學 A：討論內容、收集資料、組裝機台、撰寫報告 同學 B：討論內容、收集資料、組裝機台、撰寫程式 同學 C：討論內容、收集資料、組裝機台、撰寫程式

113	02	01~18	討論內容、收集資料、撰寫程式、撰寫報告、拍攝影片、影片剪輯	地點：學校 器材：電腦 時間：46 小時	同學 A：討論內容、收集資料、撰寫報告、拍攝影片 同學 B：討論內容、收集資料、撰寫報告、撰寫程式、拍攝影片 同學 C：討論內容、收集資料、撰寫程式、影片剪輯
-----	----	-------	-------------------------------	----------------------------	---