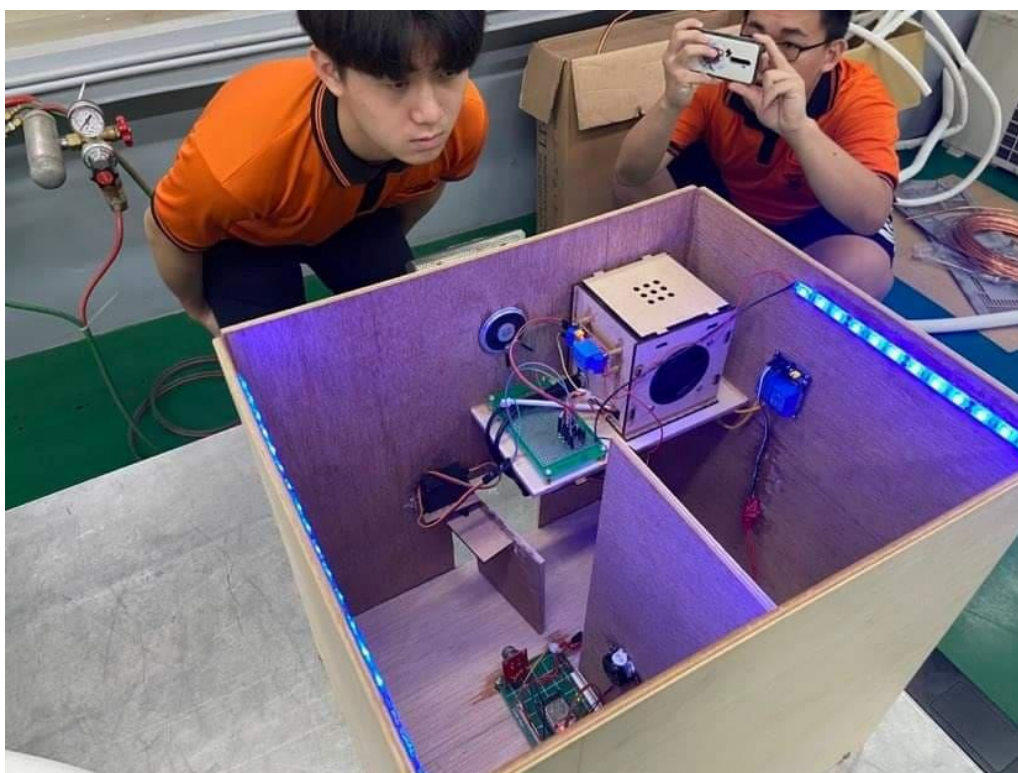


臺北市立大安高級工業職業學校專題製作競賽

作品說明書



類群：電機電子群

作品名稱：智慧居家系統

關鍵詞：智慧居家、物聯網、Arduino

學校名稱/科別：大安高級工業職業學校冷凍科

指導老師：陳睿家老師、歐鎮寬老師、林聖原老師

研究學生姓名：洪啓洋、陳奕廷、林以謙、陳琦斌、

目錄

壹、摘要.....	1
貳、研究動機.....	1
參、主題與課程之相關性或教學單元之說明.....	2
肆、研究方法及過程	4
一、理論討論	4
(一) 物聯網	4
(二) Pixetto 鏡頭.....	4
(三) Arduino	5
(四) Esp32	5
(五) MQTT	6
(六) Node-RED	6
二、設備與器材	7
三、研究流程	8
(一) 研究步驟	8
(二) 操作步驟	9
四、研究過程	10
(一) 程式流程圖	11
(二) 安裝 Pixetto 鏡頭及模板匹配(人臉辨識)	12
(三) 繪畫裝置鏡頭的木板及雷射切割	13
(四) 撰寫伺服馬達、LED 燈及喇叭訊息提醒的程式.....	13
(五) DHT11 連接 MQTT 及控制冷氣的程式.....	14
(六) Node-RED 節點應用	14
(七) MQ2 偵測.....	15
(八) 從環保署網站抓取空氣品質並且連接 Node-RED.....	15
(九) 利用 Node-RED 傳送訊息到 LINE.....	16
(十) Esp32 的接線圖	16
(十一) MQTT Dash 的應用	17

伍、研究結果.....	18
一、利用人臉辨識開關燈及門	18
二、手機控制冷氣啟動	19
陸、討論.....	20
柒、結論.....	20
捌、參考資料及其他	21

表目錄

表 1 專題材料與器材表.....	7
表 2 使用軟體.....	7
表 3 外部功能測試.....	18
表 4 手機控制冷氣配線盤執行表.....	19

圖目錄

圖 1 雷射切割機.....	2
圖 2 Arduino 程式截圖	2
圖 3 冷氣配線盤.....	3
圖 4 物聯網應用.....	4
圖 5 Pixetto 鏡頭	4
圖 6 Arduino Uno 板	5
圖 7 Esp32.....	5
圖 8 MQTT 的運作	6
圖 9 Node-RED 示意圖	6
圖 10 研究步驟.....	8
圖 11 時間分配表.....	8
圖 12 整體流程圖.....	9
圖 13 鏡頭+Arduino 的控制流程圖	10
圖 14 Esp32 控制流程圖	11
圖 15 Pixetto startup 頁面	12
圖 16 模板匹配人臉.....	12
圖 17 放置鏡頭、風扇的木板設計圖.....	13
圖 18 程式截圖.....	13
圖 19 程式截圖.....	14
圖 20 讀取溫濕度及控制冷氣開關的節點.....	14
圖 21 MQ2 程式截圖	15
圖 22 從 json 格式的網站讀取空氣品質+Line 訊息	15
圖 23 Line 訊息節點	16
圖 24 Line 訊息提醒	16
圖 25 Esp32 接線圖	16
圖 26 MQTT Dash 的介面	17

壹、摘要

由於科技的進步，人們對於生活的標準會從原先的實用而轉變成方便，隨著時間的演變，有些人對於居家設備的要求會越來越高，所以會利用遠端遙控來控制家中的電力設備，這樣不但使生活更加便利，也符合未來的變遷。

本專題使用了臉部辨識，並連動 Arduino UNO 板來控制門的開關與電燈的開啟，並因應疫情的影響，我們多加了訊息提醒的功能，提醒家人們進門要噴灑酒精並確實洗手；使用 DHT11 連接 Esp32 來讀取溫溼度並控制冷氣的啟動，使用 Node-RED 連接 MQTT，在手機下載 MQTT Dash，這樣只需要一台手機就可以收到溫溼度數值並開啟冷氣。

因此我們專題做了一間房子來模擬實際情況。

貳、研究動機

現今社會中，由於生活的速度越來越快，因此人們很容易忘東忘西，相信很多學生都有忘記帶鑰匙而不知所措的時候，因此我們利用臉部辨識的方法來解決這個問題，在辨識的瞬間，提醒我們要噴灑酒精並且確實洗手，來達到防疫的效果。相信在夏天的時候，很多人都想回到家就有涼涼的冷氣可以吹，但總只能回到家才能開，因此我們利用 DHT11 讀取家裡溫度，利用 Node-RED 傳送訊息到 LINE，詢問我們是否要開啟冷氣，這樣就可以用手機來控制冷氣，這些功能可以解決生活一部分的不方便。

在這三年內，在智慧居家監控這門課內，學會了 C++ 程式語言，和物聯網的應用，在學習物聯網的過程中，學習 Node-RED 節點應用，並利用 MQTT 連接來達到智慧居家的功能。我們還結合一年級上到的空調繪圖實習，來組裝我們的鏡頭、Arduino UNO 板與風扇，我們將三年的實習所學，來呈現在我們的專題上。

參、主體與課程之相關性或教學單元之說明

一、硬體製作

由於我們的鏡頭容易因為溫度過高而發生 BUG，因此我們設計一個 10*10 的空間來放入風扇及 Arduino Uno 版。由於一年級的冷凍空調繪圖實習，我們接觸到了 AutoCAD，因此我們利用 AutoCAD 來畫我們的圖形，並請製圖科的老師協助我們進行雷射切割的動作，使我們的鏡頭用起來可以更加的順暢。



圖 1 雷射切割機

二、程式設計

軟體的部分，我們選用了高三智慧居家實習課學到的 Arduino 來撰寫及控制，Arduino 具有豐富的數位接腳、可模擬的類比接腳以及可以連接數位元件的序列埠，它使用了簡單且易上手的 C 語言撰寫，我們利用它來控制許多元件，在 Uno 板上，我們控制了伺服馬達、Mp3 播放器、LED 燈；在 Esp32 上，我們控制了 DHT11、MQ2、繼電器，有了程式，可以輕鬆達到自動控制的功能。

```
dht11 | Arduino 1.8.13
EspMQTTClient client(
16   "OPPO Reno2 Z", // Wi-Fi AP name
17   "1111111111", // Wi-Fi Password
18   "broker.emqx.io", // MQTT Broker server ip
19   "", // Broker user name; Can be omitted if not needed
20   "", // Broker password; Can be omitted if not needed
21   "ESP32_Client", // Client name that uniquely identify your device
22   1883 // The MQTT port, default to 1883. this line can be omitted
23);
24
25void setup()
26{
27   Serial.begin(115200);
28   pinMode(13, OUTPUT); // 警告D13 作為繼電器輸出
29   pinMode(21, OUTPUT); // 警告D21 作為輸出 (綠色LED)
30   pinMode(2, OUTPUT); // 警告D2 作為輸出 (紅色LED)
31   pinMode(4, OUTPUT); // 警告D4 作為輸出 (蜂鳴器)
32   pinMode(34, INPUT); // 警告D34 作為輸入 (MQ2氣體感測)
```

Arduino 程式截圖

三、冷氣控制

在高一的電工實習課中，我們學習了許多電器的構造；高二時的冷凍空調實習則是學習冷氣與冰箱，因此我們這組利用實習課所學到的電路知識，把高二所學到的冷氣配線盤拿出來使用，並在開關的地方加裝了一顆繼電器，利用 MQTT 的方式連接到手機，利用手機的 APP 程式來達到控制冷氣開關的效果。

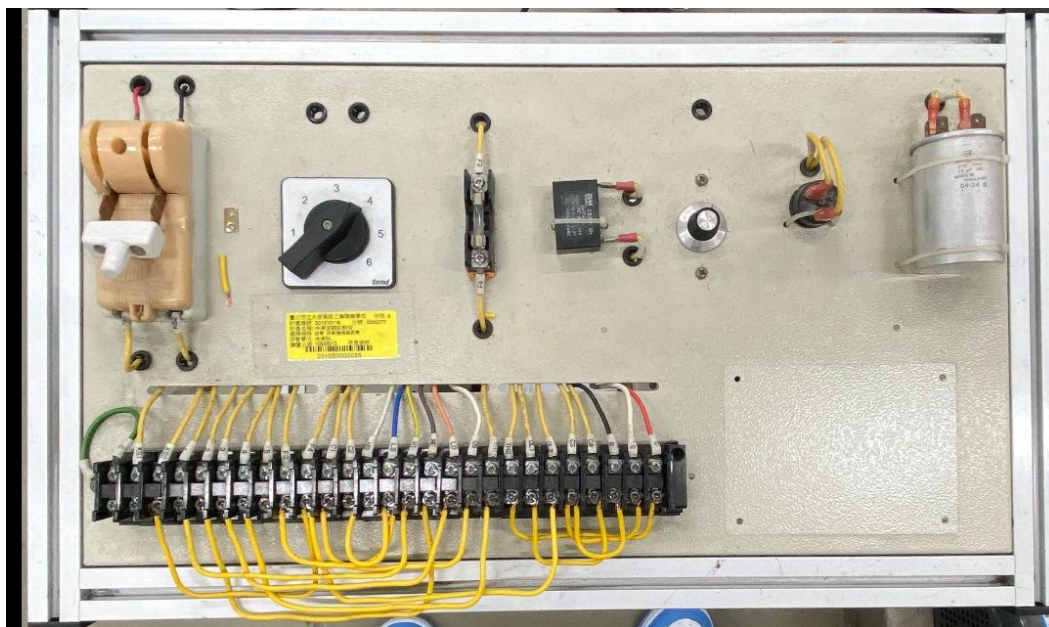


圖 3 冷氣配線盤

四、專題製作

高三的專題製作課程分成兩個組別，一組是專題製作課程，這門課程老師鼓勵大家勇於上台講話，訓練我們的口頭報告和給予我們建議，告訴我們哪裡需要改進，增加我們上台報告的能力，並且用台下同學的提問，來發掘到我們專題需改進的地方；另一組是程式設計實習，我們可以學習到一些設備元件的應用，讓我們不但可以對程式更加熟悉，也可以從中尋找自己專題的方向。

肆、研究方法及過程

一、理論討論

(一) 物聯網

Internet of things 是物聯網的全名，物聯網是將計算裝置、數位機器、機械等科技相互結合的應用，能運用的範圍相當廣闊，涉及大眾的生活，許多家電可透過手機以及語音方式進行操控，許多的應用讓使用者改善生活品質。因此我們的專題以物聯網為基礎，向外延伸出一些功能在我們的住家上。參考資料[1]

IoT裝置進入5G世代的主流應用



圖 4 物聯網應用 示意圖[2]

(二) Pixetto 鏡頭

Pixetto 為一款高畫質視覺感測器,涵蓋物體、形狀、顏色、人臉及手寫辨識等功能，並搭配機器學習，提供直觀且互動性平台以進行人工智慧與機器學習操作，進而探索更多先進技術的可行性應用。因此我們利用它人臉辨識的功能，來運作我們房子的門禁控制跟電燈開啟。參考資料[3]



圖 5 Pixetto 鏡頭 示意圖[4]

(三) Arduino

Arduino 是整合微控制晶片及燒錄功能在一塊開發板上，就可以它的作用，就好比人類的大腦，可以設計判斷不同條件，來去決定身體要做什么，可以利用 Arduino 上的腳位連接原件來達到微控制器的效果。因此我們專題利用 Arduino Uno 板連接鏡頭、伺服馬達、燈與喇叭，來達到我們控制元件的效果。參考資料[5]

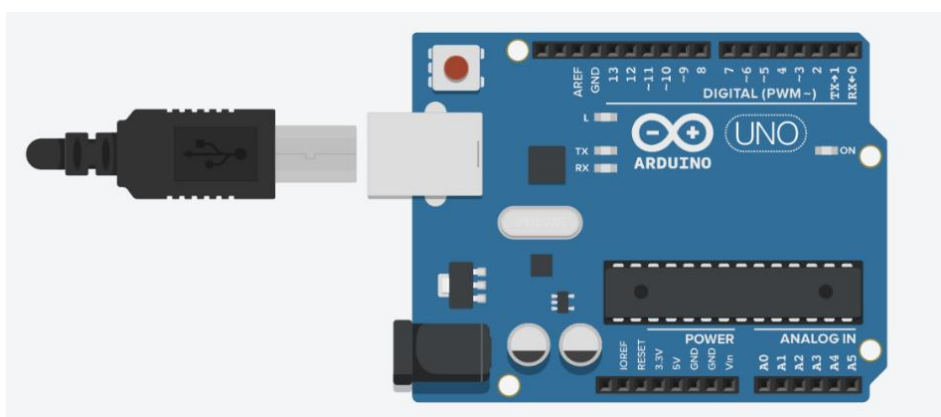


圖 6 Arduino Uno 板

(四) Esp32

Esp32 是一系列低成本，低功耗的單晶片微控制器，整合了 Wi-Fi 和雙模藍牙。跟 Arduino 相比較而言，Esp32 的功能比 Arduino 方便很多，而且腳位大多同時支援類比及數位腳位，並且改善原先 Esp8266 的缺點，現今已成為 IoT 課程最佳的教學工具。因此我們專題把 Esp32 作為連接物聯網的重要工具，連接上 Wi-Fi，並連接一些元件，利用通知來達到智慧控制的功能。參考資料[6]



圖 7 Esp32 示意圖[7]

(五) MQTT

MQTT 協定的訊息內容很精簡，非常適合用於處理器資源及網路頻寬有限的物聯網裝置。MQTT 即是採用 Publish(發佈)/Subscribe(訂閱)模式，其中 Publisher 和 Subscriber 都是用戶端(Client)，Broker 是伺服器端(Server)負責轉發 Topic(標題)。Subscriber 告知 Broker 想要訂閱的 Topic，每當 Publisher 發布訊息時，Broker 會依照 Topic，傳送給訂閱的 Subscriber。因此我們利用 MQTT 傳送溫濕度的數值到我們的用戶端，並利用 Node-RED 裡面的功能來做更多的功能。參考資料[8]

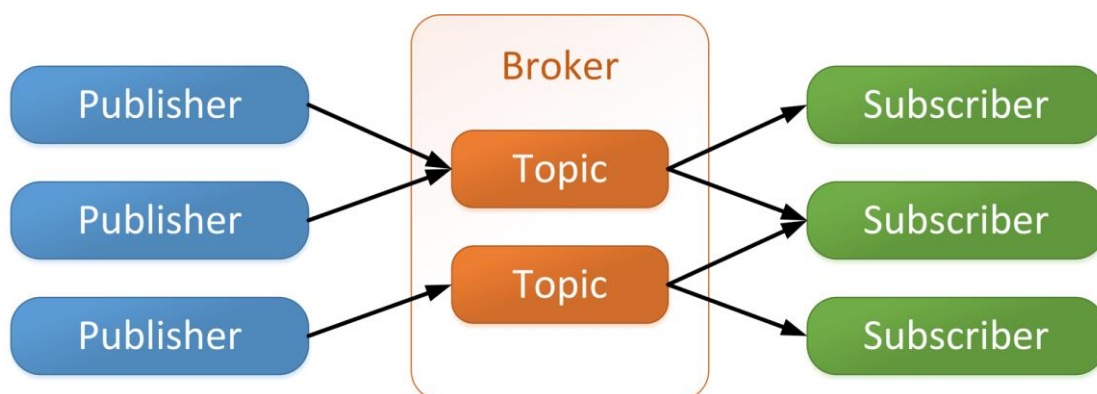


圖 8 MQTT 的運作 示意圖[9]

(六) Node-RED

Node-Red 是一個使用瀏覽器介面的強大物聯網(IoT)開發工具，其視覺開發環境以流程(flows)為基礎，使應用程式撰寫更加簡易。其程式語言涵蓋範圍相當廣。我們可以利用拉節點的方式，來組合我們所想要的流程。因此我們訂閱 MQTT 發送的溫溼度訊息，並利用 Node-RED 連接手機的方式來控制我們的冷氣系統。參考資料[10]

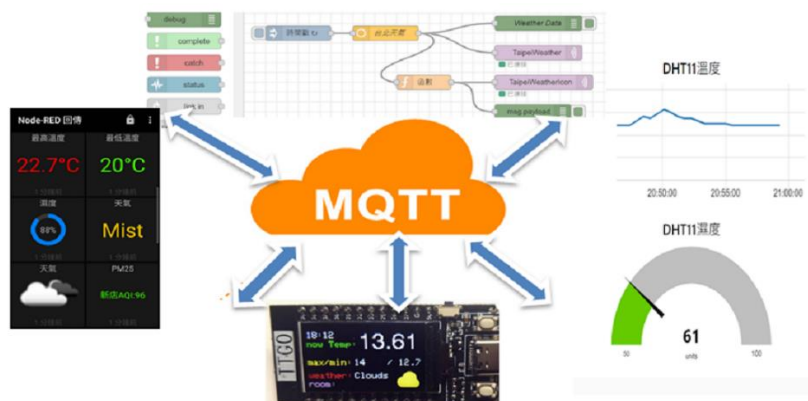


圖 9 Node-RED 示意圖[11]

二、設備及器材

表 1 專題材料與器材表

材料名稱	規格	數量	功能
Pixetto 鏡頭	-	1	人臉辨識
Esp32	WROOM-32	1	連接物聯網
Arduino UNO	ATmega328P	1	連接鏡頭及元件
智慧手機	Android	1	接收 line 訊息/控制冷氣系統
伺服馬達	MG995	1	轉動門的元件
煙霧感測器	MQ2	1	偵測室內煙霧
溫度感測器	DHT11	1	偵測溫度
LED 燈條	5050	1	作為日光燈
喇叭	-	1	防疫訊息提醒
LED 燈	紅、綠	1	指示燈
繼電器	SRD-05VDC-SL-C	1	控制冷氣開關
蜂鳴器	BZ-DC3V	1	提醒濃度過高
散熱風扇	FD1260-S3112C	1	散熱鏡頭
冷氣配線盤	-	1	模擬冷氣系統啟動
雷射切割機	-	1	製作鏡頭空間

表 2 使用軟體

軟體名稱	說明
Pixetto Startup	模板匹配(人臉辨識)
Arduino	撰寫 Esp32 和 Arduino 元件程式
Node-RED	訂閱 MQTT 訊息/傳送訊息至 Line Notify
MQTT Dash	手機溫溼度數值/控制冷氣開關

三、研究流程

(一) 研究步驟

在九月多的時候我們定下了專題題目，我們先從 AI 鏡頭開始著手，藉由模板匹配來達到人臉辨識的效果，為了降低鏡頭的溫度，因此我們在鏡頭後方裝置一顆風扇，我們利用 Auto-CAD 將外觀畫出，後來利用雷射切割把外觀做出來並組裝；內部構造我們使用了 Esp32、DHT11 與 MQ2，我們將溫溼度的數值利用 MQTT 傳送到 Node-red，並設置一些節點，來讓我們可以與手機連接並且控制冷氣。因此我們將冷氣配線盤(模擬冷氣系統)上的線路配好，在開關的地方加裝繼電器來達到手機控制的效果。最終將硬軟體結合，以及一些外觀的改進，經歷了三個月的製作，我們在十二月初時將成品做出來。專題的研究步驟及時間分配分別如下圖 7 及圖 8。

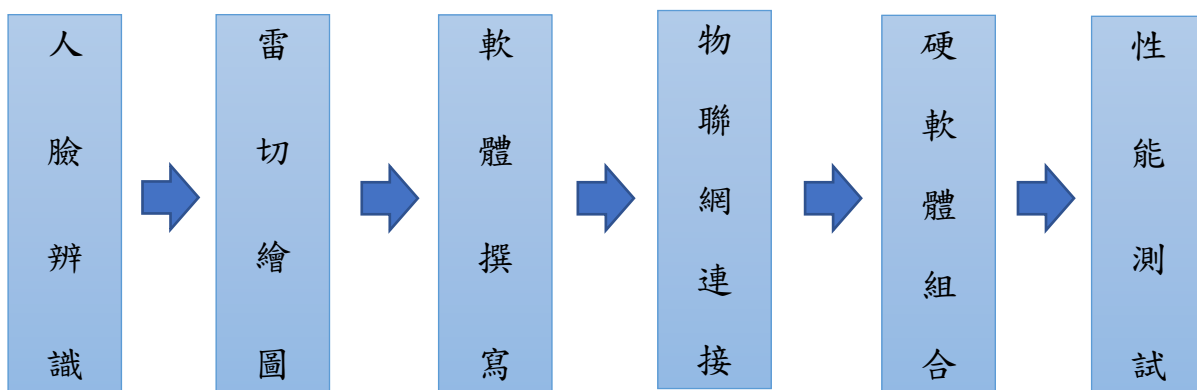


圖 10 研究步驟

	9 月	10 月	11 月	12 月
購買材料				
蒐集資料				
程式設計				
電路設計				
外殼設計				
鏡頭空間設計				
成品測試				

圖 11 時間分配表

(二) 操作流程

室外是利用鏡頭的人臉辨識來開門，同時電燈也會跟著一起亮，並且因應現在疫情的狀況，會利用喇叭提醒我們：「請確實洗手並噴灑酒精」。室內則是利用 DHT11 讀取溫溼度訊息，並會接收溫溼度數值到手機，如果溫度大於 28 度，傳送一則訊息到 LINE：「是否開啟冷氣？」；若低於 22 度，就會發送另一則訊息：「是否關閉冷氣？」，我們可以藉由手機來決定是否開啟冷氣。MQ2 是用來偵測濃度：如果濃度正常則綠燈，若偏高則亮紅燈，過高峰鳴器就會響，來達到警示的效果。

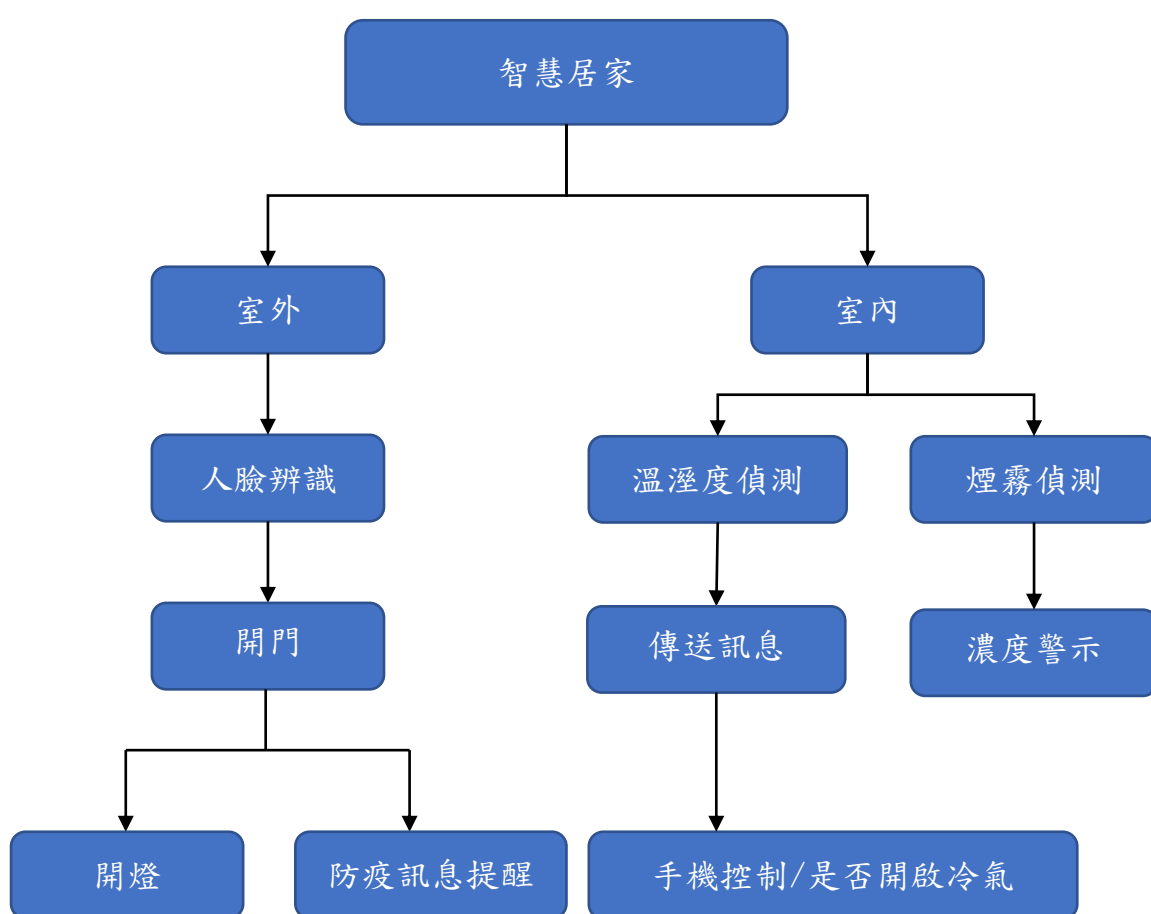


圖 12 整體流程圖

S

四、研究過程
(一) 程式流程圖

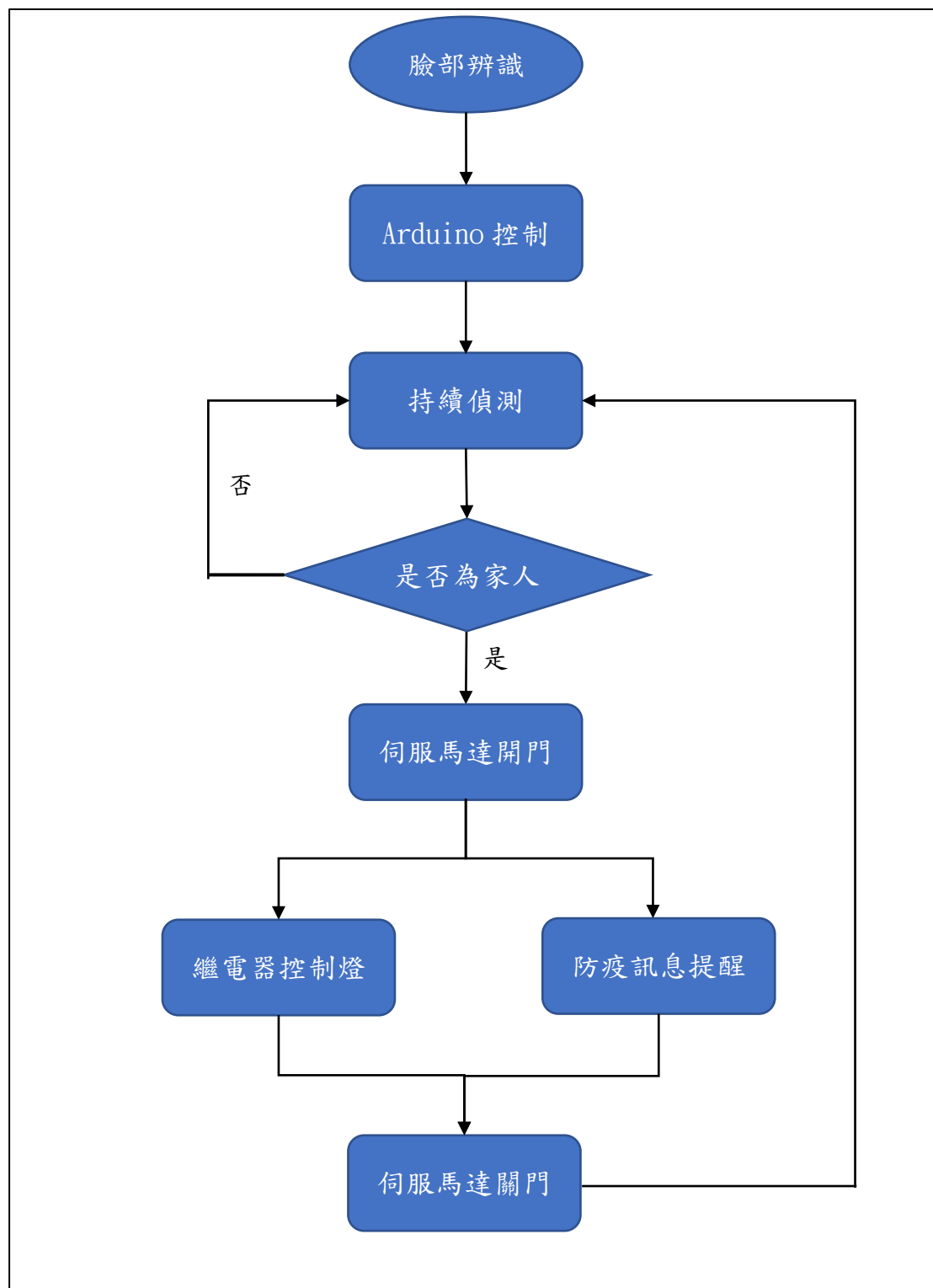


圖 13 鏡頭+Arduino 的控制流程圖

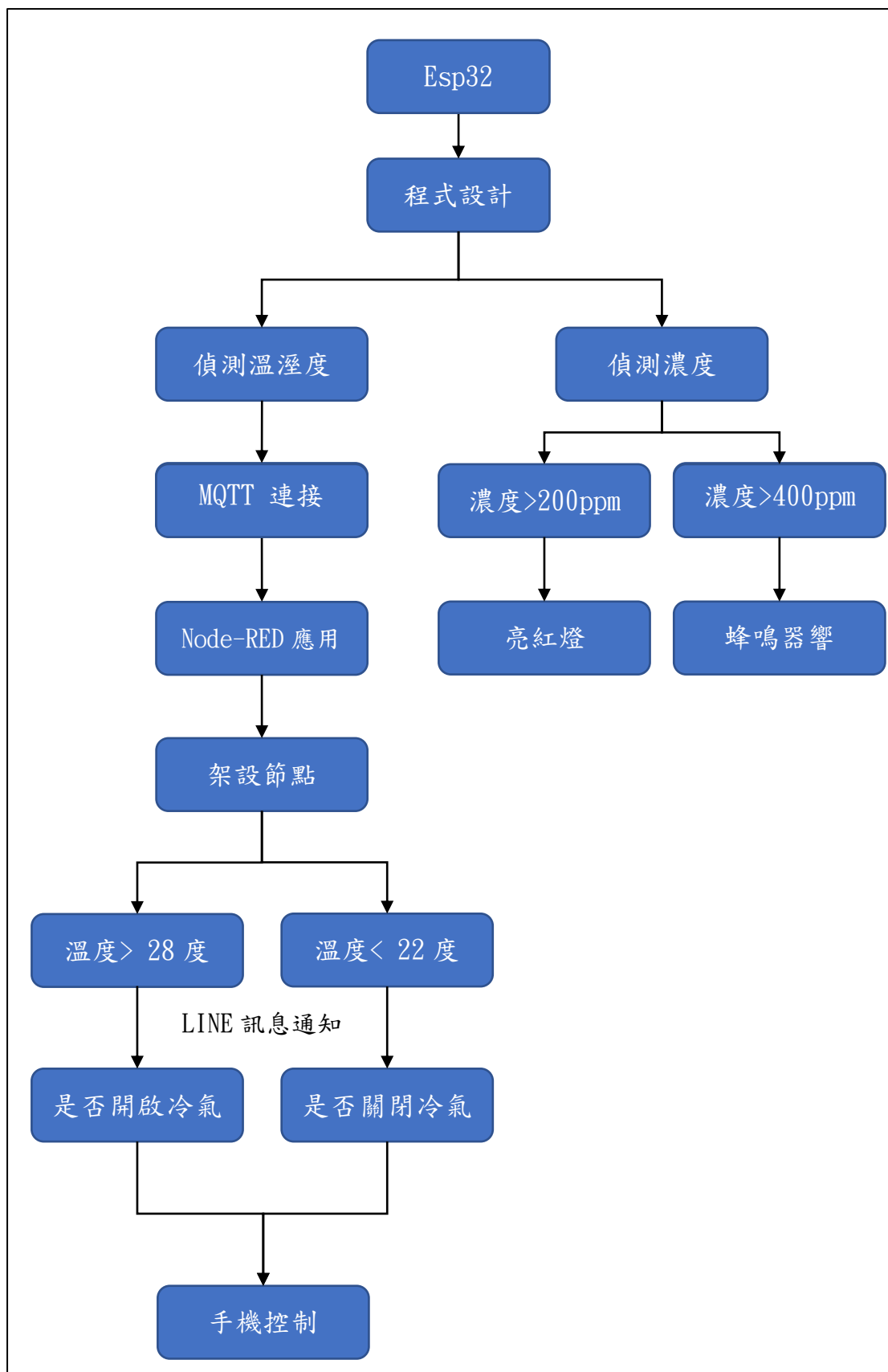


圖 14 Esp32 控制流程圖

(二) 安裝 Pixetto 鏡頭及模板匹配(人臉辨識)

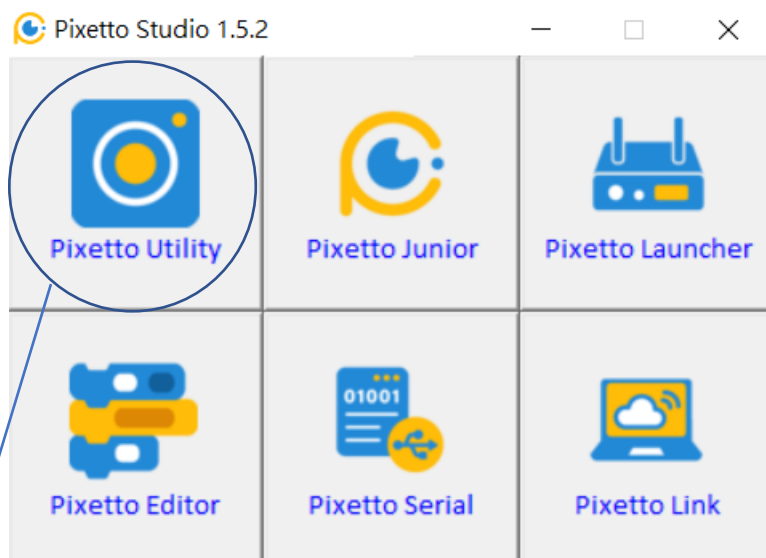


圖 15 Pixetto Startup 頁面

利用此軟體進行模板匹配

下面這張圖有顯示一個 1，代表他是編號 1，有匹配成功，因此有把他的臉部樣子記錄在鏡頭裡



圖 16 模板匹配人臉

(三) 繪畫裝置鏡頭的木板及雷射切割

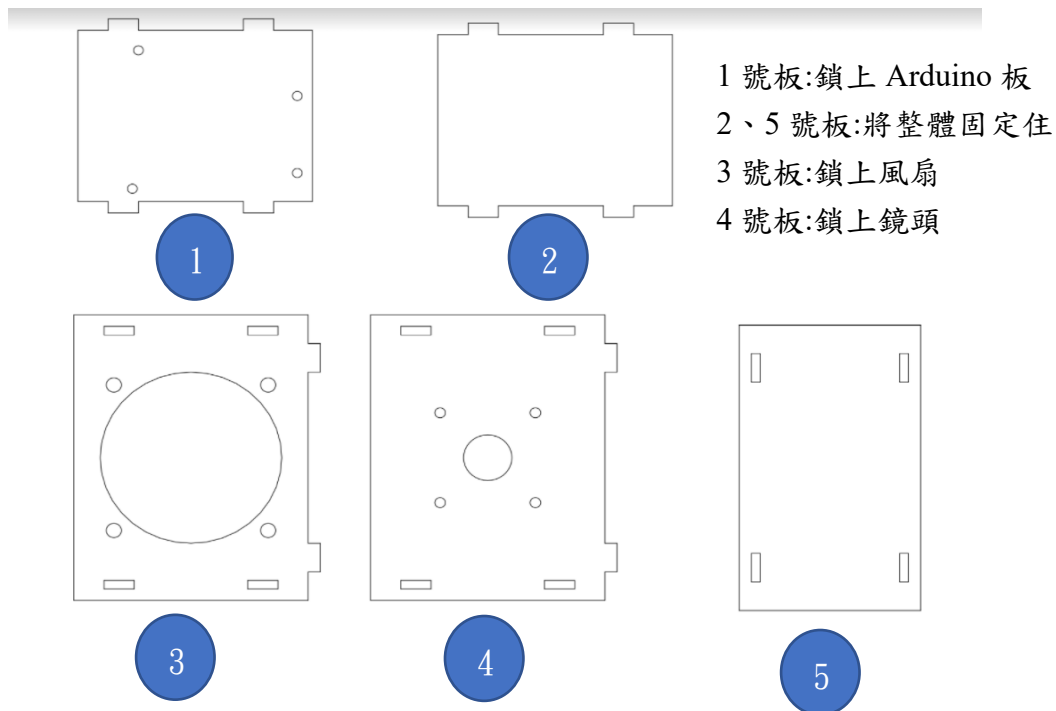


圖 17 放置鏡頭、風扇的木板設計圖

(四) 撰寫伺服馬達、LED 燈及喇叭訊息提醒的程式

```
#include <Pixetto.h> //匯入 Pixetto 程式庫
#include <Servo.h> //匯入伺服馬達程式庫

Pixetto ss(7,8); //定義 Pixetto 鏡頭串列埠腳位
Servo servo_5; //建立 Servo_5 變數

void setup()
{
    ss.begin(); //鏡頭初始化
    servo_5.attach(5); //設定要將伺服馬達接到 5 腳
    pinMode(3,OUTPUT); //第 3 腳位為輸出
    pinMode(13,OUTPUT); //第 13 腳位為輸出
    servo_5.write(90); //設定伺服馬達初始角度為 90 度
}

void loop()
{
    if (ss.isDetected()) //當偵測到物體 {
        if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING && ss.getTypeID() == 1) //當模板匹配數據為 1 {
            servo_5.write(90); //伺服馬達旋轉為 90 度
            digitalWrite(13,HIGH); //13 腳為高電位
            digitalWrite(3,LOW); //3 腳為低電位
            delay(1 * 5000); //延遲 5 秒
            servo_5.write(0); //伺服馬達旋轉為 90 度
            digitalWrite(13,LOW); //13 腳為低電位
            digitalWrite(3,HIGH); //3 腳為高電位
            delay(1 * 5000); //延遲 5 秒
        } else if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING && ss.getTypeID() == 2) //當模板匹配數據為 2 {
            servo_5.write(90);
            digitalWrite(13,HIGH);
            digitalWrite(3,LOW);
            delay(1 * 5000);
            servo_5.write(0);
            digitalWrite(13,LOW);
            digitalWrite(3,HIGH);
            delay(1 * 5000);
        } else if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING && ss.getTypeID() == 3) //當模板匹配數據為 3 {
            servo_5.write(90);
            digitalWrite(13,HIGH);
            digitalWrite(3,LOW);
            delay(1 * 5000);
            servo_5.write(0);
            digitalWrite(13,LOW);
            digitalWrite(3,HIGH);
            delay(1 * 5000);
        } else {
            servo_5.write(0); //伺服馬達不動作
        }
    }
}
```

圖 18 程式截圖

(五) DHT11 溫溼度感測器連接 MQTT 及控制冷氣的程式

```
const long TEMP_updateInterval = 10000; // 10 分鐘讀取一次數值
unsigned long TEMP_currentMillis = 0;
unsigned long TEMP_previousMillis = 0;

#include "DHT.h"
#define DHTPIN 5 //ESP32 DHT11 接腳 pin 5
#define DHTTYPE DHT11
String message ;

DHT dht(DHTPIN, DHTTYPE); // 初始化 DHT 感測器
#include "EspMQTTClient.h"

EspMQTTClient client// 定義
(
    "OPPO Reno2 Z", // Wi-Fi 名稱
    "1111111111", // Wi-Fi 密碼
    "broker.emqx.io", // MQTT Broker 服務器
    "", // Broker user name; Can be omitted if not needed
    "", // Broker password; Can be omitted if not needed
    "ESP32_Client", // Client name that uniquely identify your device
    1883 // The MQTT port
);

void setup()
{
    Serial.begin(115200);
    pinMode(13, OUTPUT); // 宣告 D13 作為繼電器輸出
    //Start DH11 sensor
    dht.begin();

    client.enableDebuggingMessages();
}

void onConnectionEstablished()
{
    // Subscribe to "mytopic/test" and display received message to Serial
    client.subscribe("DAAN/3A18/sw1", [](const String &payload)
    { message = payload;
      Serial.println(message);
    });
}

//繼電器控制冷氣
if (message == "0")
{
    digitalWrite(13,HIGH);
} else if (message == "1") {
    digitalWrite(13,LOW);
} });
client.publish("DAAN/3A18/temp", "Let's go!");
}

void loop()
{
    //讀取溫溼度
    TEMP_currentMillis = millis();
    if(TEMP_currentMillis - TEMP_previousMillis >= TEMP_updateInterval){
        TEMP_previousMillis = TEMP_currentMillis;

        float humidity = dht.readHumidity(); //濕度的數值
        float temperature = dht.readTemperature(); //溫度的數值

        if (!isnan(humidity) || !isnan(temperature))
        {
            Serial.println(F("Failed to read from DHT sensor!")); //讀取錯誤
            return;
        }
        String tmp = "Current Temperature: ";
        String hud = "Current Humidity: ";
        Serial.print(tmp);
        Serial.println(String(temperature));
        Serial.print(hud);
        Serial.println(String(humidity));
        //發布
        client.publish("DAAN/3A18/temp",String(temperature));"DAAN/3A18/temp"
        client.publish("DAAN/3A18/humd",String(humidity));"DAAN/3A18/humd"
        { Serial.println(payload); });
    }
    client.loop();
}
```

圖 19 程式截圖 參考資料[12]

(六) Node-RED 節點應用

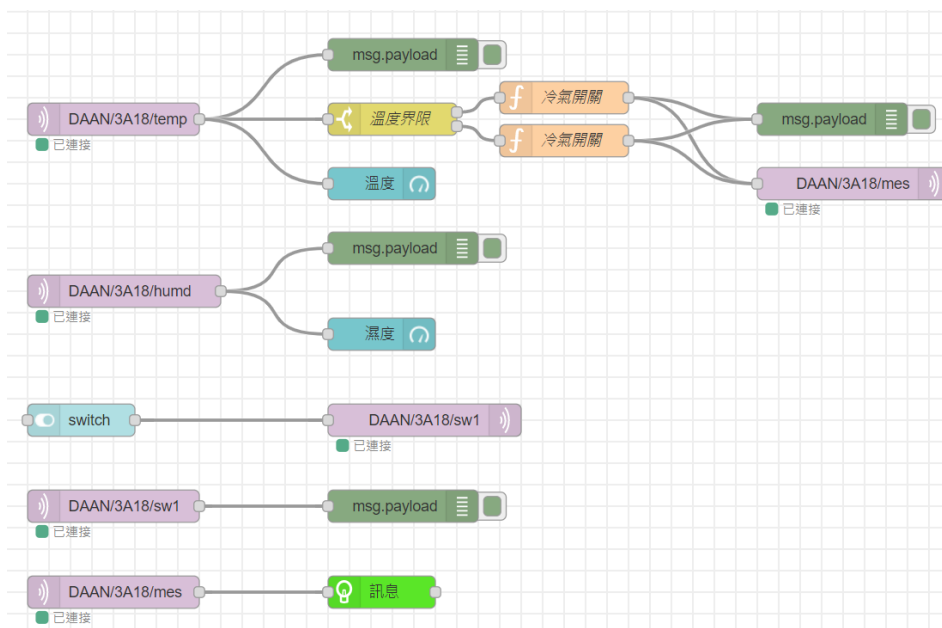


圖 20 讀取溫濕度及控制冷氣開關的節點

(七) MQ2 煙霧感測器

```
void setup()
Serial.begin(115200);

pinMode(21, OUTPUT); //選告 D21 作為輸出 (綠色 LED)
pinMode(2, OUTPUT); //選告 D2 作為輸出 (紅色 LED)
pinMode(4, OUTPUT); //選告 D4 作為輸出 (蜂鳴器)
pinMode(34, INPUT); //選告 D34 作為輸入 (Mq2 氣體感測)

void loop()
{
//MQ2 煙霧感測
int GasValue;//宣告變數 GasValue
GasValue = analogRead(34); //讀取 D34 的數值放在 GasValue
Serial.println(GasValue); //GasValue 顯示在序列視窗
if (GasValue < 200) {
//小於 200，代表一切正常，亮綠燈
digitalWrite(21, HIGH);
digitalWrite(2, LOW);
digitalWrite(4, LOW);
}

if (GasValue >= 200 && GasValue < 400) {
//200-400 之間，紅燈顯示外洩警戒
digitalWrite(21, LOW);
digitalWrite(2, HIGH);
digitalWrite(4, LOW);
}

if (GasValue >= 400) {
//超過 400，蜂鳴器發出外洩警報
digitalWrite(21, LOW);
digitalWrite(2, LOW);
digitalWrite(4, HIGH);
}
delay(100); //延遲 0.1 秒
}
```

圖 21 MQ2 程式截圖 參考資料[13]

(八) 從環保署網站抓取空氣品質並且連接 Node-RED

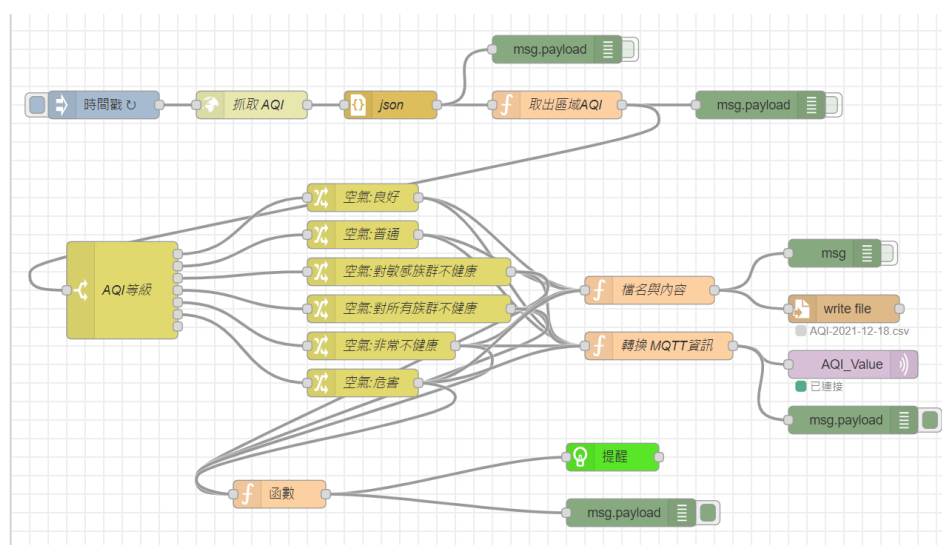


圖 22 從 json 格式的網站讀取空氣品質+Line 訊息

(九) 利用 Node-RED 傳送訊息到 LINE

如果是要直接傳送訊息的話，需要拉一個節點，叫做”function”，而我這張圖的”function”請看圖 20、圖 22。

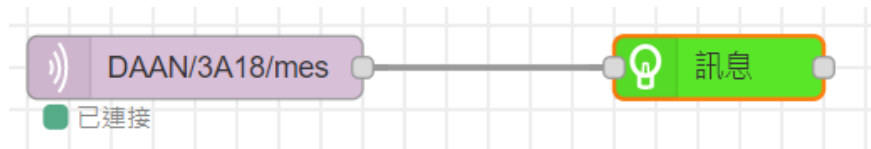


圖 23 Line 訊息節點



圖 24 Line 訊息提醒

(十) Esp32 接線圖

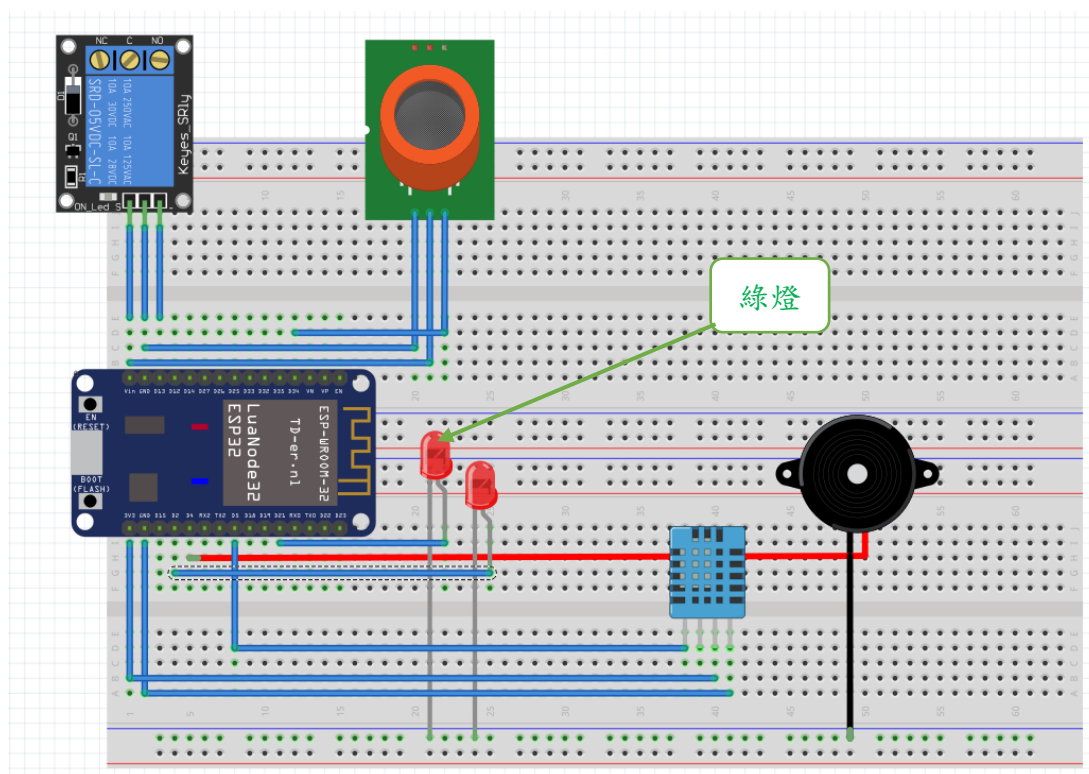


圖 25

(十一) MQTT Dash 的應用

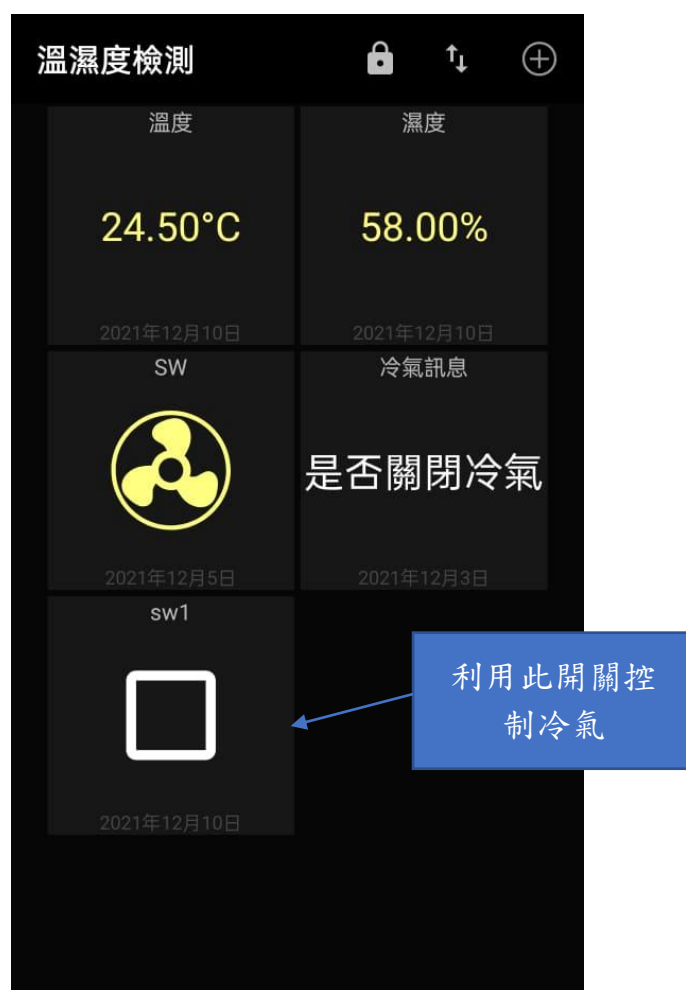
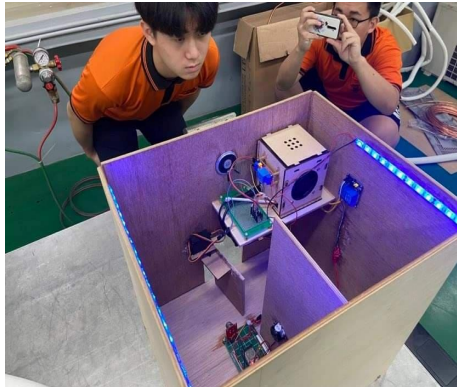
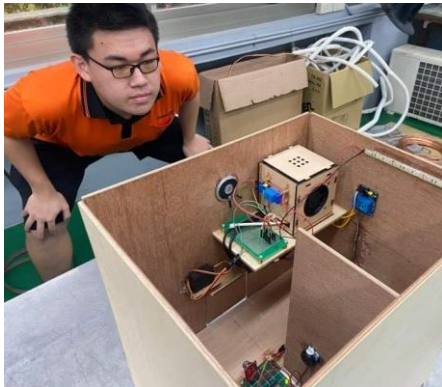


圖 26 MQTT Dash 的介面

伍、研究結果

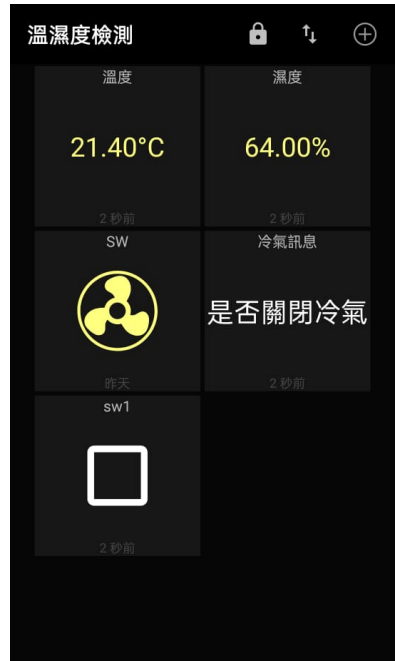
一、利用人臉辨識開關燈及門

表 3 外部功能測試

1. 人臉辨識	2. 開門與開燈
	
3. 過了 5 秒後關門與關玄關的燈	4. 辨識到陌生人門不開啟
	

二、手機控制冷氣啟動

表 4 手機控制冷氣配線盤執行表

	開 關 打 開		壓縮機啟動(啟動電流約1.2安培)
	開 關 關 閉		壓縮機停止(無啟動電流)

陸、討論

1. 在操作繼電器的部分碰到困難

在我們的程式中，原先的程式是沒有繼電器的，因此在當我加入繼電器的程式時，電腦會一直顯示無法編譯，但是原先的程式是沒問題的，我們一直想不通，因此我從我連接 MQTT 的部分開始檢查，最後發現到我有一個變數沒有定義到，變數定義完後程式就可以順利運行了。

2. 伺服馬達運作時會受到門框的限制

在程式控制伺服馬達的過程中，若沒有歸零伺服馬達的角度，就會因為角度轉動過大導致門的甩幅過大不好控制，害我們很頭痛，幸好在老師的推薦下，介紹了 server test(寬頻窄頻機測試)可以幫助我們歸零伺服馬達，並剪去了一部分固定伺服馬達的樞紐，解決了甩幅過大的問題。

3. 鏡頭不穩定性太高，導致一開始程式無法執行

這個鏡頭是老師介紹給我們的，所以就用這個鏡頭加入我們的專題，但是在執行程式時，鏡頭總無法運行，而且網路範例也過少，導致我們只能不斷嘗試用各種方法編譯，到最後發現這個鏡頭有固定的程式碼，再放入要執行的程式，最終總算成功了。

4. 雷射切割後有誤差導致無法正常組裝

因不知道準確的鏡頭尺寸導致再組裝鏡頭上不能馬上完成：因為廠商沒有提供鏡頭完整的尺寸，所以要自行量測和測試所切木板是否有足夠的空間和能否組裝在一起。而且切割一次就要六個面都切割，所以只要有一點誤差無法組裝就必須去不重切，但最後經過多次的嘗試，終於可以完美的將所有木板組裝上去。

柒、結論

我們的智慧居家系統擁有許多的功能，利用 Arduino 和人臉辨識，達到控制門的開啟，並開啟燈與防疫訊息的提醒；並使用 Esp32 和物聯網，將溫溼度數值利用 Node-RED，傳送到手機並用手機控制冷氣模擬配線盤的開啟。這些功能可以解決我們許多生活的不方便，也可因應疫情狀況下的各種防疫需求。

透過這次的專題實作，我們能夠了解未來居家監控的趨勢，看到其發展的可能。我們將高職三年所學到的東西合併在我們的專題上，雖然整個過程都是跌跌撞撞的，光是一些小地方就花去我們許多的時間，但我們能藉由查詢資料來解決這些問題。

在實驗的過程中，由於對程式的不熟悉，因此我們花很大的心力在理解程式邏輯的部分，花很多時間在查詢相關的資料，也花很多的時間在跟老師討論，過程中很疲乏也很辛苦，但看到最後的成果是順利的，我們非常的開心，因為付出是有收穫的。

我們的專題不只結合了我們冷凍空調科所學，更多的是需要我們自己統合上網查詢的資料，雖然很多東西是我們第一次接觸到，但網路可以協助解決我們大部分的問題。一個專題最重要的不是我們之前學到了什麼，而是我們在過程中收穫了什麼，對於我們這組學到最多的地方是「做事的態度」，我們互相監督，互相進步，互相幫助，遇到困難一起處理，這使我們的專題更加的完美。

捌、附錄

1. 參考資料

[1]物聯網資料來源

<https://www.machsync.com.tw/information/IoT>

[2]物聯網圖片

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.digitimes.com.tw%2Fiot%2Fpackage_show.asp%3Fcat%3D158%26id%3D546356%26packageid%3D13117&psig=AOvVaw1_suqGgJHgxeiUg3TfcLn_&ust=1640077830606000&source=images&cd=vfe&ved=0CAgQjRxqFwoTCJi-8pmE8vQCFQAAAAAdAAAAABAD

[2]鏡頭資料來源

<https://pixetto.ai/tw/about-tw/>

[4]鏡頭圖片

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.motoduino.com%2Fproduct%2Fvia_ai_camera%2F&psig=AOvVaw1Y0qfw085ujrW43wJnj0rR&ust=1640078000780000&source=images&cd=vfe&ved=0CAgQjRxqFwoTCLDj9uaE8vQCFQAAAAAdAAAAABAN

[5]Arduino 資料來源

<https://crazymaker.com.tw/what-is-arduino/>

[6]esp32 資料來源

<https://makerpro.cc/2020/06/esp32-review-and-why-recommend-nodemcu-32s/>

[7]esp32 圖片

<https://www.051376.com/wp-content/uploads/2021/02/Getting-Started-with-ESP32-1.jpg>

[8]mqtt 資料來源

<https://medium.com/%E5%BD%BC%E5%BE%97%E6%BD%98%E7%9A%84-swift-ios-app-%E9%96%8B%E7%99%BC%E6%95%99%E5%AE%A4/ios-x-iot-2-mqtt-%E7%B0%A1%E4%BB%8B-e750aa420162>

[9]mqtt 圖片

<https://blog.gtwang.org/wp-content/uploads/2016/12/mqtt-publisher-subscriber-broker-20161230-1.png>

[10]node-red 資料來源

<https://www.inside.com.tw/article/6638-developing-with-node-red>

[11]node-red 圖片

<https://static.accupass.com/eventbanner/2107191241426384316600.jpg>

[12]dht11 程式參考

<https://stonez56.blogspot.com/2021/10...>

[13]Mq2 程式參考

<https://youyouyou.pixnet.net/blog/post/120272743-%E7%AC%AC%E4%B8%83%E7%AF%87-esp32%E7%93%A6%E6%96%AF%E8%AD%A6%E5%A0%B1%E5%99%A8>

2. 程式

```
#include <Pixetto.h>
#include <Servo.h>
Pixetto ss(7,8);
Servo servo_5;

void setup()
{
    ss.begin();
    servo_5.attach(5);
    pinMode(3,OUTPUT);
    pinMode(13,OUTPUT);
    servo_5.write(90);
}
void loop()
{
```

```

    if (ss.isDetected()) {
        if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING &&
ss.getTypeID() == 1) {
            servo_5.write(90);
            digitalWrite(13,HIGH);
            digitalWrite(3,LOW);
            delay(1 * 5000);
            servo_5.write(0);
            digitalWrite(13,LOW);
            digitalWrite(3,HIGH);
            delay(1 * 5000);
        } else if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING &&
ss.getTypeID() == 2) {
            servo_5.write(90);
            digitalWrite(13,HIGH);
            digitalWrite(3,LOW);
            delay(1 * 5000);
            servo_5.write(0);
            digitalWrite(13,LOW);
            digitalWrite(3,HIGH);
            delay(1 * 5000);
        } else if (ss.getFuncID() == Pixetto::FUNC_TEMPLATE_MATCHING &&
ss.getTypeID() == 3) {
            servo_5.write(90);
            digitalWrite(13,HIGH);
            digitalWrite(3,LOW);
            delay(1 * 5000);
            servo_5.write(0);
            digitalWrite(13,LOW);
            digitalWrite(3,HIGH);
            delay(1 * 5000);
        } else {
            servo_5.write(0);
        }
    }
}

```

```

const long TEMP_updateInterval = 10000; // 10 分鐘讀取一次數值
unsigned long TEMP_currentMillis = 0;
unsigned long TEMP_previousMillis = 0;

#include "DHT.h"
#define DHTPIN 5 //ESP32 DHT11 接腳 pin 5
#define DHTTYPE DHT11
String message ;

DHT dht(DHTPIN, DHTTYPE); // 初始化 DHT 感測器
#include "EspMQTTClient.h"

EspMQTTClient client// 定義
(
    "OPPO Reno2 Z",      // Wi-Fi 名稱
    "1111111111",        // Wi-Fi 密碼
    "broker.emqx.io",     // MQTT Broker 服務器
    "",                  // Broker user name; Can be omitted if not needed
    "",                  // Broker password; Can be omitted if not needed
    "ESP32_Client",      // Client name that uniquely identify your device
    1883                 // The MQTT port
);
void setup()
{
    Serial.begin(115200);
    pinMode(13, OUTPUT); //選告 D13 作為繼電器輸出
    //Start DH11 sensor
    dht.begin();

    client.enableDebuggingMessages();
}
void onConnectionEstablished()
{
    // Subscribe to "mytopic/test" and display received message to Serial
    client.subscribe("DAAN/3A18/sw1", [] (const String &payload)
    { message = payload;
    Serial.println(message);
    //繼電器控制冷氣

```



```

        if (message == "0")
        {
            digitalWrite(13,HIGH);
        } else if (message == "1") {
            digitalWrite(13,LOW);
        } });
        client.publish("DAAN/3A18/temp", "Let's go!");
    }
    void loop()
    {
        //讀取溫溼度
        TEMP_currentMillis = millis();
        if(TEMP_currentMillis - TEMP_previousMillis >= TEMP_updateInterval){
            TEMP_previousMillis = TEMP_currentMillis;

            float humidity = dht.readHumidity();          //濕度的數值
            float temperature = dht.readTemperature();    //溫度的數值

            if (isnan(humidity) || isnan(temperature))
            {
                Serial.println(F("Failed to read from DHT sensor!")); //讀取錯誤
                return;
            }
            String tmp = "Current Temperature: ";
            String hud = "Current Humidity: ";
            Serial.print(tmp);
            Serial.println(String(temperature));
            Serial.print(hud);
            Serial.println(String(humidity));
            //發佈
            client.publish("DAAN/3A18/temp",String(temperature));"DAAN/3A18/temp"
            client.publish("DAAN/3A18/humd",String(humidity));"DAAN/3A18/humd"
            { Serial.println(payload); });
        }
        client.loop();
    }
}

```