

# 嵌入式系統設計 期末報告

## 智慧家電

指導教授：陳慶瀚

學生：111522034 陳彥廷

111522033 徐裕翔

111522151 吳苡菡

# 目錄

一、摘要.....	3
二、前言.....	3
三、硬體架構.....	4
四、系統設計.....	5
1. IDEF0.....	5
2. Grafcet.....	6
3. C code 合成.....	8
4. 軟體合成.....	12
5. 硬體驗證.....	17
五、未來展望.....	18
六、團隊分工.....	18
七、參考資料.....	18

## 一、 摘要

本專題利用 esp32、溫溼度感測器 DHT22、伺服馬達 SG90（模擬電風扇的狀態），以及 Blynk (IoT platform)，實作智慧家電。另外目前我們使用手機作為拍攝照片的工具，但其實之後是可以考慮使用 esp32-CAM 開發版，因為 esp32-CAM 具備拍照、串流視訊以及人臉辨識功能。

esp32 作為控制中樞，蒐集由 DHT22 所感測出的溫濕度資料和影像資料（分析人數），esp32 連結電腦後開始進行資料分析，分析完成後輸出電風扇的模式，並以 SG90 的 position 變化速度，來模擬電風扇的不同模式，最後再透過 Blynk 將資料（溫度、濕度、人數、電風扇模式）顯示在 Blynk app 上。

## 二、 前言

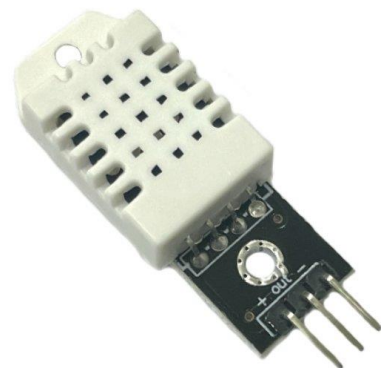
近年來科產業發展迅速，尤其攜帶式電子裝置，例如智慧型手機、平板電腦等等，更是成為人們生活中不可或缺的一部分，我們希望透過 esp32 以及 Blynk 平台，去模擬一個 IoT 的智慧家電系統。

目前我們僅僅是在進行溫溼度以及人數進行分析後，利用 SG90 去模擬電風扇的不同狀態，但在此系統的基礎之上，我們可以加入其他不同的感測器，並用 Blynk 做整合，以達成更多豐富的功能。

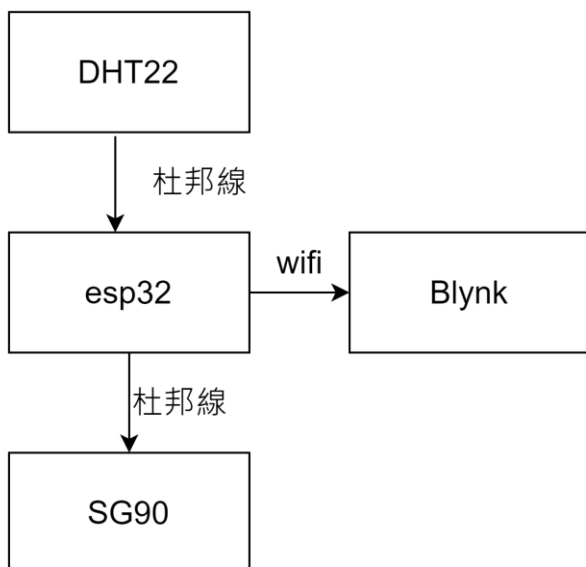
### 三、 硬體架構



SG90 9g 伺服馬達  
棕色： GND  
紅色： VCC 3~7.2V (建議 5V)  
橙色： 控制訊號

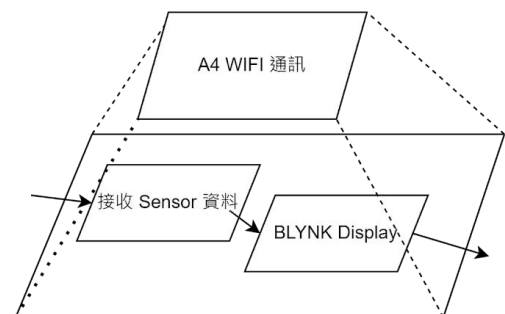
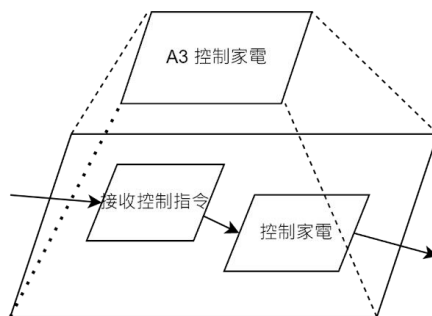
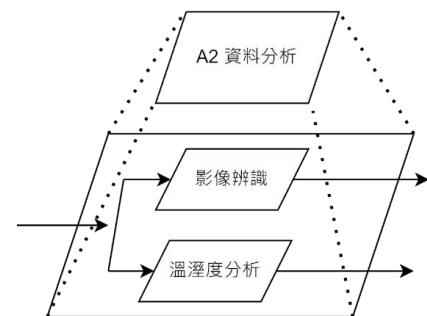
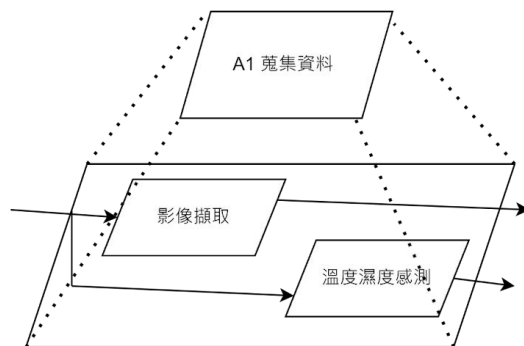
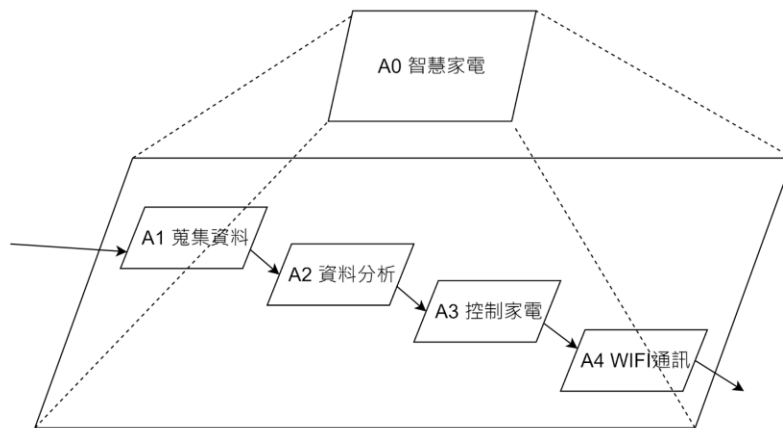


DHT22 溫溼度感測器  
溫度測量範圍： $-40^{\circ}\text{C}$ ~ $80^{\circ}\text{C}$   
測量精度： $0.5^{\circ}\text{C}$   
濕度測量範圍： $0\sim 100\%\text{RH}$   
+：VCC 5V  
Out：GND  
-：控制訊號



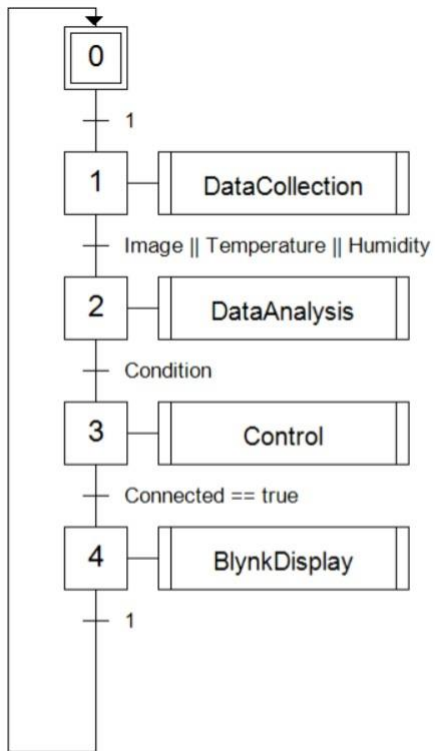
## 四、系統設計

### 1. IDEF0

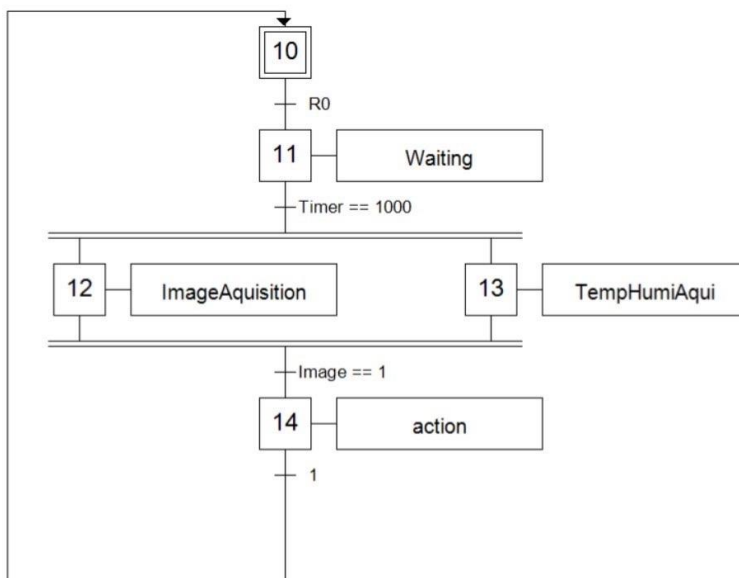


## 2. Grafcet

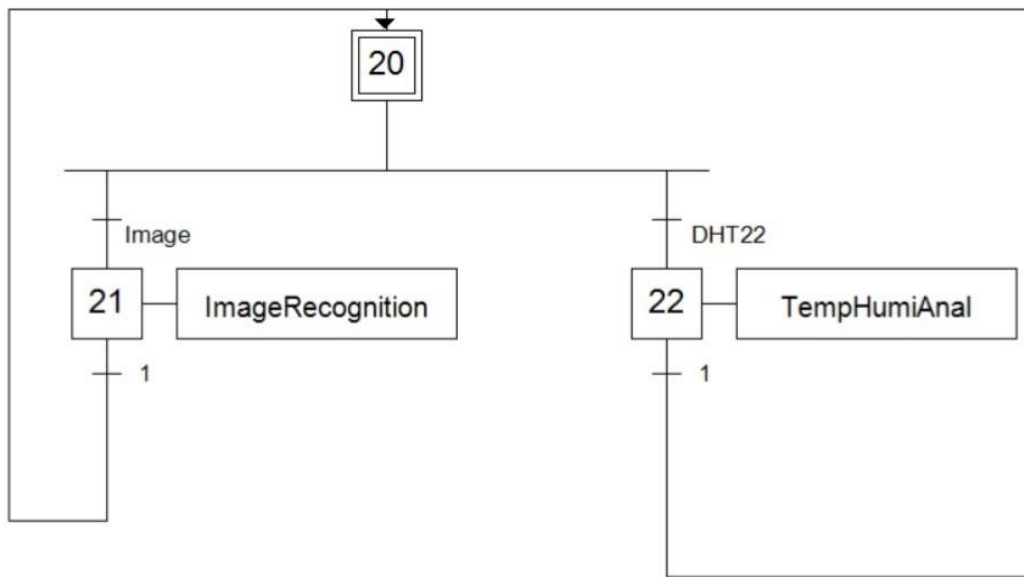
- G0



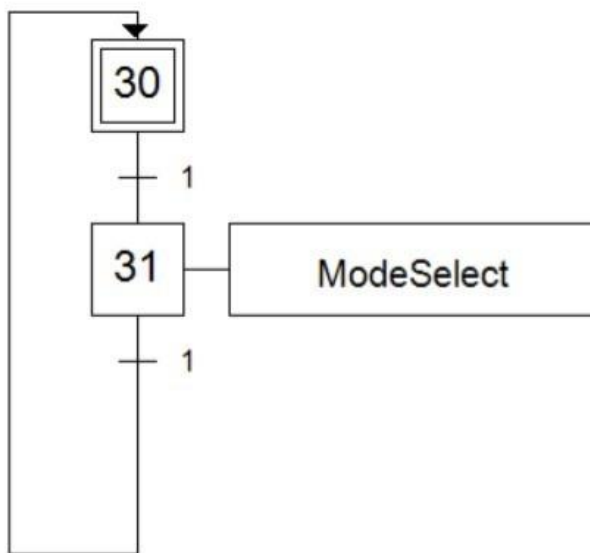
- G1



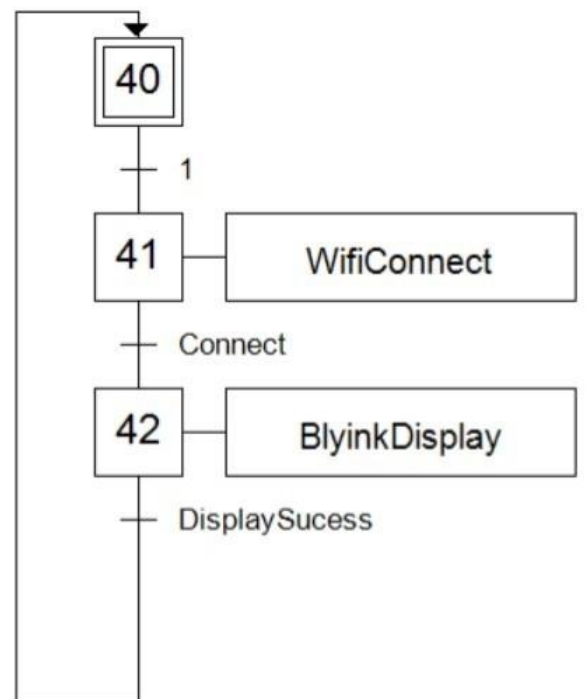
- G2



- G3



- G4



### 3. C code 合成

#### A. 模擬狀態轉移

```
X0 = 1,X1 = 0,X2 = 0,X3 = 0,X4 = 0  
action start !  
X0 = 0,X1 = 1,X2 = 0,X3 = 0,X4 = 0  
DataCollection activate !!  
X0 = 0,X1 = 0,X2 = 1,X3 = 0,X4 = 0  
DataAnalysis activate !!  
X0 = 0,X1 = 0,X2 = 0,X3 = 1,X4 = 0  
Control activate !!  
X0 = 0,X1 = 0,X2 = 0,X3 = 0,X4 = 1  
BlynkDisplay activate !!  
X0 = 1,X1 = 0,X2 = 0,X3 = 0,X4 = 0
```



## B. C code

```
#include<stdio.h>

int X0 = 1, X1 = 0, X10 = 1, X11 = 0, X12 = 0, X13 = 0, X14 = 0, X2 = 0, X20 = 1,
    X21 = 0, X22 = 0, X3 = 0, X30 = 1, X31 = 0, X4 = 0, X40 = 1, X41 = 0, X42 = 0;

void grafcet0();
void datapath0();
void action();
void DataCollection();

void grafcet1();
void datapath1();
void DataAnalysis();

void grafcet2();
void datapath2();
void Control();

void grafcet3();
void datapath3();
void BlynkDisplay();

void grafcet4();
void datapath4();
void Waiting();
void ImageAquisition();
void TempHumiAqui();
void ImageRecognition();
void TempHumiAnal();
void ModeSelect();
void WifiConnect();
void BlynkDisplay();

void main(){
    printf("X0 = %d,X1 = %d,X2 = %d,X3 = %d,X4 = %d\n",X0 ,X1 ,X2 ,X3 ,X4 );
    while(1){
        datapath0();
        grafcet0();
        printf("X0 = %d,X1 = %d,X2 = %d,X3 = %d,X4 = %d\n",X0 ,X1 ,X2 ,X3 ,X4 );
    }
}
```

```

void grafcet0(){

    if((X0 == 1) && (1)){
        X0 = 0;
        X1 = 1;
        return;
    }

    if((X1 == 1) && (Image || Temperature || Humidity)){
        X1 = 0;
        X2 = 1;
        return;
    }

    if((X2 == 1) && (Condition)){
        X2 = 0;
        X3 = 1;
        return;
    }

    if((X3 == 1) && (Connected == true)){
        X3 = 0;
        X4 = 1;
        return;
    }

    if((X4 == 1) && (1)){
        X4 = 0;
        X0 = 1;
        return;
    }

}

```

```

void grafcet1(){

    if((X10 == 1) && (R0)){
        X10 = 0;
        X11 = 1;
        return;
    }

    if((X11 == 1) && (Timer == 1000)){
        X11 = 0;
        X12 = 1;
        X13 = 1;
        return;
    }

    if(X12 == 1 && (Image == 1)){
        X12 = 0;
        X14 = 1;
        return;
    }

    if(X13 == 1 && (Temperature && Humidity)){
        X13 = 0;
        X14 = 1;
        return;
    }

    if((X14 == 1) && (1)){
        X14 = 0;
        X10 = 1;
        return;
    }

}

```

```

void grafcet2(){

    if(X20 == 1){
        if(Image){
            X20 = 0;
            X21 = 1;
        }
        else if( DHT22){
            X20 = 0;
            X22 = 1;
        }
        return;
    }

    if((X21 == 1) && (1)){
        X21 = 0;
        X20 = 1;
        return;
    }

    if((X22 == 1) && (1)){
        X22 = 0;
        X20 = 1;
        return;
    }

}

```

```

void grafcet3(){

    if((X30 == 1) && (1)){
        X30 = 0;
        X31 = 1;
        return;
    }

    if((X31 == 1) && (1)){
        X31 = 0;
        X30 = 1;
        return;
    }

}

```

```

void grafcet4(){

    if((X40 == 1) && (1)){
        X40 = 0;
        X41 = 1;
        return;
    }

    if((X41 == 1) && (Connect)){
        X41 = 0;
        X42 = 1;
        return;
    }

    if((X42 == 1) && (DisplaySucess)){
        X42 = 0;
        X40 = 1;
        return;
    }

}

```

```

void action(){
    printf("action activate !!\n");
}

void DataCollection(){
    printf("DataCollection activate !!\n");
    datapath1();
    grafcet1();
    printf("X10 = %d,X11 = %d,X12 = %d,X13 = %d,X14 = %d\n",X10 ,X11 ,X12 ,X13 ,X14 );
}

void DataAnalysis(){
    printf("DataAnalysis activate !!\n");
    datapath2();
    grafcet2();
    printf("X20 = %d,X21 = %d,X22 = %d\n",X20 ,X21 ,X22 );
}

void Control(){
    printf("Control activate !!\n");
    datapath3();
    grafcet3();
    printf("X30 = %d,X31 = %d\n",X30 ,X31 );
}

void BlynkDisplay(){
    printf("BlynkDisplay activate !!\n");
    datapath4();
    grafcet4();
    printf("X40 = %d,X41 = %d,X42 = %d\n",X40 ,X41 ,X42 );
}

```

```

void Control(){
    printf("Control activate !!\n");
    datapath3();
    grafcet3();
    printf("X30 = %d,X31 = %d\n",X30 ,X31 );
}

void BlynkDisplay(){
    printf("BlynkDisplay activate !!\n");
    datapath4();
    grafcet4();
    printf("X40 = %d,X41 = %d,X42 = %d\n",X40 ,X41 ,X42 );
}

void Waiting(){
    printf("Waiting activate !!\n");
}

void ImageAquisition(){
    printf("ImageAquisition activate !!\n");
}

void TempHumiAqui(){
    printf("TempHumiAqui activate !!\n");
}

```

```

void ImageRecognition(){
    printf("ImageRecognition activate !!\n");
}

void TempHumiAnal(){
    printf("TempHumiAnal activate !!\n");
}

void ModeSelect(){
    printf("ModeSelect activate !!\n");
}

void WifiConnect(){
    printf("WifiConnect activate !!\n");
}

void BlynkDisplay(){
    printf("BlynkDisplay activate !!\n");
}

```

## 4. 軟體合成

### – Arduino code

```
- // Fill-in information from your Blynk Template here
- #define BLYNK_TEMPLATE_ID "TMPLWpNwHeAp"
- #define BLYNK_DEVICE_NAME "Final Blink"
-
- #define BLYNK_FIRMWARE_VERSION "0.1.0"
-
- #define BLYNK_PRINT Serial
- //#define BLYNK_DEBUG
-
- #define APP_DEBUG
-
- // Uncomment your board, or configure a custom board in Settings.h
- //#define USE_WROVER_BOARD
- //#define USE_TTGO_T7
- //#define USE_TTGO_T_OI
- //#define USE_ESP32C3_DEV_MODULE
- //#define USE_ESP32S2_DEV_KIT
-
- #define PIN_SERVO 16
-
- #include <Arduino.h>
- #include <ESP32Servo.h>
- #include <string.h>
- #include "DHTesp.h"
- #include "BlynkEdgent.h"
-
- DHTesp dht;
- int T, H;
-
- BlynkTimer timer;
-
- int condition = 0;
- Servo myservo;
-
- char userInput;
-
- void myTimer(){
-   TempAndHumidity data = dht.getTempAndHumidity();
-   T = data.temperature;
-   H = data.humidity;
-   Serial.print(F("Temperature : "));
-   Serial.print(T);
```

```

-   Serial.print(F("°C Humidity : "));
-   Serial.print(H);
-   Serial.println("%");
-
-   Blynk.virtualWrite(V0, T);
-   Blynk.virtualWrite(V1, H);
- }
-
- void control(int condition){
-
-     int i = 0;
-
-     while(i < 2){
-         if(condition == 1){
-             for(int pos = 0; pos < 360; pos += 1) {
-                 myservo.write(pos);
-                 delay(50);
-             }
-         }
-         if(condition == 2){
-             for(int pos = 0; pos < 360; pos += 2) {
-                 myservo.write(pos);
-                 delay(50);
-             }
-         }
-         if(condition == 3){
-             for(int pos = 0; pos < 360; pos += 5) {
-                 myservo.write(pos);
-                 delay(50);
-             }
-         }
-         if(condition == 4){
-             for(int pos = 0; pos < 360; pos += 7) {
-                 myservo.write(pos);
-                 delay(50);
-             }
-         }
-         i++;
-     }
- }
-
- void setup() {
-     // put your setup code here, to run once:
-     dht.setup(11, DHTesp::AM2302);
-     delay(100);

```

```

-   BlynkEdgent.begin();
-   timer.setInterval(100L, myTimer);
-
-   Serial.begin(115200);
-   myServo.attach(PIN_SERVO); // 設置舵機控制腳位
- }
-
- void loop() {
-     // put your main code here, to run repeatedly:
-     // python code 辨識人數
-     userInput = Serial.read();
-     //Serial.println(userInput);
-     /*if (userInput == '1') Serial.println('open1');
-     else if (userInput == '2') Serial.println('open2');*/
-
-     // 型別轉換
-     String a = String(userInput);
-     int pos = a.toInt();
-
-     // 感測器資料分析
-     if (T <= 25 && pos == 1) condition = 1;
-     else if (T <= 25 && pos == 2) condition = 2;
-     else if (T <= 25 || H >= 60 || pos == 3) condition = 3;
-     else condition = 4;
-
-     BlynkEdgent.run();
-     timer.run();
-     //控制模擬風扇階段
-     control(condition);
-
-     Blynk.virtualWrite(V2, pos);
-     Blynk.virtualWrite(V3, condition);
- }

```

– Python code

```
– # -*- coding: utf-8 -*-
– """OpenCV.ipynb
–
– Automatically generated by Colaboratory.
–
– Original file is located at
–   https://colab.research.google.com/drive/1WVIHqotG1T921\_jc7XyMsHHdqapHr\_Ua
– """
–
– #!pip install opencv-python
– #!python -m pip install pyserial
–
– import cv2
– import numpy as np
– import serial
– import time
–
– """# FaceRecognition
–
– """
–
– def detectFace(img):
–     filename = img.split(".")[0] # 取得檔案名稱(不添加副檔名)
–     img = cv2.imread(img) # 讀取圖檔
–     grayImg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 透過轉換函式轉為灰
–     階影像
–     color = (0, 255, 0) # 定義框的顏色
–
–     # OpenCV 人臉識別分類器
–     face_classifier = cv2.CascadeClassifier(cv2.data.harcascades +
–     "haarcascade_frontalface_default.xml")
–
–     # 調用偵測識別人臉函式
–     faceRects = face_classifier.detectMultiScale(
–         grayImg, scaleFactor=1.25, minNeighbors=2, minSize=(25, 25))
–
–     # 大於 0 則檢測到人臉
–     if len(faceRects):
–         # 框出每一張人臉
–         for faceRect in faceRects:
–             x, y, w, h = faceRect
–             cv2.rectangle(img, (x, y), (x + h, y + w), color, 2)
–     #print(len(faceRects))
```

```

- # 將結果圖片輸出
- cv2.imwrite(filename + "_face.jpg", img)
-
- return len(faceRects)
-
- val = detectFace('S6.jpg')
- print(val)
-
- """# Return Back to Arduino"""
-
- data_serial = serial.Serial('/dev/cu.usbserial-1120',115200)
-
- def getValue_on():
-
-     if val == 1:
-         data_serial.write(b'1')
-     if val == 2:
-         data_serial.write(b'2')
-     if val==3:
-         data_serial.write(b'3')
-     if val >= 4:
-         data_serial.write(b'4')
-
- while(1):
-     getValue_on()
-     time.sleep(5)

```

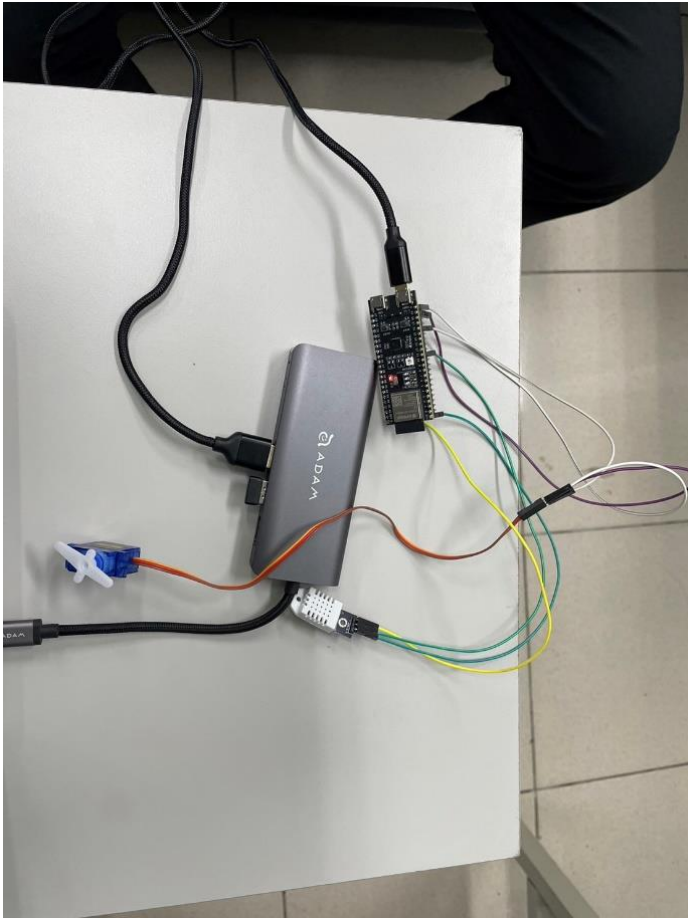
➤ 模型正確辨識出該環境有三人





## 5. 硬體驗證

硬體驗證



App Display



## 五、 未來展望

在我們實驗中的電風扇模式，是分成四種不同的 condition 去模擬現實生活中的電風扇狀態，這和過去在市面上流通的電風扇是一樣的。然而，若是想要增加商品的價值，我們可以不被電風扇的離散模式所侷限，將運用溫度、濕度、人數資料所分析出來的結果，直接對應到風扇的轉速，而非不同的 condition，這樣的結果會是連續性的，可以對應到目前較新型的變頻家電，是我們的專題能進一步改進的方向。

## 六、 團隊分工

姓名	工作內容
陳彥廷	Blynk wifi 連結、DHT22 感測器、硬體驗證、報告
徐裕翔	影像辨識、C code 生成、Grafcet 繪製、軟硬合成、報告
吳苡菡	SG90 感測器、Grafcet 繪製、硬體驗證、報告

## 七、 參考資料

1. Python serial input  
<https://swf.com.tw/?p=1188>
2. DHT 22  
<https://koding.work/use-esp-32-and-dht-sensor-to-detect-temp-and-humidity/>
3. SG 90  
<https://blog.jmaker.com.tw/arduino-servo-sg90/>
4. Blynk  
<https://blynk.io/>
5. 上課講義