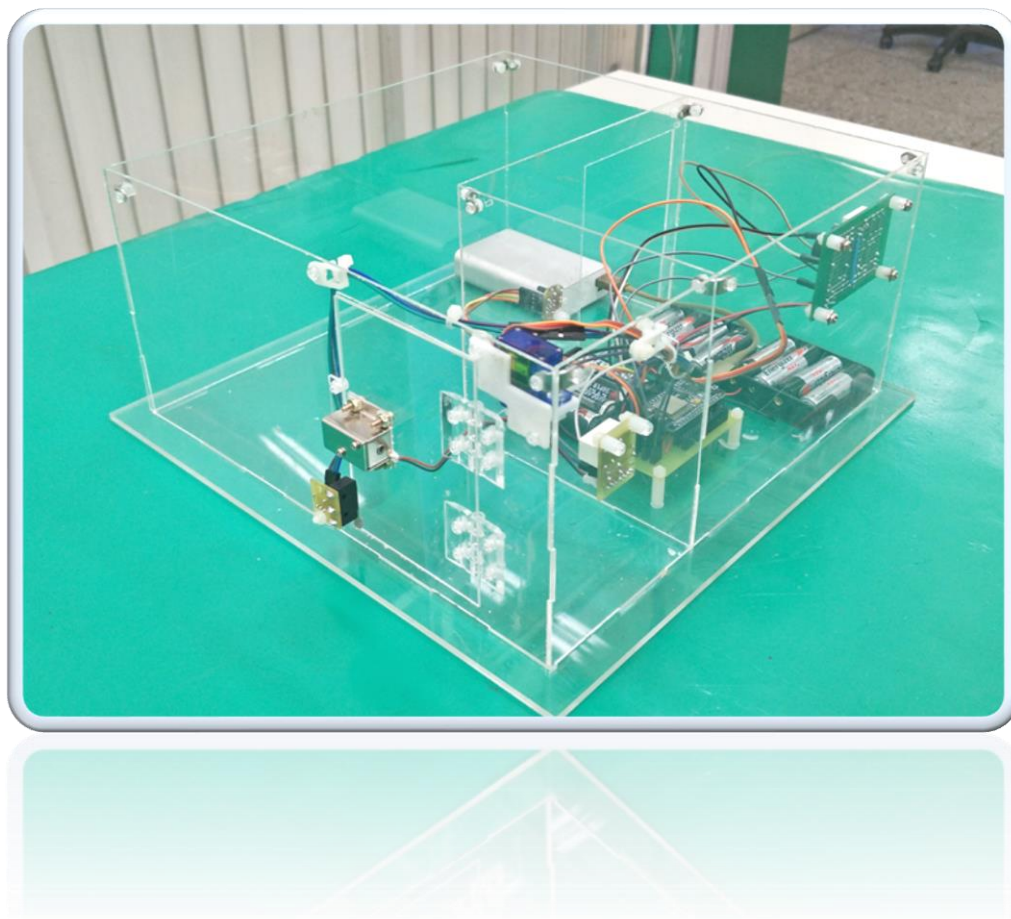


全國高級中等學校專業群科 109 年專題及創意製作競賽

「專題組」作品說明書



群別：電機與電子群

作品名稱：遠端防盜

關 鍵 詞：物聯網、遠端控制、居家安全

全國高級中等學校專業群科 109 年專題及創意製作競賽

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.....	1
肆、 研究方法及過程	2
一、 研究探討.....	2
二、 使用工具.....	2
三、 使用軟體與研究過程.....	3
伍、 研究結果.....	11
陸、 討論.....	12
一、 伺服器的穩定性.....	12
二、 微控器與模組之間的連接.....	12
三、 系統自動開啟	12
四、 功能的延伸.....	12
柒、 結論.....	12
捌、 參考資料及其他	13

表目錄

表 1. 課程名稱與專題用之相關性說明	1
表 2. 其他元件名稱與用途	2

圖目錄

圖 1. ESP8266 NODEMCU	2
圖 2. 由 Git Hub 下載程式碼，以擷取修改進行測試	3
圖 3. ESP8266 NODE MCU 目前與 aREST 連線的狀態	4
圖 4. ESP8266 NODE MCU 目前 D5(也就是 GPIO 14)腳位為低態	4
圖 5. D5 腳位設定為高態時，LED 亮燈	4
圖 6. D5 腳位設定成 OUTPUT	4
圖 7. D5 腳位設定成 INPUT	4
圖 8. 類比輸出	5
圖 9. 類比讀取	5
圖 10. APP 介面編排	5
圖 11. APP 初始動作的程式	6
圖 12. 比較文字的資料型態	7
圖 13. APP 關閉警報器的程式	8
圖 14. APP 關門及開門的程式	8
圖 15. 最後完成「讓微控器主動傳訊息到手機」程式(part I)	9
圖 16. 最後完成「讓微控器主動傳訊息到手機」程式(part II)	10
圖 17. 成品實體(由上朝下拍攝)	11
圖 18. 成品實體(由左至右拍攝)	11

壹、摘要

物聯網拉近分散的資訊，統整物與物的數位資訊，物聯網的應用領域主要的其中之一是智慧型環境（家庭、辦公、工廠）領域。我們家中因為被小偷偷竊，所以我們選擇製作遠端防盜系統配合現在物聯網的趨勢，設計一個能夠讓使用者在第一時間得知家中發生的狀況的專題，以方便做即時的應變。本專題使用 ESP8266 作為中樞，結合外部元件與雲端伺服器，偵測門的開閉、使用手機進行門的控制、模擬警報器，並且傳 LINE 通知在外家人的手機。

貳、研究動機

現今的社會家庭結構大都是以小家庭組成形式為主，很容易因為家中成員早上都要工作或上學之類的因素。屆時，家中會完全沒人，小偷就很容易趁虛而入。當你在公司裡打著電腦進行工作時，小偷可能已經將你家行竊完成！重點是你當下完全不知道有這件事情發生，而要經過數小時後發現家裡狀況才能得知。我們希望能透過設計一些簡單的機制，來防範這類事件的發生。

再者，大部分的家庭沒有充裕的資金裝設保全系統，但是家中與我們常用的手機都有網路。那我們就能將手邊的電子產品成為我們的遠端接收器，並能幫助我們即時得知此類狀況。因此我們的專題將會是一個相當經濟實惠且具有可行性的輔助防盜方案。

參、主題與課程之相關性或教學單元之說明

小組經由多次的討論依據高中學程課目名稱學習內容及平日生活經驗日常所需中，逕行製表下列的所學課程與本次專題的相關性說明內容。

表 1. 課程名稱與專題用之相關性說明

課程名稱	專題應用
專題製作	蒐集資料、創意思考、題目討論
計算機概論	物聯網基礎概論
單晶片實習	ESP8266 網路分享
雙極性接面電晶體	BJT 之工作原理
數位邏輯實習	程式撰寫、數位邏輯概念
特殊電機	伺服馬達

肆、研究方法及過程

一、研究探討

(一)專題是採用模型元件方式呈現，主要以「建構遠端監控的路徑」為核心，透過手機網路連結家中 Wi-Fi 進行門戶監控。期能達成四項功能，

第 1 項，偵測到門沒關的話，會傳訊息通知手機。

第 2 項，透過手機開關門。

第 3 項，偵測到疑似有人闖入時，觸發警報器，並傳訊息通知手機。

第 4 項，透過手機關閉警報器。

(二)使用主要和相關之元件製成模型實體，再利用相關網站提供電子程式碼範例，修改編碼，期達遠端監控目標四項功能，更進一步以手機 APP 介面觸控操作，以「按一顆按鈕，APP 就會自動查網址」藉此達到較為合理外部呈現。

二、使用工具

(一) 主要元件

ESP8266 NODEMCU 是整合 Wi-Fi 模組和微控器、體積小，且能用 Arduino IDE 編譯，網路上的範例資源也很豐富。



圖 1. ESP8266 NODEMCU

(二) 其他元件

表 2. 其他元件名稱與用途

元件名稱	用途	元件名稱	用途
微動開關	檢測門是否關上	電晶體(2N3904)	推蜂鳴器、繼電器
按鈕開關	模擬紅外線開關	繼電器(BD2-5-5)	推電磁鎖
伺服馬達(SG90)	關門用	穩壓 IC (LM7805)	伺服馬達電源
蜂鳴器(BQM-06)	模擬警報	電磁鎖	開門用
LED	模擬警報	電池	電源

三、 使用軟體與研究過程

- (一) 從網路搜尋物聯網相關網站 aREST，得知網站出版 DISCOVER THE aREST Framework 範例說明書，即由 Git Hub 下載書中我們所需的程式碼予以擷取修改並進行測試。

```
#include <ESP8266WiFi.h>+  
#include <PubSubClient.h>+  
#include <aREST.h>+  
+  
#define LED 14+  
+  
WiFiClient espClient;+  
PubSubClient client(espClient);+  
aREST rest = aREST(client);+  
char* device_id = "LED test"; //這邊建議設一個較為獨特的名稱，等一下會用到+  
+  
const char* ssid = "Wi-Fi 名稱";+  
const char* password = "Wi-Fi 密碼";+  
+  
void callback(char* topic, byte* payload, unsigned int length);+  
//=====+  
void setup(void){+  
  Serial.begin(115200);+  
  +  
  client.setCallback(callback);+  
  +  
  rest.set_id(device_id);+  
  rest.set_name("測試燈"); +  
  +  
  WiFi.begin(ssid, password);+  
  while (WiFi.status() != WL_CONNECTED) {+  
    delay(500);+  
    Serial.print(".");+  
  }+  
  Serial.println("");+  
  Serial.println("WiFi connected"); +  
  Serial.println("IP address: ");+  
  Serial.println("WiFi connected");+  
  +  
  char* out_topic = rest.get_topic();+  
  +  
  pinMode(LED,OUTPUT);+  
  char* out_topic = rest.get_topic();+  
  +  
  pinMode(LED,OUTPUT);+  
  digitalWrite(LED,LOW);+  
}+  
//=====+  
void loop() {+  
  rest.handle(client);+  
}+  
//=====+  
void callback(char* topic, byte* payload, unsigned int length) {+  
  rest.handle_callback(client, topic, payload, length);+  
} +  
  digitalWrite(LED,LOW);+  
}+  
//=====+  
void loop() {+  
  rest.handle(client);+  
}+  
//=====+  
void callback(char* topic, byte* payload, unsigned int length) {+  
  rest.handle_callback(client, topic, payload, length);+  
} +  
}
```

圖 2. 由 Git Hub 下載程式碼，以擷取修改進行測試

(二) 放入 Arduino IDE 進行編譯去測試控制單顆 LED，連接好線路後，再讓 ESP8266 NODE MCU 連接上 Wi-Fi。經過一連串的測試後發現，可以利用搜尋引擎來監控 ESP8266 NODE MCU，其具體功能如下。

1. 在網址的部分輸入 [https://cloud.arest.io/LED test/id](https://cloud.arest.io/LED%20test/id) 可顯示網頁回應，代表著 ESP8266 NODE MCU 目前與 aREST 連線狀態。

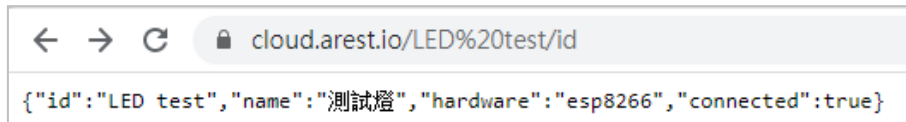


圖 3. ESP8266 NODE MCU 目前與 aREST 連線的狀態

2. 在網址的部分輸入 [https://cloud.arest.io/LED test/digital/5](https://cloud.arest.io/LED%20test/digital/5) 可顯示網頁回應，代表著 ESP8266 NODE MCU 目前 D5(也就是 GPIO 14)腳位為低態。



圖 4. ESP8266 NODE MCU 目前 D5(也就是 GPIO 14)腳位為低態

3. 在網址的部分輸入 [https://cloud.arest.io/LED test/digital/5/1](https://cloud.arest.io/LED%20test/digital/5/1) 可顯示網頁回應，將 D5 腳位設定為高態時，LED 亮燈。

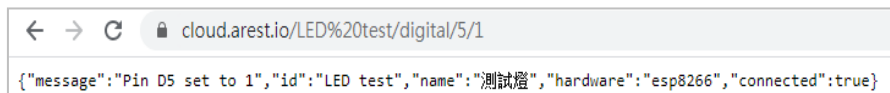


圖 5. D5 腳位設定為高態時，LED 亮燈

4. 其他的功能像是輸入 [https://cloud.arest.io/LED test/mode/5/o](https://cloud.arest.io/LED%20test/mode/5/o) 可顯示網頁回應，將 D5 腳位設定為 OUTPUT。



圖 6. D5 腳位設定成 OUTPUT

5. 輸入 [https://cloud.arest.io/LED test/mode/5/i](https://cloud.arest.io/LED%20test/mode/5/i) 可顯示網頁回應，將 D5 腳位設定為 INPUT。

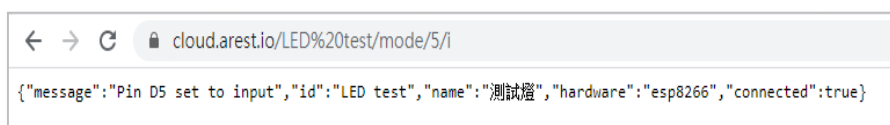


圖 7. D5 腳位設定成 INPUT

- 輸入 [https://cloud.arest.io/LED test/analog/5/500](https://cloud.arest.io/LED%20test/analog/5/500) 可顯示網頁回應，呈現類比輸出。

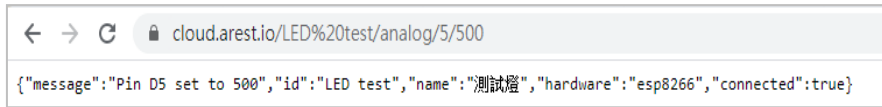


圖 8. 類比輸出

- 輸入 [https://cloud.arest.io/LED test/analog/5](https://cloud.arest.io/LED%20test/analog/5) 可顯示網頁回應，呈現類比讀取。

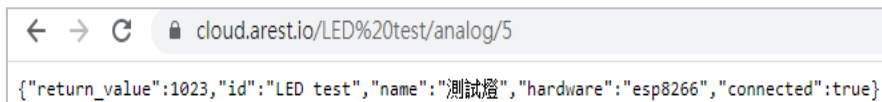


圖 9. 類比讀取

透過搜尋引擎功能可以做到這些功能，可作出檢查微控器是否上線、數位輸出、數位讀取、設定工作模式、類比輸出、類比讀取，上述的功能本組擷取前三項為該專題使用，所以將上述的範例程式碼加入本專題的程式中。

- 讀寫能力某種程度上達成遠端監控功能，但手動輸入網址對現代人不具親合力。所謂科技始終來自於人性，構成本組繼續後製 APP 的想法，期望達成「按一顆鈕，APP 就會自動查網址」藉此可達較為合理外部呈現。利用 APP Inventor 2 來製作 APP，下列為 APP 的介面和程式，介面編排設計為三顆按鈕、兩個標籤、五個網路元件。



圖 10. APP 介面編排

2. 程式的細部主要可設定四項功能，分別為開啟 APP 的初始動作、關閉警報器、開門動作、關門動作。

(1) 初始動作：建立完一個空清單之後，先關閉所有按鈕，讓網路 1 向「/id」執行 GET 請求，GET 請求會將網頁的所有內容抓下來，當然也代表 APP 查詢了「/id」，再將抓下來的文字做比對，判斷微控器是否上線，如果是上線狀態的話，則致能所有按鈕。

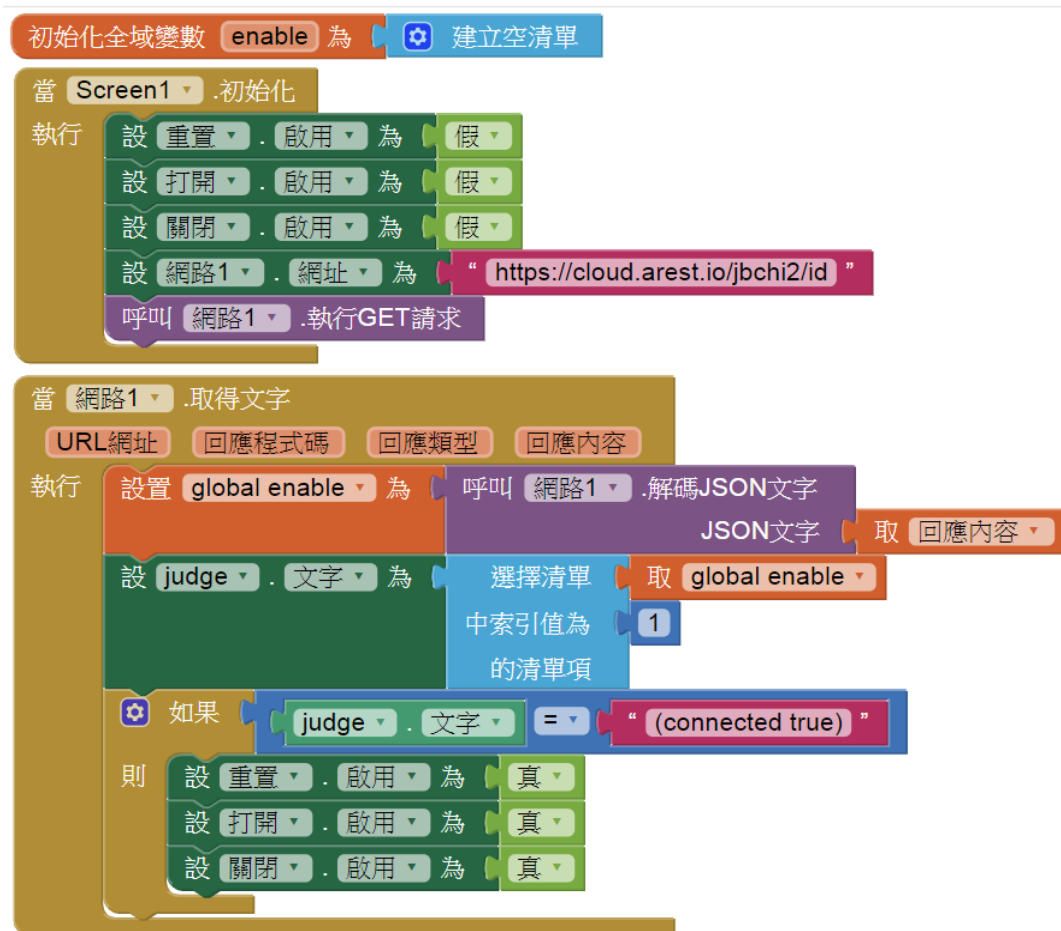


圖 11. APP 初始動作的程式

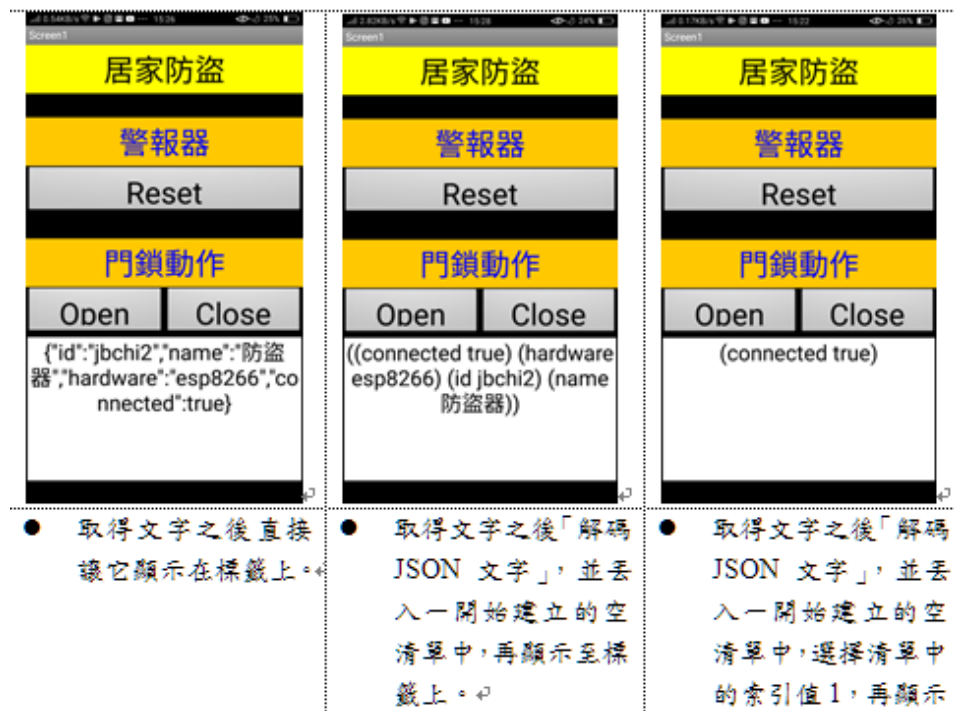


圖 12. 比較文字的資料型態

(2) 上圖說明介面下的程式碼轉換成最下方的文字解碼。經過測試結果，可產生下列的內部確定。

- 將文字解碼並丟入清單後，冒號的部分會被括號選取起來，然後重新排序。
- 排序後括號內的文字會被編號。(舉例來說，如果取索引值 1 的話，會顯示「(connected true)」，取索引值 2 的話，會顯示「(hardware esp8266)」，可以此類推設定其他三項功能。

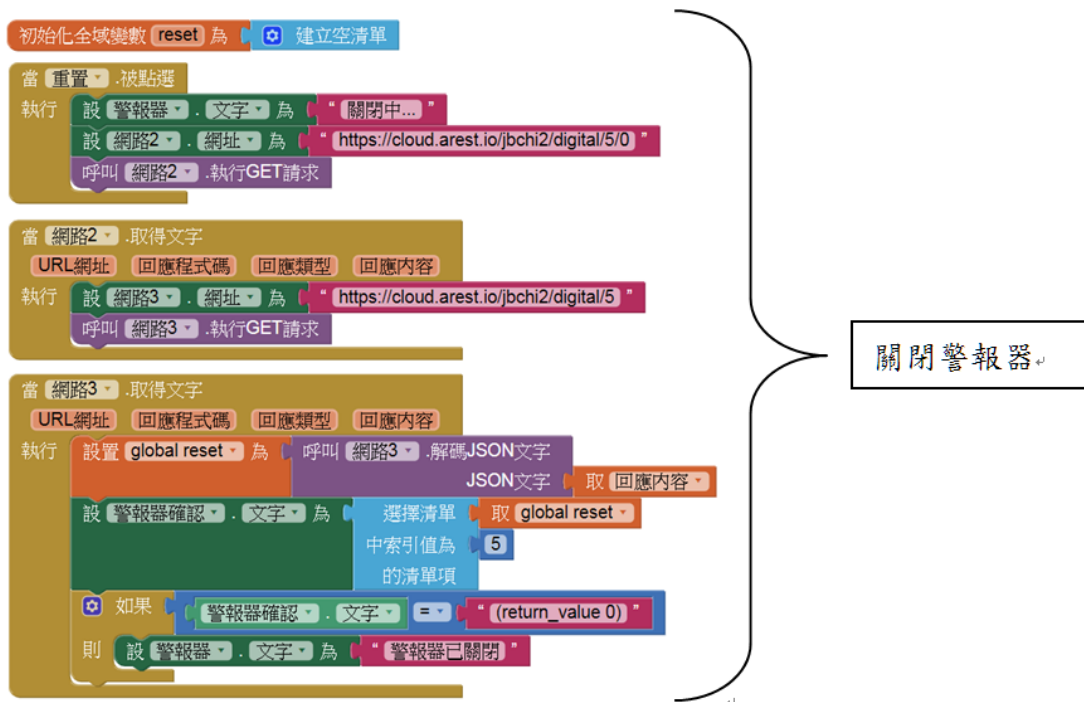


圖 13. APP 關閉警報器的程式



圖 14. APP 開門及關門的程式

透過 aREST 與 APP Inventor 2 能夠辦到「透過手機開關門」(第 2 項功能)和「透過手機關閉警報器」(第 4 項功能),但無法達成「偵測到門沒關的話,會傳訊息通知手機」(第 1 項功能)及「偵測到疑似有人闖入時,觸發警報器,並傳訊息通知手機」(第 3 項功能)。要達成第 2 項和第 4 項功能,現有的工具「digital/5/1」,能「讓 APP 查網址」辦到,但要達成第 1 項和第 3 項,以現有的工具「digital/5」,會有「不斷讓 APP 查網址」問題,這是不合理的

情況，造成問題一、手機無法負荷；問題二、會消耗流量大，以致於功能效能非常差。

- (3) 增加機制「讓微控器主動傳訊息到手機」，網路搜尋 IFTTT 網站提供範例程式碼，可以將訊息到手機，故擷取所需範例程式碼，測試讓 LED 燈亮後，可傳訊息至手機。程式編譯完並接好線後，到 IFTTT 註冊，並創建 applets。
- (4) IFTTT 是 If This Then That 的縮寫，把 This 想像是觸發條件，That 是動作，觸發條件選用的是 webhooks，讓網站跟微控器做溝通，動作選用的是 LINE，條件觸發後讓網站傳 LINE 至手機。經測試結果成功，達成「讓微控器主動傳訊息到手機」，故將範例程式與主程式加入專題的程式。

```
~
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <aREST.h>
#include <Servo.h>

#define RedLine1 5
#define RedLine2 4
#define JBCHI 14
#define OpenLock 13
#define LockSensor 15
#define ServoSignal 12

int buzzerPin=16;
Servo myServo;

WiFiClient espClient;
PubSubClient client(espClient);
aREST rest = aREST(client);

char* device_id = "jbchi2";

const char* ssid = "lancehouse";
const char* password = "f29548778f";

const char* host = "maker.ifttt.com";
const char* privateKey = "j9TgZ_jBID8bmVjYxWxoSBC-FitLkuegLyxxfD2PzmA";

void callback(char* topic, byte* payload, unsigned int length);
void send_event(const char* event);
//=====
void setup(void){
  Serial.begin(115200);
  client.setCallback(callback);
  rest.set_id(device_id);
  rest.set_name("防盜器");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println("WiFi connected");
  char* out_topic = rest.get_topic();

  pinMode(RedLine1, INPUT_PULLUP);
  pinMode(RedLine2, INPUT_PULLUP);
  pinMode(JBCHI, OUTPUT);
  pinMode(OpenLock, OUTPUT);
  pinMode(LockSensor, INPUT_PULLUP);
  pinMode(ServoSignal, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  digitalWrite(JBCHI, LOW);
  digitalWrite(OpenLock, LOW);
  digitalWrite(ServoSignal, LOW);

  myServo.attach(0);
  myServo.write(90);
}
//=====
void loop() {
  rest.loop(client);
  //=====紅外線判斷
  if (digitalRead(RedLine1) == LOW and digitalRead(RedLine2) == LOW){
    digitalWrite(JBCHI, HIGH);
  }
  if (digitalRead(JBCHI) == HIGH){
    alarmBeep(buzzerPin);
  }
}
```

圖 15. 最後完成「讓微控器主動傳訊息到手機」程式(part I)

```

        send_event("jbchi");
        delay(500);
    }
    //=====伺服馬達動作
    if (digitalRead(ServoSignal) == HIGH){
        myServo.write(0);
        delay(1250);
    }
    myServo.write(90);
    delay(1250);
    digitalWrite(ServoSignal,LOW);
}
//=====電子鎖動作
if (digitalRead(OpenLock) == HIGH){
    delay(500);
    digitalWrite(OpenLock,LOW);
}
//=====微動開關
if(digitalRead(LockSensor) == HIGH){
    send_event("door lock");
    delay(5000);
}
rest.handle(client);
}
//=====
void alarmBeep(int pin) {
    tone(pin, 1000, 1000);
    delay(2000);
}
//=====
void callback(char* topic, byte* payload, unsigned int length) {
    rest.handle_callback(client, topic, payload, length);
}
//=====
void send_event(const char *event)
{
    Serial.print("Connecting to ");
    Serial.println(host);
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("Connection failed");
        return;
    }

    // Create a URI for the request
    String url = "/trigger/";
    url += event;
    url += "/with/key/";
    url += privateKey;

    Serial.print("Requesting URL: ");
    Serial.println(url);

    // This will send the request to the server
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "Connection: close\r\n\r\n");

    // Read all the lines of the reply from server and print them to Serial,
    // the connection will close when the server has sent all the data.
    while(client.connected())
    {
        if(client.available())
        {
            String line = client.readStringUntil('\r');
            Serial.print(line);
        } else {
            // No data yet, wait a bit
            delay(50);
        }
    }
    Serial.println();
    Serial.println("closing connection");
    client.stop();
}
}

```

圖 16. 最後完成「讓微控器主動傳訊息到手機」程式(partII)

伍、研究結果

我們是採用模型的方式來呈現作品，成品實體如下。貼紅色膠帶式房屋模型的正面，主要分成大門和屋內門，大門旁的伺服馬達用來關門、電磁鎖通電吸磁便能把門推開、微動開關用來偵測門是否關閉。另外屋內門旁裝設紅外線開關(這裡用按鈕開關模擬)，被觸碰到的話會啟動警報器(通知手機與啟動響鈴)。

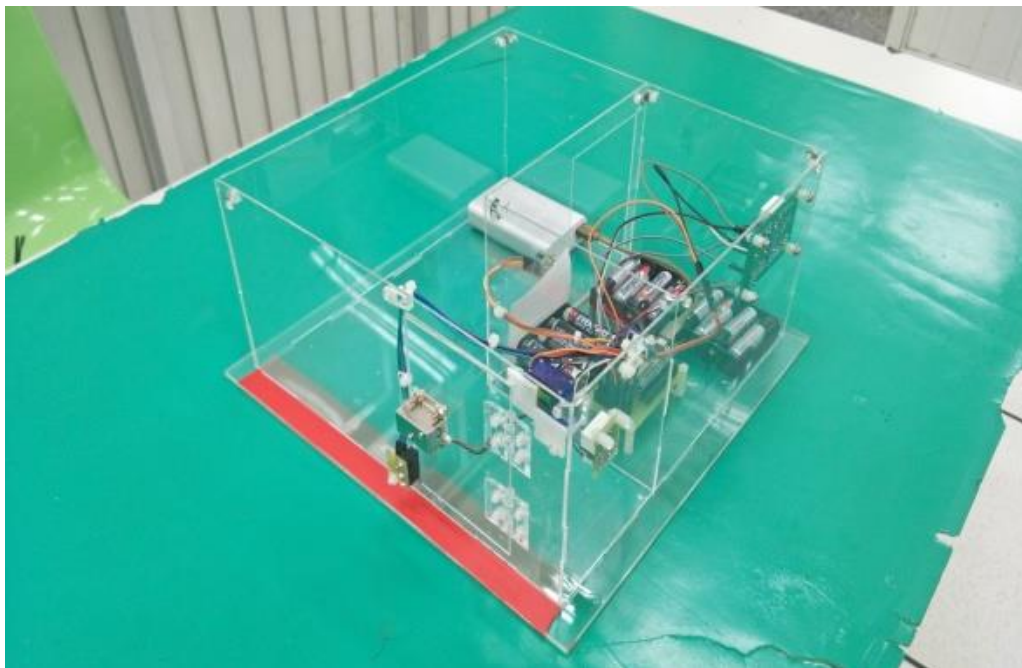


圖 17. 成品實體(由上朝下拍攝)

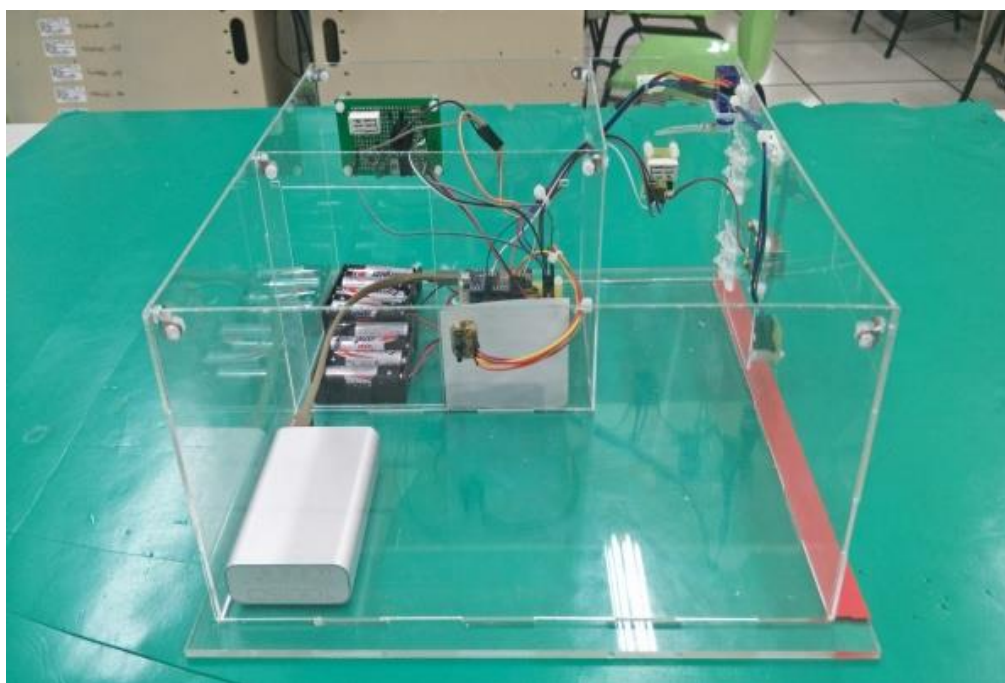


圖 18. 成品實體(由左至右拍攝)

陸、討論

一、 伺服器的穩定性

迫於製作專題的時間與人力有限，誠如各位所見，我們使用的伺服器是網路上別人免費提供的，因此不能夠保證永遠都會開啟，沒開啟就等於專題沒有功能，之前就有過伺服器故障的經驗，這是一個蠻嚴重的問題，因此如果之後還有充裕的時間，自行架設一個伺服器來作為遠端傳輸的中繼點也較為理想。

二、 微控器與模組之間的連接

如果我們模型作成實體，使用實體線連接可能不太合適，裝置和裝置間的距離可能差上好幾公尺，微控器發出的電壓訊號太小，可能會被線路消耗掉，所以有機會的話，希望能讓裝置間的溝通改成用藍芽或 Wi-Fi 等短距離的無線傳輸，而非實體線路。

三、 系統自動開啟

我們專題的出發點是「小家庭會因為家中成員在同一時間都要通勤，因此頓時家中會沒有人」，所以得要由最後一個出門的人開啟系統電源，或是第一個到家的人關閉系統電源，這是一件相當不自然的事，因為是人很可能會忘記，忘記的話就沒有系統的保障，或是容易收到假警報，因此我們希望之後也能夠加上影像辨識，得知有人看家時自動關閉系統，沒人看家時自動開啟系統。

四、 功能的延伸

本專題的核心是「物聯網」，在製作的過程中找到了遠端監控的方法，那其實除了做防盜之外，只要加上其他感測器或模組，就能應用在更多情況上，比如說：相信大家都有出門忘記關冷氣的經驗吧，只要在冷氣的電源線路上加裝繼電器，讓他與 ESP8266 連接，即便屋主不在家，也能在外頭確實將冷氣關閉，加裝簡單的裝置，卻能有意想不到且實用的功能。

柒、結論

在製作專題的過程中，我們遇到了許多困難，不管是運用學過或們學過的知識，都遇到許多困難，學過的部分像是科裡的老師們常說的：「理論和實際會有落差」這也是親身經驗，那沒學過的部分呢？當初為了要達成「真遠端」，在沒

有太多相關背景知識的條件下，得花費了大量的時間上網找資料，有時會因為一直沒有找到方法，而讓專題的進度停滯很長一段時間，期間是相當煎熬的，因為可能會做不出東西來，在高壓的情況下還是層層突破難關，這個「無論如何都要設法解決問題」的能力，相信是我們在這次專題中最大的收穫。

捌、參考資料及其他

<1>Arduino 筆記(38)：透過 IFTTT 發送 DHT-11 的溫濕度值到 Line 群組

<https://atceiling.blogspot.com/2019/07/arduino38dht-11-line.html>

<2>Arduino 筆記(38)：透過 IFTTT 發送 DHT-11 的溫濕度值到 Line 群組

<https://atceiling.blogspot.com/2019/07/arduino38dht-11-line.html>

<3>Control Every Device From Everywhere with aRest

[https://www.hackster.io/Levantino/control-every-device-from-everywhere-with-a-rest-d79a](https://www.hackster.io/Levantino/control-every-device-from-everywhere-with-a-rest-d79a3d)

[3d](#)