

# 高英高級工商職業學校

Kao Ying Industrial Commercial Vocational High School

## 教師專業精進自主研習報告



## 實現以 ESP32-CAM 做居家安全 監控系統

老師姓名： 蘇志雄 老師

科 別： 電機科

中 華 民 國 109 年 08 月

## 中文摘要

現代大多數人的工作皆是屬於早出晚歸類型，大部分的時間不在家，因此也使被闖空門的機會大增。有鑑於此，我們決定製作一個居家安全監控系統，用以提高居家安全性，不但可以利用硬體拍照回傳，更可以零時差的配合 LINE 系統發送警報。

本研究是一種以「ESP32-CAM」為基礎的居家安全監控系統，在有人經過時，可以即時拍照並發送文字及圖片訊息到自己的 Line 上來通知我們有陌生人靠近，以便我們可以做後續的處理；本作品動作測試皆無誤，系統已可以達到真正的居家安全功能，並實際運用在日常生活上。

關鍵字：門禁監控、ESP32-CAM

## 目 錄

中文摘要.....	i
目錄.....	ii
表目錄.....	iii
圖目錄.....	iv
壹、前言.....	1
一、研究動機.....	1
二、研究目的.....	1
三、研究架構.....	1
貳、理論探討.....	2
一、系統方塊圖.....	2
二、軟、硬體介紹.....	2
參、專題研究過程與方法.....	10
一、製作材料.....	10
二、組裝電路圖.....	10
肆、研究成果.....	12
一、作品完成圖.....	12
二、Arduino IDE 程式.....	12
三、序列埠監控畫面.....	18
四、LINE Notify 傳送文字訊息及圖片畫面.....	18
伍、研究結論與建議.....	19
陸、參考文獻.....	19

## 表目錄

表 2-1：材料及用途說明 .....	3
表 3-1：作品使用材料 .....	10
表 3-2：HC-SR04 超音波感測器模組與 ESP32-CAM 接腳對照表.....	12

## 圖目錄

圖 1-1：研究架構圖 .....	1
圖 2-1：系統方塊圖 .....	2
圖 2-2：ESP32-CAM 開發板及 OV2640 攝像頭 .....	4
圖 2-3：Arduino IDE 偏好設定頁面 .....	5
圖 2-4：Arduino IDE/工具設定步驟 .....	5
圖 2-5：HC-SR04 超音波感測器模組 .....	6
圖 2-6：超音波感測器測距示意圖 .....	6
圖 2-7：Arduino IDE 程式 .....	7
圖 2-8：LINE Notify .....	7
圖 2-9：LINE Notify 連動服務頁面 .....	8
圖 2-10：LINE Notify 發行權杖頁面 .....	9
圖 2-11：LINE Notify 權杖代碼頁面 .....	9
圖 2-12：LINE Notify 訊息通知 .....	10
圖 3-1：ESP32-CAM 與超音波模組組裝電路圖 .....	11
圖 3-2：CH340 模組接線圖 .....	11
圖 4-1：作品全貌 .....	12
圖 4-2：序列埠監控視窗畫面 .....	18
圖 4-2：LINE Notify 傳訊文字訊息及圖片畫面 .....	18

## 壹、前言

### 一、研究動機

現代大多數人的工作皆是屬於早出晚歸類型，大部分的時間不在家，因此也使被闖空門的機會大增。有鑑於此，我們決定製作一個居家安全監控系統，用以提高居家安全性，不但可以利用硬體拍照回傳，更可以零時差的配合 LINE 系統發送警報。

本研究是一種以「ESP32-CAM」為基礎的居家安全監控系統，在有人經過時，可以即時拍照並發送文字及圖片訊息到我們的 Line 上來通知我們有陌生人靠近，以便我們可以做後續的處理。

### 二、研究目的

本系統務求可以達到真正的居家安全功能，並實際運用在日常生活上，對於系統要求有幾個目的：

- (一)能夠在家中的任意位置架設監控系統。
- (二)除了文字訊息，還要加上及時的影像。
- (三)與物聯網做結合，持續 24 小時監控不斷。

### 三、研究架構

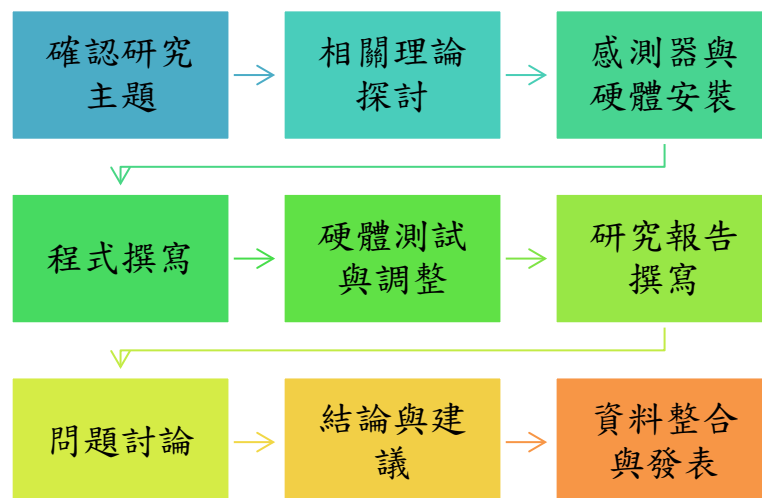


圖 1-1：研究架構圖

資料來源：研究者自行繪製

## 貳、理論探討

要達成本專題，需要測試並完成以下步驟，包括正確連接 ESP32-CAM 與超音波感測器、安裝 ESP32-CAM 核心程式、設定超音波感測器偵測距離以及申請 LINE Notify 等，以下分別就上述主題來研究與探討。

### 一、系統方塊圖：

系統使用 ESP32-CAM 開發板為基礎來製作居家安全監控系統，作品使用超音波感測器來負責偵測人體，當在一定距離以內（預設 30CM）偵測到有人經過時，ESP32-CAM 會先自動拍照並儲存在卡槽內，再將照片透過 WIFI 傳送到 LINE Notify 網站內，再依照我們所設定要傳訊的人或群組來發送文字及圖片訊息，以下是系統之整體架構圖：

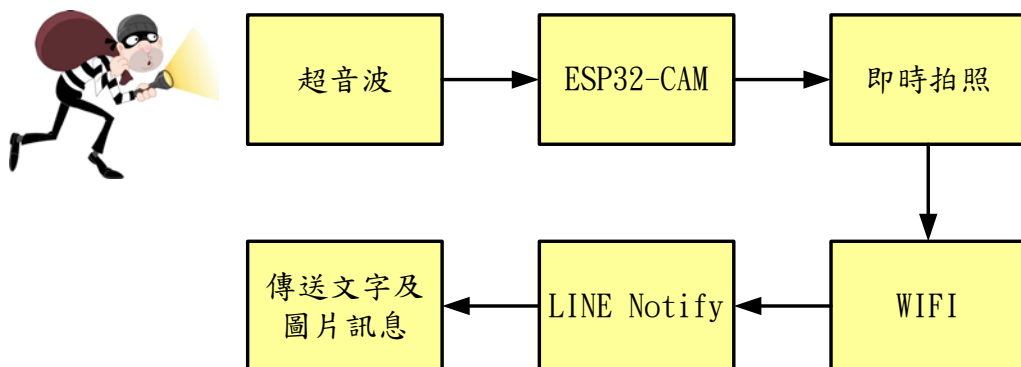


圖 2-1：系統方塊圖

資料來源：研究者自行繪製

### 二、軟、硬體介紹：

本作品運用到以下硬體，其功能及用途詳述如下表：

- ESP32-CAM 開發板
- 超音波感測器模組
- USB 轉 TTL 模組 (CH340)
- 杜邦線
- 智慧型手機

表 2-1：材料及用途說明

名稱	用途說明
ESP32-CAM 開發板	負責執行主程式，同時兼具 WIFI 及拍照功能
超音波感測器模組	負責偵測人體並將訊號傳送至 ESP32CAM 內
USB 轉 TTL 模組	1.Arduino IDE 與 ESP32-CAM 溝通界面。 2.燒錄 Arduino IDE 程式至 ESP32-CAM 開發板內。 3.連接 ESP32-CAM 與電源設備。
智慧型手機	接收 LINE Notify 所傳送的訊息

資料來源：研究者自行繪製

#### (一) ESP32-CAM 開發板：

ESP32-CAM 是一片以 ESP32 為基礎的開發版，負責執行主程式及拍照功能，ESP32-CAM 本身內建 WIFI 及藍芽 BLE 之外，再加上 OV2640 兩百萬畫素相機模組、閃光燈模組等所組成的開發版。

ESP32-CAM 是一個非常小的相機模塊，除了 OV2640 相機和幾個用於連接外圍設備的 GPIO 外，它還具有 microSD 卡插槽，可用於儲存相機拍攝的圖像。

ESP32-CAM 與過往的 ESP8266 Wifi 模組一樣相容於 Arduino IDE，使用者可不必另學程式語言，開發板上可接上各種電子元件如 LED 燈、喇叭、馬達、開關、溫濕度感測器、紅外線發射與接收器、LCD 顯示裝置以及 GPS 等各種模組。

更重要的是，ESP32-CAM 可透過 Arduino 的一個範例程式 CameraWebServer，就可以在開發板上建立一個網頁伺服器 (Web Server)，而後只要用瀏覽器輸入網址 (即



輸入 192.168.x.x)，就可以看到攝影機角度的影像內容，進一步也可以作一些影像辨識應用。

ESP32-CAM 有以下幾個特點：

- 1.採用低功耗雙核 32 位 CPU，可作應用處理器。
2. 802.11b/g/n Wi-Fi BT SoC 模塊。
- 3.頻率高達 240MHz，運算能力高達 600 DMIPS。
- 4.內置 520 KB SRAM，外置 4M PSRAM。
- 5.GPIO 支援 UART/SPI/I2C/PWM/ADC/DAC 等接口。
- 6.可使用 115200 bps 的 Baund rate。
- 7.內建 SD 卡匣，可儲存相機拍攝的圖像。
- 8.相容 OV2640 和 OV7670 攝像頭，內置閃光燈。

以上種種好處是我們使用 ESP32-CAM 來做居家安全系統的原因。

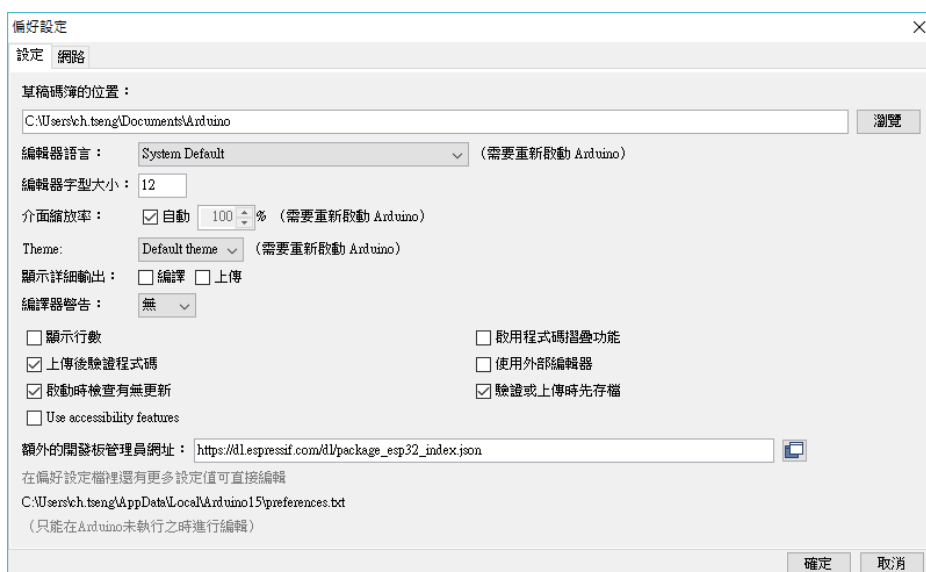


圖 2-2：ESP32-CAM 開發板及 OV2640 攝像頭

資料來源：<https://makerpro.cc/2019/08/esp32-cam-unboxing/>

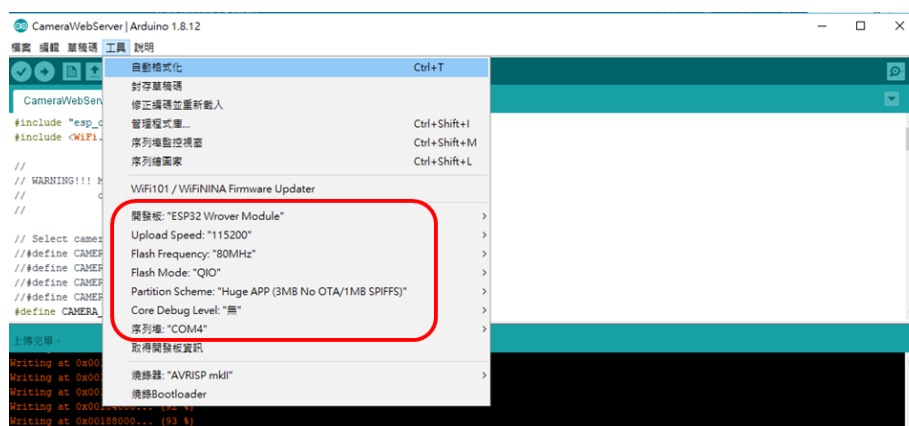
## (二) 安裝 ESP32-CAM 核心程式

第一次使用 ESP32-CAM 時，需要安裝 ESP32 的 libraries，首先打開 Arduino IDE 後，在「偏好設定」中的「額外的開發管理員模組」中輸入以下網址：[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)，按確定後便開始下載。



資料來源：研究者自行繪製

接著，進入開發板管理員，輸入 ESP32 關鍵字找到 ESP32 板子，直接安裝完成後，從工具選項中，選擇板子及設定如下，便完成 ESP32-CAM 的安裝了。



資料來源：研究者自行繪製

### (三) HC-SR04 超音波感測器模組

我們使用 HC-SR04 超音波感測器模組做人體距離的偵測，HC-SR04 超音波感測器模組（Ping）是由超音波發射器、接收器和控制電路所組成，當它被觸發的時候，會發射一連串 40K HZ 的聲波並且從離它最近的物體接收回音。超音波是人類耳朵無法聽見的聲音，因為它的頻率很高。



圖 2-5：HC-SR04 超音波感測器模組

資料來源：<http://coopermaa2nd.blogspot.com/2012/09/hc-sr04.html>

下圖所示為超音波測量距離的方法，是測量聲音在感測器與物體之間往返經過的時間，聲音在空氣中的傳播速度大約是每秒 340 公尺，傳播速度會受溫度影響，溫度愈高、傳播速度愈快。

假設以每微秒 340 公尺計算 ( $1000000/340*100=29.4$  microseconds)，在四捨五入後，可知聲音傳播 1 公分所需的時間為 29 microseconds (百萬分之一秒)。

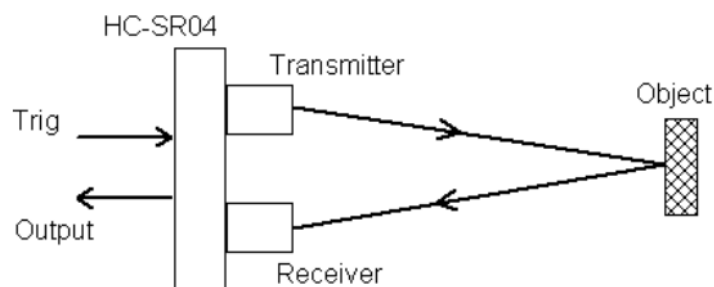


圖 2-6：超音波感測器測距示意圖

資料來源：<http://coopermaa2nd.blogspot.com/2012/09/hc-sr04.html>

由於超音波從發射到返迴是兩段距離，因此在計算時必須將結果除以 2 才是正確的物體距離。所以我們可以利用底下的公式算出物體距離：距離=timing/29/2 (timing：測量得到的音波傳播時間、距離單位：公分)。

基於以上理論，我們在程式上設定當距離低於 30cm 時，系統即觸發 ESP32-CAM 做拍照的動作，並且傳送文字及圖片訊息至 LINE Notify。

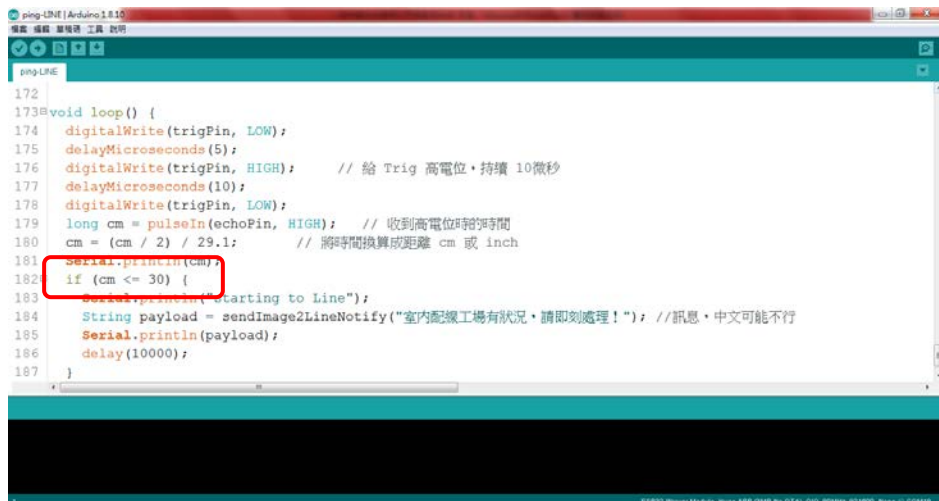


圖 2-7：Arduino IDE 程式  
資料來源：研究者自行繪製

#### (四) LINE Notify 自動傳訊通知：

若要正確傳送文字與圖片訊息，則要申請 LINE Notify 權杖，以便做為系統的驗證，LINE Notify 是 LINE 在 2016 年推出的一個服務，只要進行登錄，就可以透過 LINE Notify 官方帳號來「自動」幫您推播通知訊息。我們可以透過 LINE Notify 免費群發文字與圖片訊息，但是圖片一小時只能傳 50 張。



圖 2-8：LINE Notify  
資料來源：[https://notify-bot.line.me/zh\\_TW/](https://notify-bot.line.me/zh_TW/)

在與網站服務連動完成後，LINE 所提供的官方帳號「LINE Notify」將會傳送通知，也就是利用訂閱的方式

來接收網站或 Facebook 通知，目前可連動的服務有：GitHub、IFTTT 與 Mackerel；例如使用 IFTTT 來設定條件與觸發，即可利用 LINE 發送訊息通知，無須額外下載應用程式。

申請 LINE Notify 權杖的步驟如下：

1. 打開 LINE Notify 的網站，使用自己的 LINE 帳號登入，登入後滑鼠移至上方個人帳號，選擇「個人頁面」，在個人頁面可以發行「權杖」。

權杖的作用在於讓「連動的服務」可以透過 LINE Notify 來發送訊息通知，而發行後的權杖與其名稱，就會出現在下方的清單中，如果你之前有申請過像是 IFTTT 的連動通知，就會發現上方出現 IFTTT 的服務。

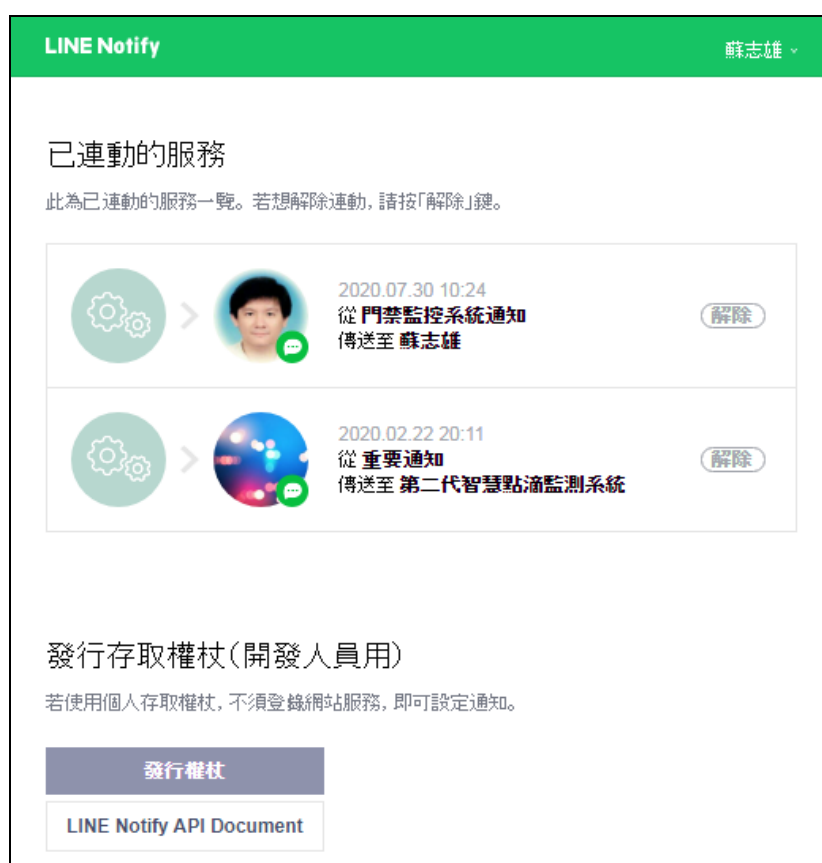


圖 2-9：LINE Notify 連動服務頁面

資料來源：<https://notify-bot.line.me/my/>

2. 點選「發行權杖」，指定權杖名稱（傳送通知訊息時所顯示的名稱）以及選擇是要一對一接收、或是讓群組也

可以接收通知。



圖 2-10：LINE Notify 發行權杖頁面

資料來源：<https://notify-bot.line.me/my/>

- 3.點選「發行」，會出現一段權杖代碼，這段代碼「只會出現一次」，複製這段代碼，先找個地方貼上並儲存這段代碼，就可以點選下方按鈕「關閉」。



圖 2-11：LINE Notify 權杖代碼頁面

資料來源：<https://notify-bot.line.me/my/>

- 4.完成後就會發現在連動的服務裡，出現了我們自訂的連動服務；同時，LINE 訊息裡也會收到 LINE Notify 發出「已發行個人存取權杖」的訊息，看到這一步，表示 LINE Notify 已經設定完成。

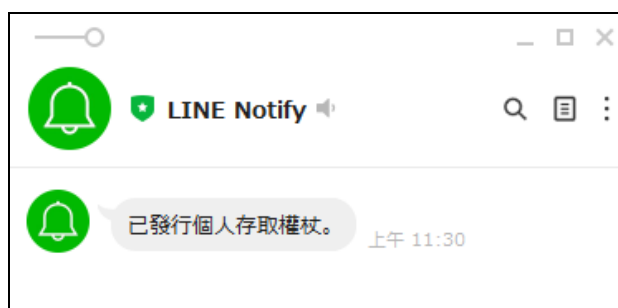


圖 2-12：LINE Notify 訊息通知

資料來源：研究者自行繪製

## 參、專題研究過程與方法：

### 一、製作材料：

表 3-1：作品使用材料

品名／規格	數量
ESP32-CAM 開發板	1 片
超音波感測器模組	1 個
CH340 模組	1 個
麵包板	1 片
杜邦線	若干
智慧型手機	1 支

資料來源：研究者自行繪製

### 二、組裝電路圖：



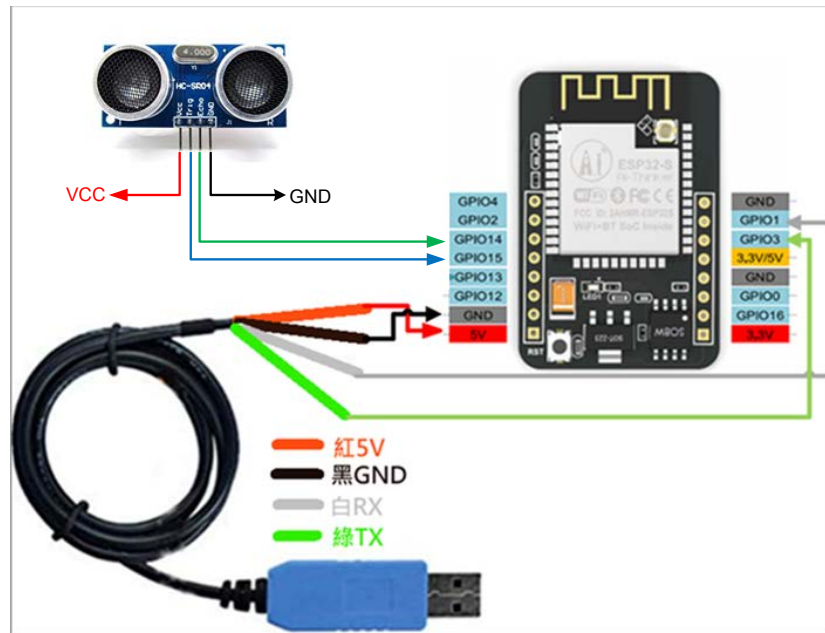


圖 3-1：ESP32-CAM 與超音波模組組裝電路圖

資料來源：使用者自行繪製

由於成本及體積因素，ESP-CAM 沒有內建 USB 燒錄介面，因此必須另外準備 USB 轉 TTL 模組（CH340），其接法如下：

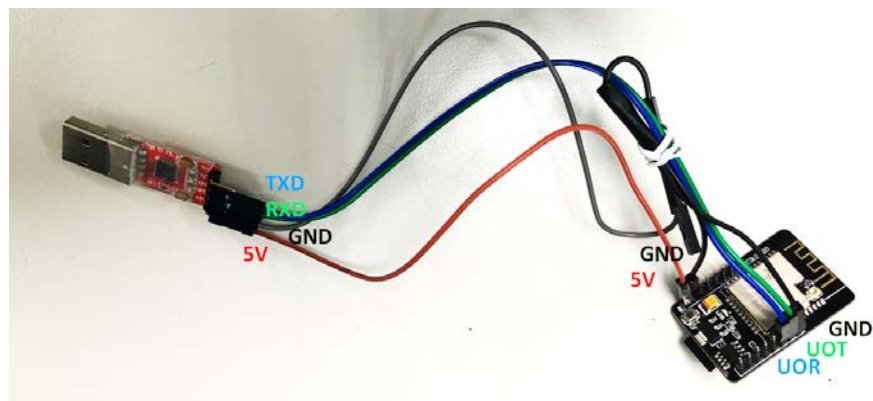


圖 3-2：CH340 模組接線圖

資料來源：<https://chtseng.wordpress.com/2020/05/01/>

常見的 HC-SR04 超音波感測器模組都有 4 個腳，除了 VCC 和 GND 以外，Trig 就是發送訊號，Echo 就是接收返回的訊號。然後我們就能利用一發一收，去算出中間的距離了。但是這太麻煩了，一般我們都會先安裝適合的函式庫，這樣可以省掉這些計算的麻煩。



表 3-2：HC-SR04 超音波感測器模組與  
ESP32-CAM 接腳對照表

HC-SR04	ESP32-CAM
Vcc	+5V
Trig	GPIO
Echo	GPIO
GND	GND

## 肆、研究成果

### 一、作品完成圖：

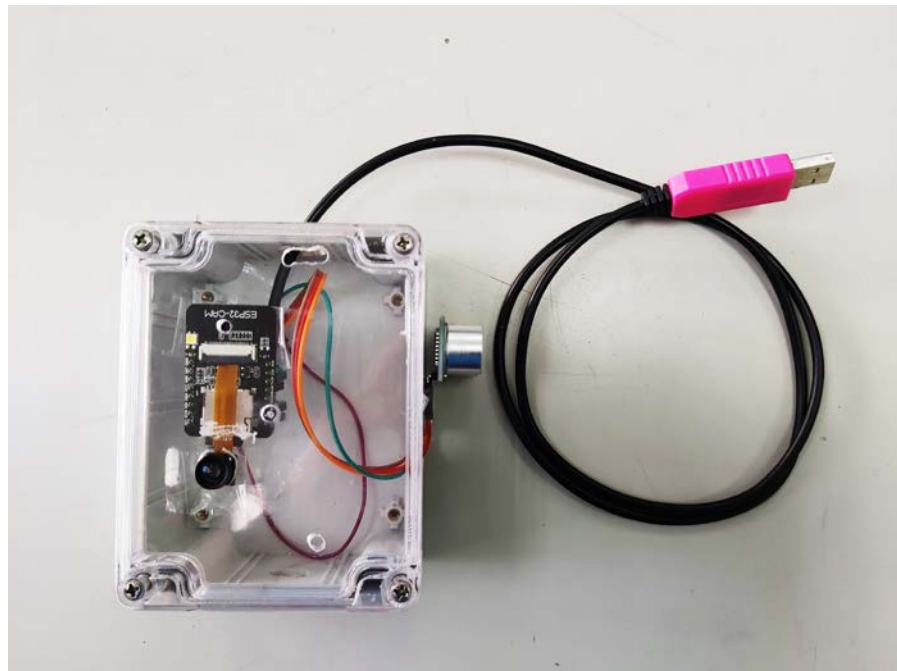


圖 4-1：作品全貌

資料來源：研究者自行拍攝

### 二、Arduino IDE 程式：

```
// Enter your WiFi ssid and password
const char* ssid      = "SSID";    //your network SSID
const char* password = "PW";      //your network password
String myLineNotifyToken = "LINE Token";    //Line Notify
```

Token , You can refer this post to get Line token : <https://t.ly/LZwKn>

```
int trigPin = 15;           //請將 Trig 接 GPIO15
int echoPin = 14;          //Echo Pin 接 GPIO14
```

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"

#define CAMERA_MODEL_AI_THINKER

#define PWDN_GPIO_NUM      32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM      26
#define SIOC_GPIO_NUM      27
#define Y9_GPIO_NUM        35
#define Y8_GPIO_NUM        34
#define Y7_GPIO_NUM        39
#define Y6_GPIO_NUM        36
#define Y5_GPIO_NUM        21
#define Y4_GPIO_NUM        19
#define Y3_GPIO_NUM        18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM     25
#define HREF_GPIO_NUM      23
#define PCLK_GPIO_NUM      22
```

```
String sendImage2LineNotify(String msg) {
    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get();//取得相機影像放置 fb
    if (!fb) {
        delay(100);
        Serial.println("Camera capture failed, Reset");
        ESP.restart();
    }
    WiFiClientSecure client_tcp;//啟動 SSL wificlient
    Serial.println("Connect to notify-api.line.me");
    if (client_tcp.connect("notify-api.line.me", 443)) {
```

```

Serial.println("Connection successful");
String head = "--Taiwan\r\nContent-Disposition: form-data;
name=\"message\"; \r\n\r\n" + msg +
"\r\n--Taiwan\r\nContent-Disposition: form-data; name=\"imageFile\";
filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
String tail = "\r\n--Taiwan--\r\n";
uint16_t imageLen = fb->len;
uint16_t extraLen = head.length() + tail.length();
uint16_t totalLen = imageLen + extraLen;
//開始 POST 傳送訊息
client_tcp.println("POST /api/notify HTTP/1.1");
client_tcp.println("Connection: close");
client_tcp.println("Host: notify-api.line.me");
client_tcp.println("Authorization: Bearer " +
myLineNotifyToken);
client_tcp.println("Content-Length: " + String(totalLen));
client_tcp.println("Content-Type: multipart/form-data;
boundary=Taiwan");
client_tcp.println();
client_tcp.print(head);
uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
Serial.println("Data Sending....");
//檔案太大，分段傳送
for (size_t n = 0; n < fbLen; n = n + 2048) {
    if (n + 2048 < fbLen) {
        client_tcp.write(fbBuf, 2048);
        fbBuf += 2048;
    } else if (fbLen % 2048 > 0) {
        size_t remainder = fbLen % 2048;
        client_tcp.write(fbBuf, remainder);
    }
}
client_tcp.print(tail);
client_tcp.println();
String getResponse = "", Feedback = "";
boolean state = false;
int waitTime = 3000;    // 最多等 3 秒

```

```

long startTime = millis();
delay(1000);
Serial.print("Get Response");
while ((startTime + waitTime) > millis())    {
    Serial.print(".");
    delay(100);
    bool jobdone=false;
    while (client_tcp.available())//當有收到回覆資料時
    {
        jobdone=true;
        char c = client_tcp.read();
        if (c == '\n')
        {
            if (getResponse.length() == 0) state = true;
            getResponse = "";
        }
        else if (c != '\r')
            getResponse += String(c);
        if (state == true) Feedback += String(c);
        startTime = millis();
    }
    if (jobdone) break;
}
client_tcp.stop();
esp_camera_fb_return(fb);//清除緩衝區
return Feedback;
}
else {
    esp_camera_fb_return(fb);
    return "Send failed.";
}
}

void setup() {
    Serial.begin(115200);
    //初始化相機結束
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;

```

```

config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
config.jpeg_quality = 10; //10-63 lower number means higher
quality
config.fb_count = 2;
config.frame_size = FRAMESIZE_XGA;// FRAMESIZE_ +
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
//Line notify don't accept bigger than XGA
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
}
//初始化相機結束，開始網路連線
WiFi.mode(WIFI_STA);
Serial.println("");
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
long int StartTime = millis();

```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    if ((StartTime + 10000) < millis()) break;
}

Serial.println("");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.println("");
if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Reset");
    delay(1000);
    ESP.restart();//連線不成功，則重新開機
}
pinMode(trigPin, OUTPUT);          //Define inputs and outputs
pinMode(echoPin, INPUT);

}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);    // 給 Trig 高電位，持續 10 微秒
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long cm = pulseIn(echoPin, HIGH); // 收到高電位時的時間
    cm = (cm / 2) / 29.1;           // 將時間換算成距離 cm 或 inch
    Serial.println(cm);
    if (cm <= 30) {
        Serial.println("starting to Line");
        String payload = sendImage2LineNotify("There is someone
coming...."); //訊息
        Serial.println(payload);
        delay(10000);
    }
    delay(1000); //You could only send up to 50 images to Line Notify
in one hour.
}

```

### 三、序列埠監控畫面：

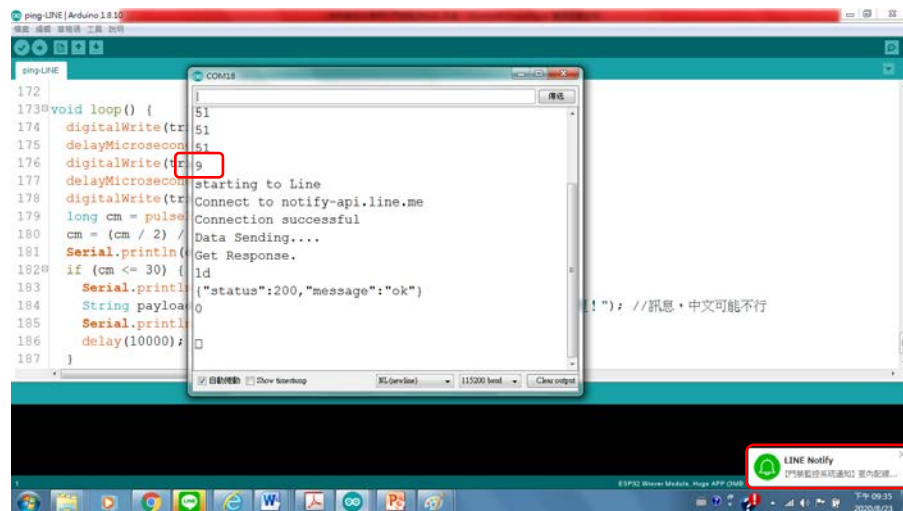


圖 4-2：序列埠監控視窗畫面

資料來源：研究者自行繪製

### 四、LINE Notify 傳送文字訊息及圖片畫面：

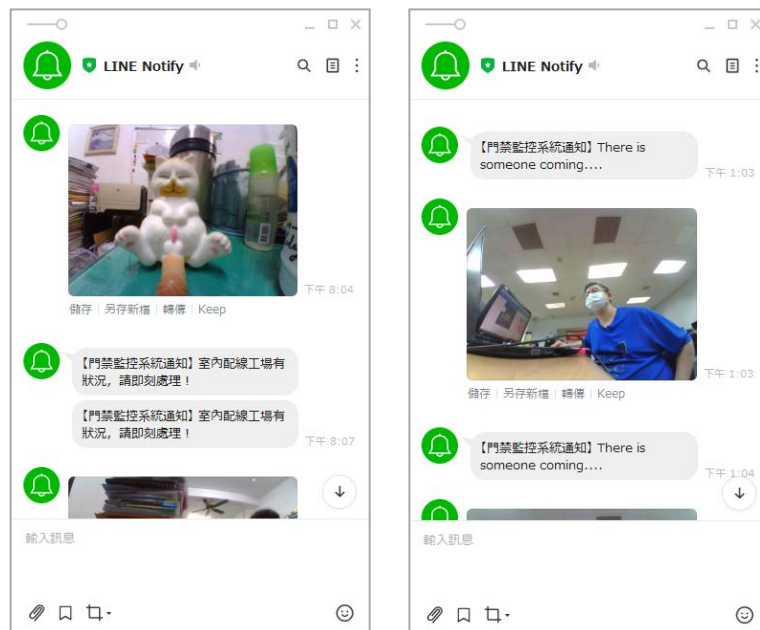


圖 4-3：LINE Notify 傳訊文字訊息及圖片畫面

資料來源：研究者自行繪製

## 伍、研究結論與建議

本研究確實可以達到在有人經過時，即時拍照並發送文字及圖片訊息到我們的 Line 上來通知我們有陌生人靠近，真正的做到以「ESP32-CAM」為基礎的居家安全監控系統，作品可架設在家中任何的位置，並持續 24 小時不斷監控。

未來，作品會不斷改良，並發展成精準度較高的人臉辨識門禁監控系統；未來作品完成後，不僅可以用於確認使用者身分，同時也包括了人臉辨識分析，像是性別、年齡、表情與視線追蹤，甚至也能用於提升與人相關的智慧應用。

## 陸、參考文獻

施士文著(2018 年 8 月)。ARDUINO 微電腦應用實習。台科大圖書，台北市。

董大偉編著(2019 年 9 月)。LINE Bot 與人工智慧辨識開發實戰 增訂版 | 使用 C#與 Azure Cognitive Services。碁峰資訊股份有限公司。

黃信惠的瘋狂教室。<https://www.davidhuanglab.com/post/hc-sr04>。

人臉辨識簡介。<http://image.cse.nsysu.edu.tw/research/Delaunay/delaunay.htm>。

傑森創工。<https://blog.jmaker.com.tw/sr04/>。