



# 物聯網智造基地

I O T S E R V I C E H U B

## 國產IC開發套件- HUB8735 示範案例 - AI 電子貓眼

指導單位：  經濟部產業發展署  
Industrial Development Administration 經濟部產業發展署  
Ministry of Economic Affairs

執行單位：財團法人資訊工業策進會



# 大綱

- 功能說明
- FaceRecognition
- 功能結合
- 成果



# 功能說明

透過 HUB8735 的人臉辨識技術，可讓電子貓眼引入更高層次的安全性及便利性。這次範例透過 瑞昱半導體 基本的人臉辨識功能，能夠快速而準確的識別家庭成員或是訪客。只要在HUB8735上設定指定成員的人臉，不需事先訓練過程。

當門口出現可辨識人臉時，門內的TFT LCD 即自動開啟顯示，即可知道誰在門外等候。若是陌生人，則會自動發出 Line訊息 做為告警。



# FaceRecognition

HUB8735 啟動後，在 Serial monitor 的訊息裡，找出這樣的提示訊息

```
Total number of faces detected = 1
Face 0 name Aniki:      738 956 241 546
SCRFD tick[0] = 35
MBFACENET tick[0] = 25
Network URL for RTSP Streaming: rtsp://192.168.0.61:554
```



開啟 VLC，在“媒體/開啟網路串流”選項。將 `rtsp://192.168.0.61:554` 填入其中



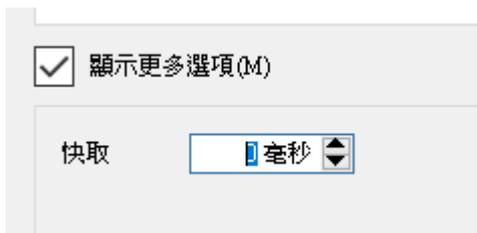
192.168.0.61 是 HUB8735 的IP 位置

RTSP 的埠口號是 554



# FaceRecognition

接著底下有個“顯示更多選項”，將快取設定為 0 毫秒





# FaceRecognition

將 HUB8735 的鏡頭 對準人像



因為還沒將人像註冊，所以會顯示 **unknow**  
並呈現 **紅框**



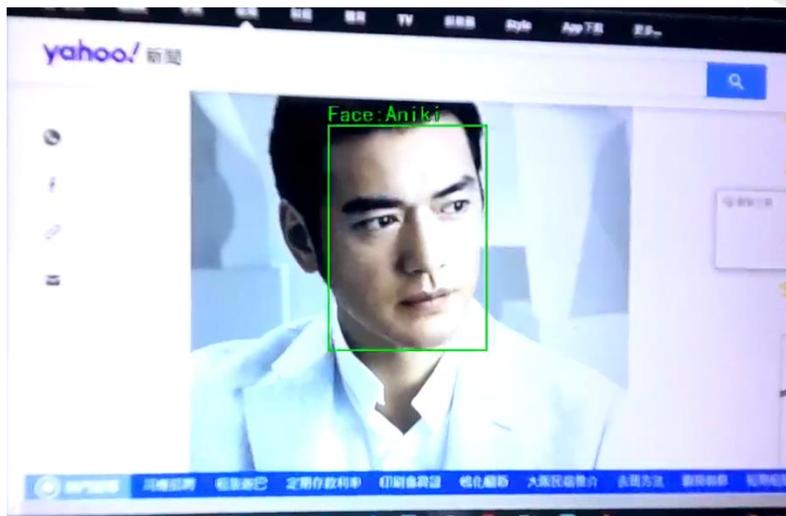
# FaceRecognition

在 Serial minitor 的輸入欄位填入 REG=xxx，例如：REG=Aniki

```
Output Serial Monitor x
REG=Aniki
SCRFD tick[0] = 35
SCRFD tick[0] = 35
SCRFD tick[0] = 35
```

# FaceRecognition

此時，HUB8735 會將 “unknow” 轉變成輸入值 “Aniki”，紅框也轉變成綠框了



注意: 註冊人像時，只能有一個人臉。辨識情境時則無人臉數量限制。



# FaceRecognition - Arduino

人臉辨識需要在 Arduino 程式，宣告引入以下 include file

```
#include "NNFaceDetectionRecognition.h"
```

才會有人臉識別及人臉註冊功能



# FaceRecognition - Arduino

透過 Serial 接收命令，有 REG (註冊)，EXIT(退出註冊模式)，RESET(清除所有人臉註冊紀錄)，BACKUP(備份人臉註冊紀錄)，RESTORE(回復註冊紀錄)

```
if (Serial.available() > 0) {
  String input = Serial.readString(); //透過 Serial 等待輸入
  input.trim();

  if (input.startsWith(String("REG="))){ //註冊名字。記憶辨識結果，註冊人臉時，請確保畫面中只有一張人臉。
    String name = input.substring(4);
    facerecog.registerFace(name);
  } else if (input.startsWith(String("EXIT"))){ //輸入命令"EXIT"退出註冊模式。
    facerecog.exitRegisterMode();
  } else if (input.startsWith(String("RESET"))){ //輸入命令"RESET"以忘記所有先前註冊的臉部。所有先前分配的面孔和名稱都將被刪除。
    facerecog.resetRegisteredFace();
  } else if (input.startsWith(String("BACKUP"))){ //輸入命令"BACKUP"將已註冊的面孔的副本保存到閃存。如果存在備份，請輸入命令"RESTORE"以從閃存加載已註冊的面孔。
    facerecog.backupRegisteredFace();
  } else if (input.startsWith(String("RESTORE"))){
    facerecog.restoreRegisteredFace();
  }
}
```



# FaceRecognition

其他命令

**BACKUP**

會將已註冊好的人臉儲存至閃存。備份之後，可以使用 **RESTORE** 命令來回復載入備份好的人臉。

**RESET**

清除註冊好的人臉。之前註冊好的人臉都將被清除

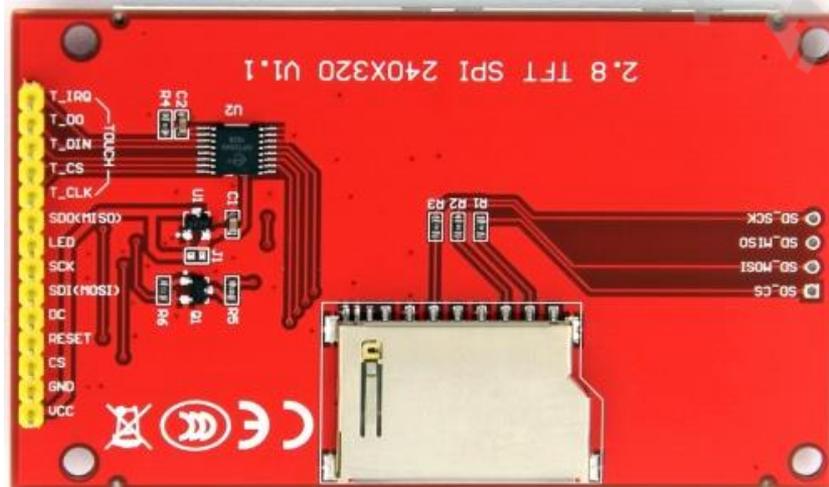
**EXIT**

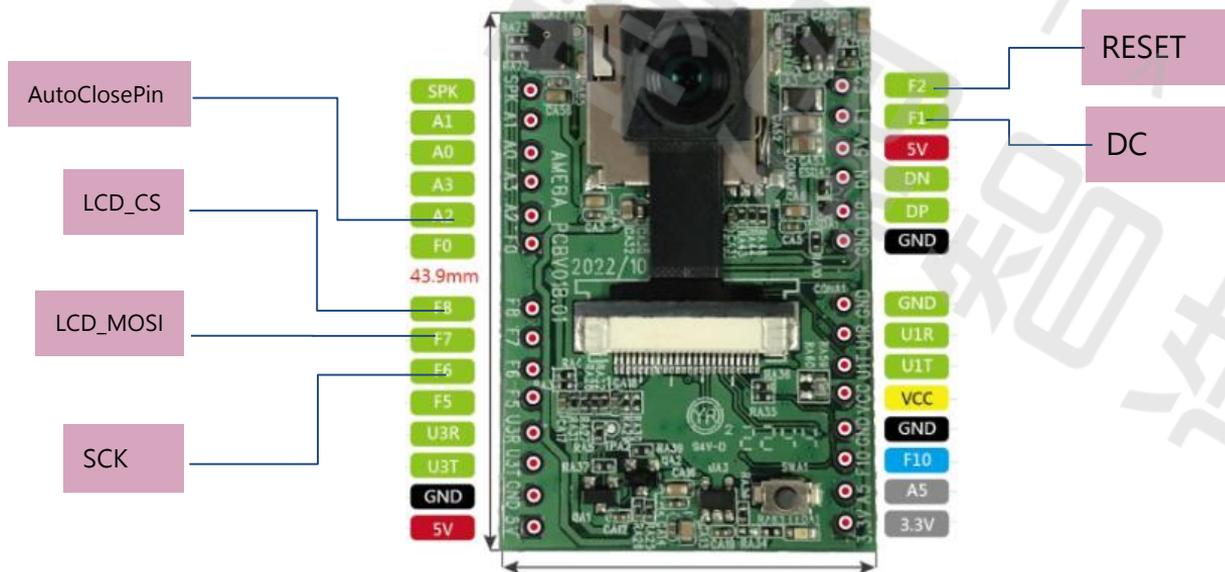
退出註冊模式。

# TFT LCD 顯示

電子貓眼的顯示功能極其重要，可以讓使用方輕鬆檢視門外狀態。免除傳統的小孔的不方便。

這個功能範例使用的 TFT 為 ili9341 LCD 驅動晶片。







# TFT LCD 顯示

**AutoClosePin** 為自動開啟/關閉 LCD 背光，當有辨識到人臉時會自動啟動，並停留 9 秒。

Arduino 程式 - 設定

在 setup() 內增加 Video setting 及新增 Video Channel. 做為 TFT 及 儲存專用設定

```
VideoSetting configTFT(VIDEO_VGA,CAM_FPS,VIDEO_JPEG,1);  
Camera.configVideoChannel(CHANNELTFT, configTFT);
```



# TFT LCD 顯示

其中

```
Camera.getImage(CHANNELTFT,&img_addr,&img_len);
```

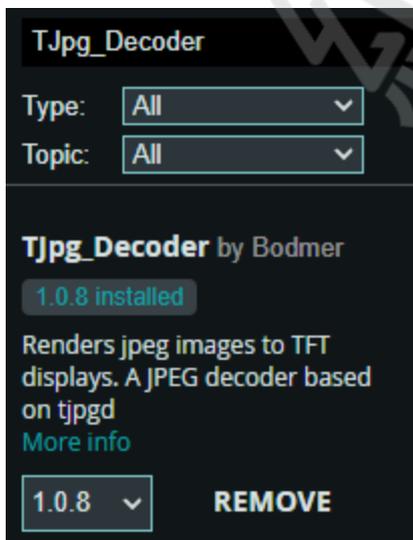
為抓取 image raw data ， **img\_addr**:資料開頭位址 ， **img\_len**:資料長度

透過 **img\_addr & img\_len** 可分別為 **SD Card** 儲存 及 **TFT LCD** 顯示



# TFT LCD 顯示

顯示 Camer to TFT LCD 功能，需先安裝 **TJpg\_Decoder Library** 並在程式宣告引入 `#include "TJpg_Decoder.h"`





# TFT LCD 顯示

首先請修改 User\_Config.h

%Your\_Project%\libraries\Tjpg\_Decoder\src\User\_Config.h

將 “#define TjpgD\_LOAD\_SD\_LIBRARY” 這一行註解

```
#if defined (ESP32) || defined (ESP8266) || (ARDUINO_ARCH_RP2040) || defined (ARDUINO_ARCH_MBED)
  #define TjpgD_LOAD_FFS
#endif

//#define TjpgD_LOAD_SD_LIBRARY
```



# TFT LCD 顯示

在 Setup()

```
TJpgDec.setJpgScale(2);
```

```
TJpgDec.setCallback(tft_output);
```

新增 tft\_output callback function

```
bool tft_output(int16_t x,int16_t y, uint16_t w, uint16_t h, uint16_t * bitmap)
{
    tft.drawBitmap(x,y,w,h,bitmap);
    return 1;
}
```



# TFT LCD 顯示

## 取得 影像 raw data

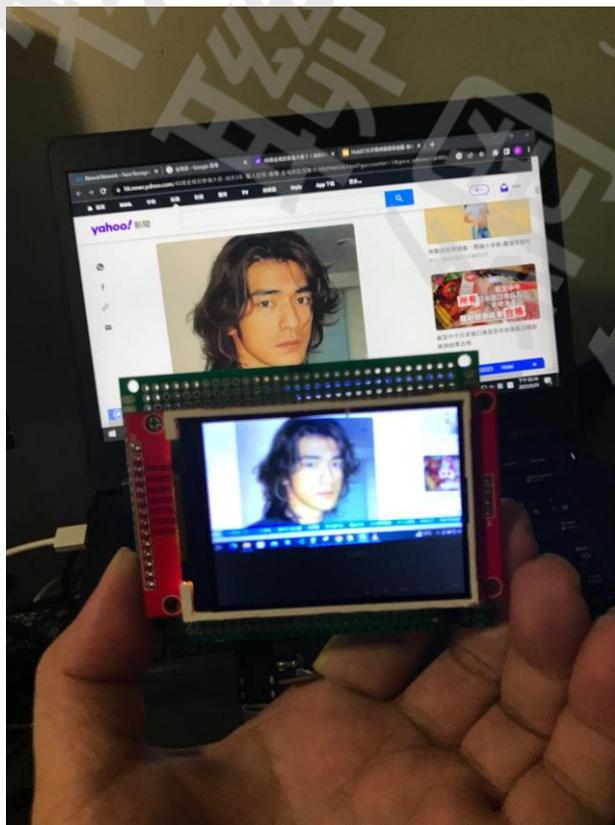
```
Camera.getImage(CHANNELTFT,&img_addr,&img_len);
```

透過 Tjpg\_Decode Library ，當有畫面被Decode 時程式會 callback 到 tft\_out funciton ，將鏡頭畫面呈現在 TFT LCD

```
TjpgDec.getJpgSize(0,0,(uint8_t *)img_addr,img_len);  
TjpgDec.drawJpg(0,0,(uint8_t *)img_addr,img_len);
```

# TFT LCD 顯示

呈現結果





# SD Card 儲存

與 TFT LCD 共用 img\_addr & img\_len 變數

```
fs.begin();  
File file = fs.open(String(fs.getRootPath()) + String(FILENAME) + String("picturecount") + String(".jpg"));  
  
file.write((uint8_t*)img_addr, img_len);  
delay(1);  
file.close();  
fs.end();
```



# LINE Notify

目的: 若辨識到為陌生人，將透過 Line notify 發出告警訊息。

登入: [https://notify-bot.line.me/zh\\_TW/](https://notify-bot.line.me/zh_TW/)

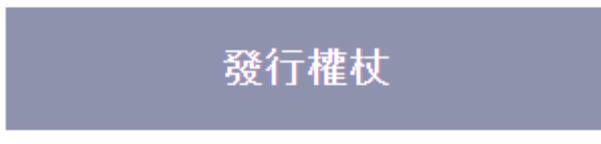
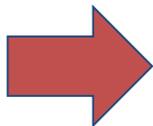
進入個人頁面

點取 發行權狀

個人頁面

管理登錄服務

登出





# LINE Notify

輸入任何名稱

請填寫權杖名稱(將於傳送提醒時顯示)

最多只能輸入20個字。

選擇 1對1

透過1對1聊天接收LINE Notify的通知

點取 發行

發行



# LINE Notify

## 複製 權杖

已發行的權杖如下。

d4I3os8Ro

若離開此頁面，將不會再顯示新發行的權杖。離開頁面前，請先複製權杖。

複製

關閉



# LINE Notify

Arduino code

將權狀填入 LineToken

```
String LineToken = "" ;
```

當偵測到陌生人，將執行  
送出 Message

```
//Line notify  
char server[] = "notify-api.line.me";  
String LineToken = "";
```

```
if(client.connect(server,443))  
{  
  Serial.println("connected to server");  
  // 使用 HTTP request:  
  message = "有陌生人在門口停留!!!";  
  String query = "message=" + message;  
  client.print("POST /api/notify HTTP/1.1\r\n");  
  client.print("Host: " + String(server) + "\r\n");  
  client.print("Authorization: Bearer " + LineToken + "\r\n");  
  client.print("Content-Type: application/x-www-form-urlencoded\r\n");  
  client.print("Content-Length: " + String(query.length()) + "\r\n");  
  client.print("\r\n");  
  
  client.print(query + "\r\n");  
}
```



# LINE Notify





Thank you



物聯網智造基地

IOT SERVICE HUB