

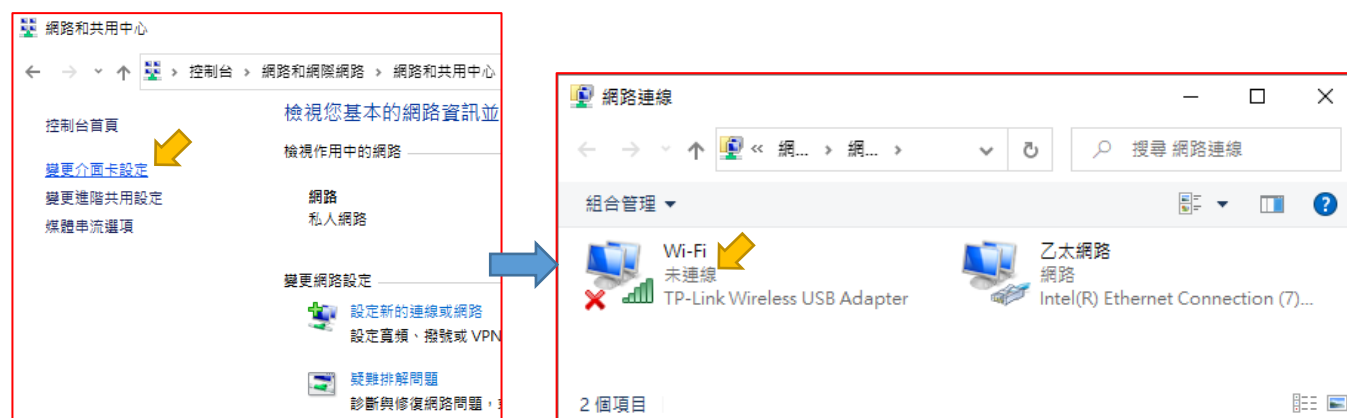
1. 由於 ESP32CAM 需與電腦在相同網域方可控制，所以桌機需要先安裝無線網路環境，首先插上小小顆的無線網卡，點擊驅動程式資料夾 **Autorun.exe** 自動執行，點選網卡圖示安裝，但筆電則不需要。




2. 左下角放大鏡輸入『控』，點選『控制台』→右上檢視方式選『類別』→『檢視網路狀態及工作』

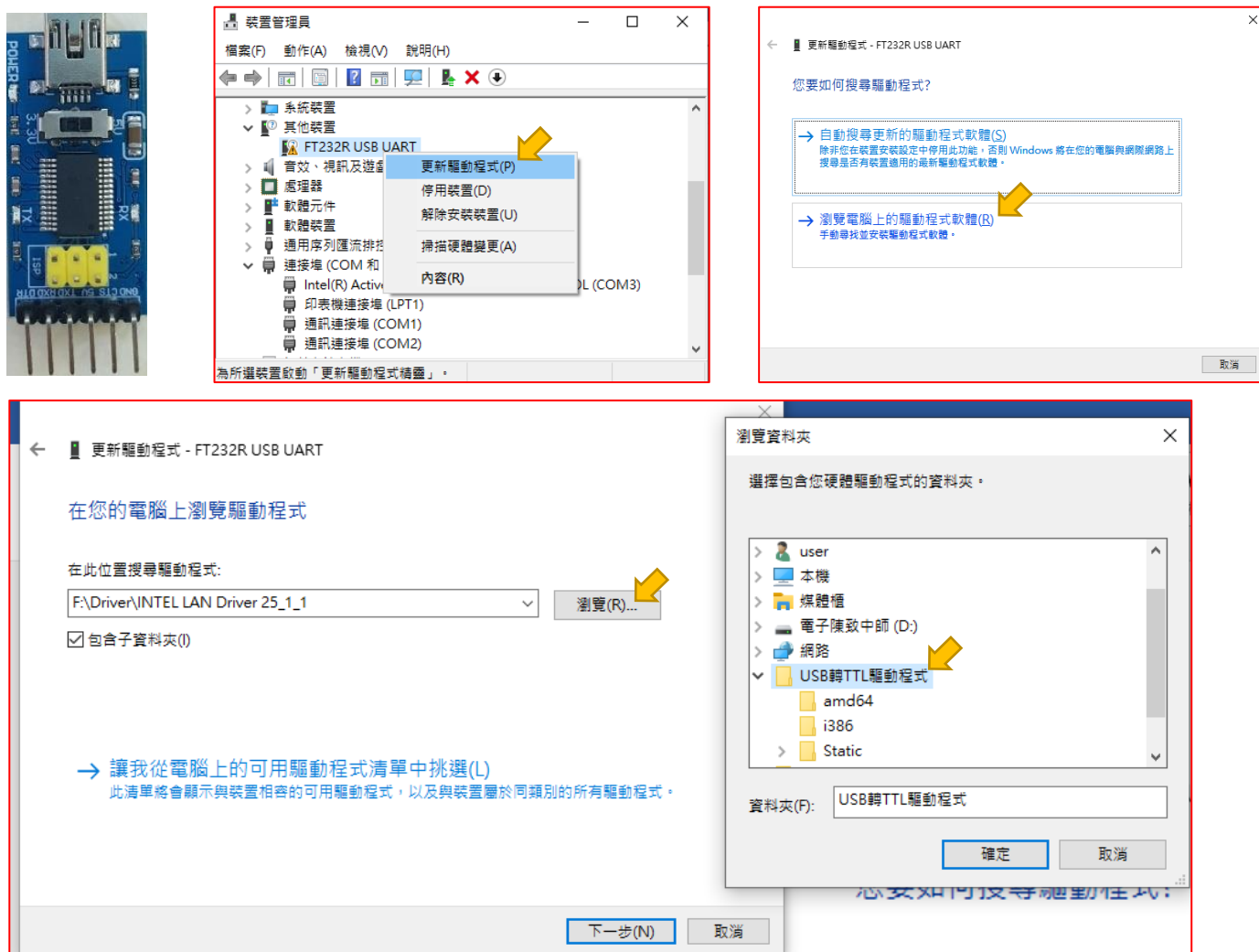


3. 左側『變更介面卡設定』→點選無線『Wi-Fi』，乙太網路是指有線的網路。

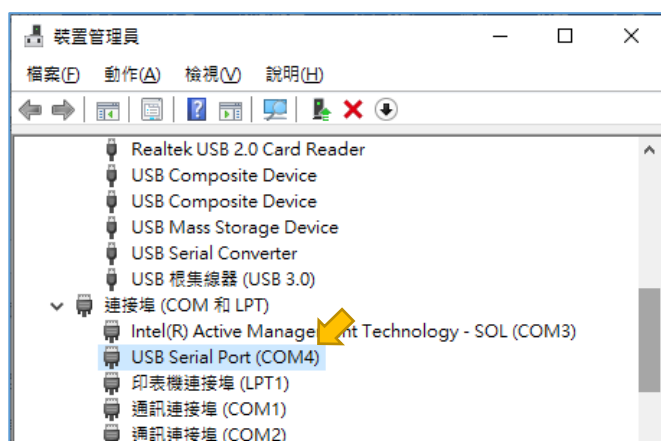


4. 選擇無線網路 SSID，輸入密碼之後，若右下角可看到  圖示表示連線成功。由於桌機的無線網卡與 ESP32 CAM 都要無線上網，若全班都同時使用學校無線 AP，當多人同時使用，極易發生網路塞車，所以**建議學生自行用手機當作無線基地台分享給自己**，若真無網路吃到飽，學校 SSID 為 E3-7，密碼為 520...521，但網路頻寬有限，敬請進行網路的分流。

5. 由於 ESP32-CAM 並無燒錄介面，需要準備一個『USB 轉 TTL』工具，連接方式是兩邊的 Tx/D 與 Rx/D 互接，先插上該裝置，在桌面『本機』圖示按右鍵，選擇『內容』→『裝置管理員』→在 FT232RL USB UART 裝置按右鍵，選擇『更新驅動程式』→『瀏覽電腦上的驅動程式軟體』→選擇 USB 轉 TTL 驅動程式資料夾→確定→下一步，但需要重複做 2 次才會成功。



6. 當在連接埠(COM 和 LPT)可看到新增的『USB Serial Port(COM?)』，即表示驅動程式安裝成功。



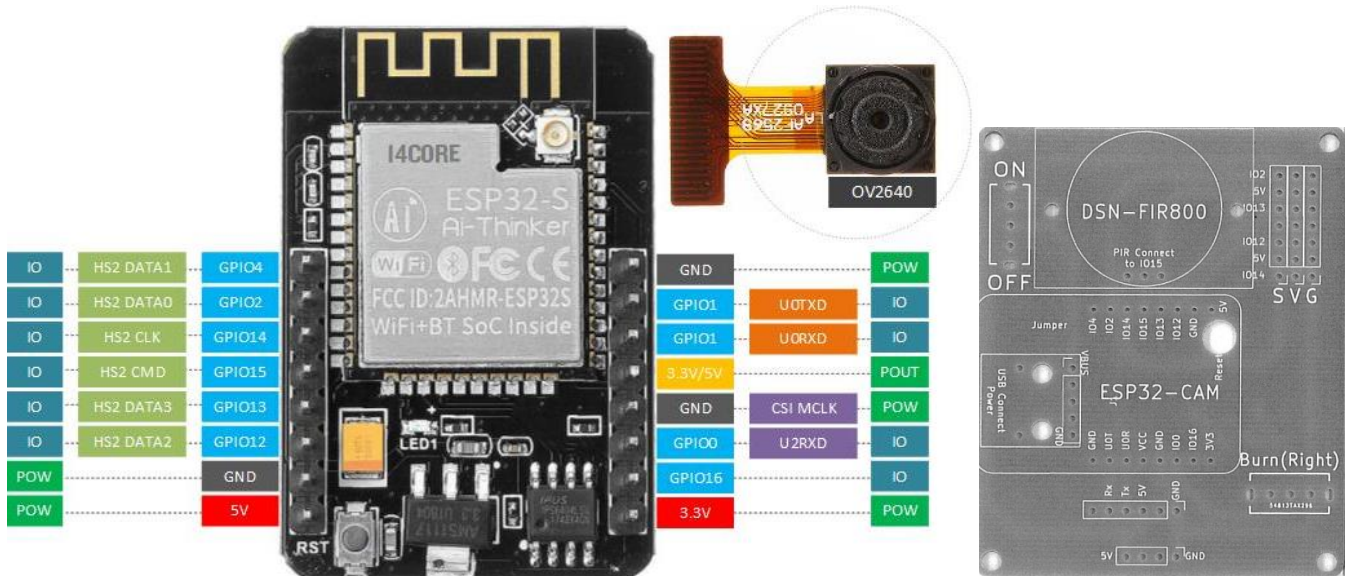
※ 在 <https://www.youtube.com> 自行搜尋『ESP32 CAM』關鍵字，網路有許多創作經驗的影片分享。



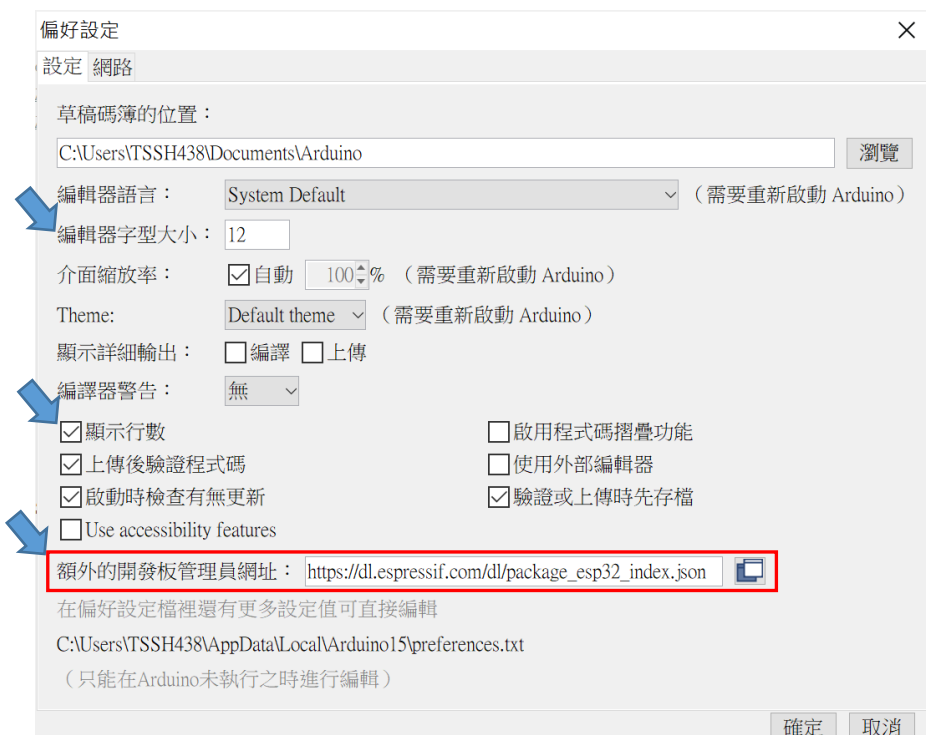
## 安可信 ESP32-CAM 規格介紹

ESP32-CAM 內建 Wi-Fi，並結合 OV2640 影像感測器模組(解析度 1632 x 1232)、PSRAM 記憶體、MicroSD 記憶卡槽(存照片，最大 32G)，ESP32-CAM 常見應用：縮時攝影、PIR 防盜監視器、一般監視器、人臉辨識門禁系統、視訊遙控車、影像辨識自走車、做為網路攝影機應用、其他各式影像辨識應用使其適合，同時與過往 ESP8266 一樣相容於 Arduino。

重要的是，只要透過 Arduino 的一個內建範例程式 CameraWebServer，將該程式透過 PC 燒寫到 ESP32 CAM，就可以在 ESP32 CAM 上建立一個網頁伺服器 (Web Server)，而後只要用瀏覽器瀏覽 192.168.x.x 內部網址，就可以看到攝影機角度影像內容，進一步也可以作一些影像辨識應用，如人臉位置標記等，這等於也是一個 WoT (Web of Things)，ESP32 CAM 的 **GPIO4 已經接閃光燈**(超亮，持續點亮太久會冒煙)、**GPIO0 是燒錄用**(燒錄接 GND，平時不可接 GND)、GPIO1 是當 TxD、GPIO3 是當 RxD，Vcc 不是外接電源而是提供電源(POUT)，外接電源(POW)標示 5V，外觀與腳位如下。



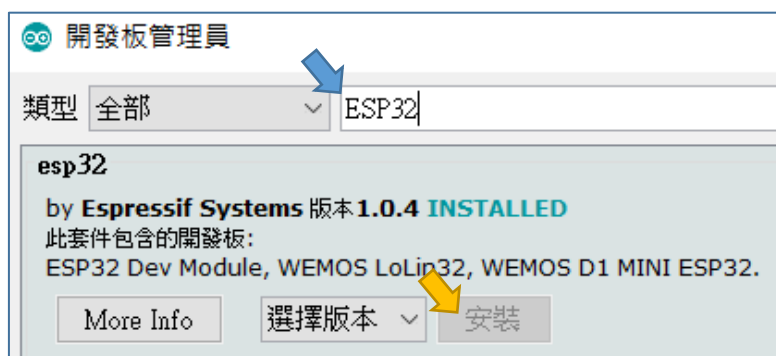
1. 請至 <https://www.arduino.cc/en/Main/Software> 網站下載最新開發套件 (註：ESP32 CAM 需要最新的核心程式，建議下載 1.8.12 以上版本，選擇 Windows 版本→**JUST DOWNLOAD** 只是下載)。
2. 點擊桌面『新版 Arduino IDE』捷徑後，選擇『檔案/偏好設定』，出現偏好設定視窗，在『額外的開發板管理員網址』輸入『[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)』，按『確定』。



右下開關往右是 **Burn** 燒錄，往左是執行模式，左上則是外接電源開關

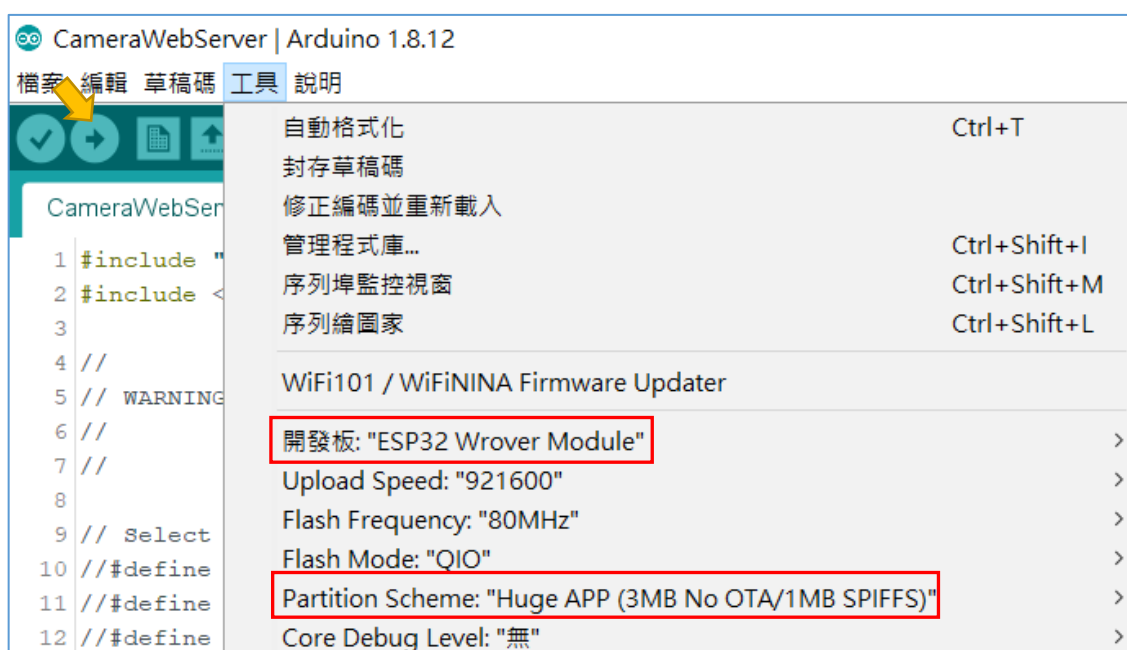


3. 在『工具→開發板→開發板管理員』中輸入『ESP32』關鍵字，經過一段時間找到 ESP32 套件，按『安裝』1.0.4 最新版，否則編譯會出現錯誤，若無法找到是因為 Arduino 版本太舊，請更新最新版。



(※檔案較大，請使用有線網路安裝)

4. 最後點選『工具→開發板』，選擇『ESP32 Arduino → ESP32 Wrover Module』板子。由於程式包含串流 stream server 及臉部辨識，因此檔案較大，需將燒錄區域方案 Partition Scheme 改成『Huge APP(3MB No OTA/1MB SPIFFS)』，否則編譯會出現錯誤。



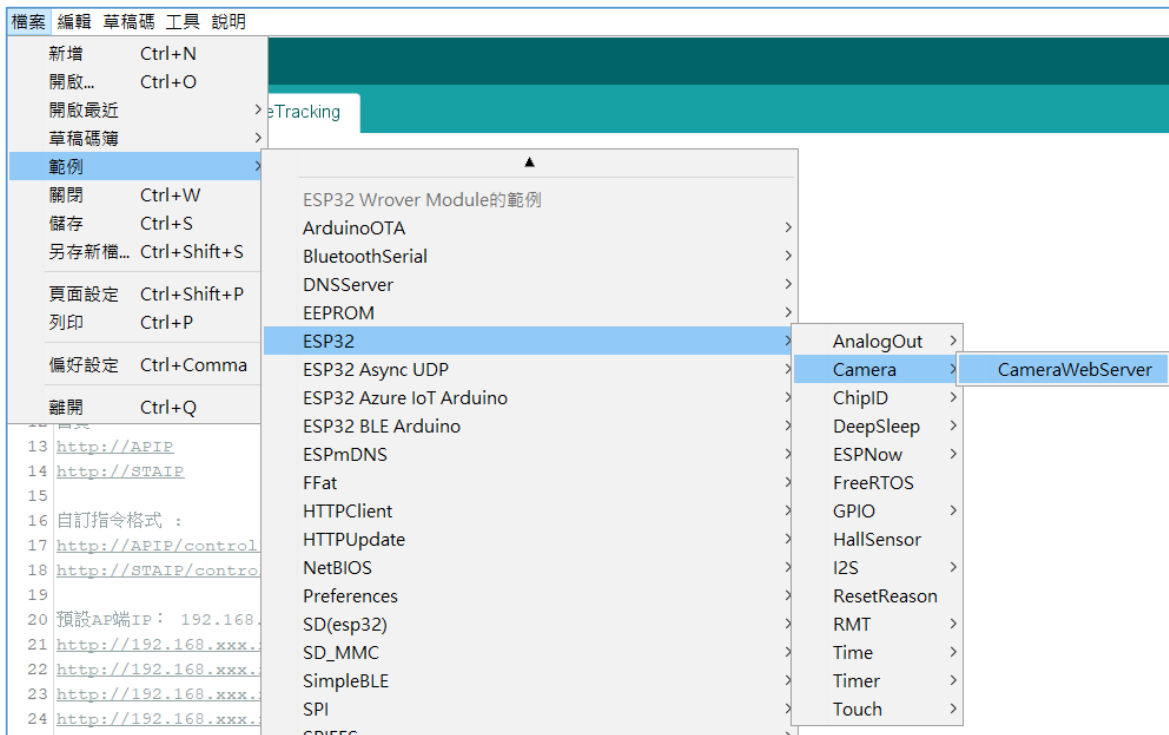
5. 如果在上傳程式時出現類似 "找不到序列埠訊息..."，建議您更換一個好一點的 USB to TTL 模組。
6. 因為 ESP32 CAM 開發板需要較穩定的電壓和較多的電流，如果您發現板子出現不穩定的狀態，或出現 brownout 的錯誤時，代表需更換較粗(好)的電源線線材或者 USB 接口（建議 USB3.0）。
7. 可以上傳程式不表示 ESP32 CAM 開發板沒有問題，因為有些板子會出現無法連接到 IP 分享器的狀況，此時序列埠會一直出現連線狀態，如下圖

```
rst:0x3 (SW_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8
```

8. 若沒安裝好鏡頭或產品本身的問題，影像可能會出現顏色異常的現象(例如黃色很淡或顯示不出來)，所以初次使用 ESP32 CAM，建議拿出各種顏色的物品讓鏡頭拍攝，仔細檢查一下顏色是否異常。

## 實作 01：利用官方 CameraWebServer 範例，進行影像追蹤與辨識

- 依序按『檔案 → 範例 → ESP32 → Camera 相機 → CameraWebServer 相機網頁伺服器』，若找不到該範例，那是因為還未選擇『ESP32 Wrover Module』板。



- 將第 10 行變註解，取消第 14 行註解，因為本開發版是 THINKER，並修改第 18 與 19 行帳號密碼。

```

9 // Select camera model
10 #define CAMERA_MODEL_WROVER_KIT
11 // #define CAMERA_MODEL_ESP_EYE
12 // #define CAMERA_MODEL_M5STACK_PSRAM
13 // #define CAMERA_MODEL_M5STACK_WIDE
14 // #define CAMERA_MODEL_AI_THINKER
15
16 #include "camera_pins.h"
17
18 const char* ssid = "*****";
19 const char* password = "*****";
    
```

```

9 // Select camera model
10 // #define CAMERA_MODEL_WROVER_KIT
11 // #define CAMERA_MODEL_ESP_EYE
12 // #define CAMERA_MODEL_M5STACK_PSRAM
13 // #define CAMERA_MODEL_M5STACK_WIDE
14 #define CAMERA_MODEL_AI_THINKER
15
16 #include "camera_pins.h"
17
18 const char* ssid = "TSSH";
19 const char* password = "12345678";
    
```

- 當燒錄時要將 GPIO0 與 GND 連接，燒錄完成後，將 GPIO0 與 GND 分開，則在界面上一直持續出現 **Connecting...** 時候，此時需立即按一下 **RESET 重置按鈕**，閃光燈會亮一下，然後就會開始燒錄，若沒看到錯誤提示，則千萬不要按 RESET，若燒錄時看到『**A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header**』錯誤訊息，表示要再用一次。

```

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

草稿碼使用了 2259298 bytes (71%) 的程式儲存空間。上限為 3145728 bytes。
全域變數使用了 52712 bytes (16%) 的動態記憶體，剩餘 274968 bytes 給區域變數。上限為 327680 bytes。
esptool.py v2.6
Serial port COM2
Connecting.....

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
    
```

4. 燒錄過程會依序出現 1% 到 100% 進度，若較大程式，時間大約要一分鐘，當燒錄成功上方標題會出現『上傳完畢』，如右圖所示。


```

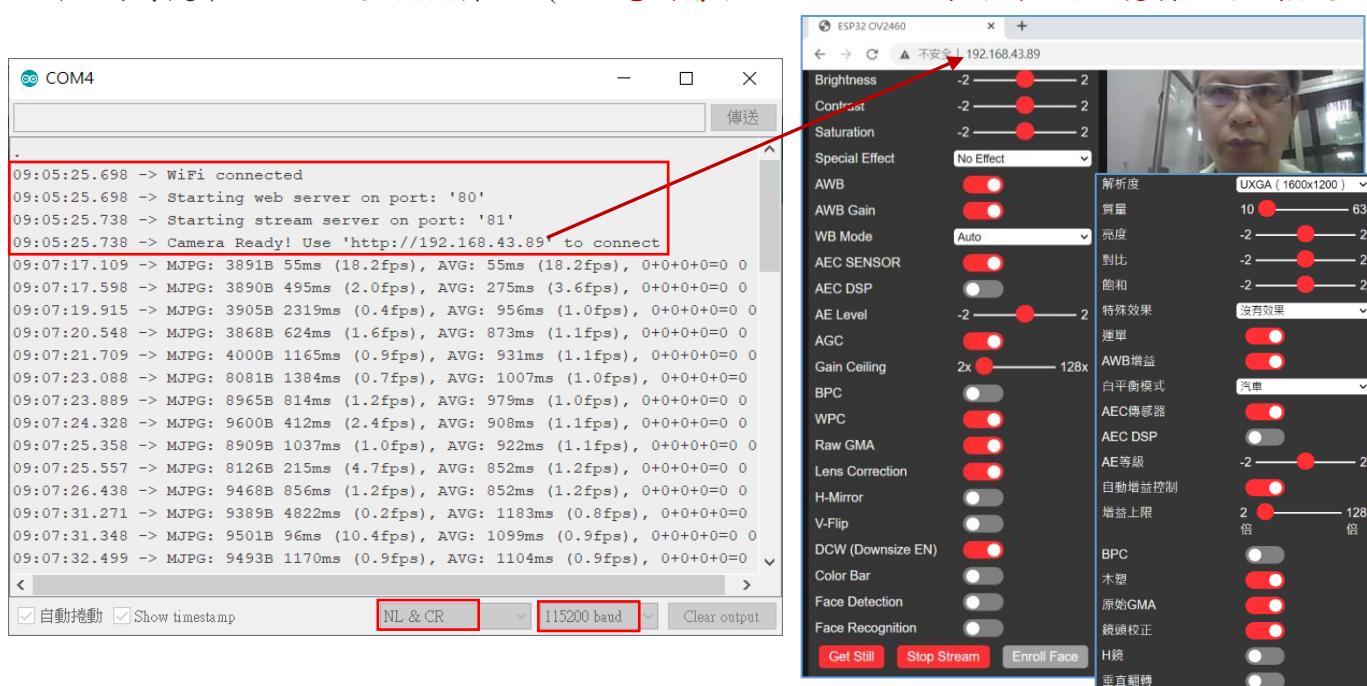
上傳完畢。
Writing at 0x001b0000... (95 %)
Writing at 0x001b4000... (96 %)
Writing at 0x001b8000... (97 %)
Writing at 0x001bc000... (98 %)
Writing at 0x001c0000... (99 %)
Writing at 0x001c4000... (100 %)
Wrote 2259408 bytes (1794155 compressed) at 0x00010000 in 158.8 seconds (effective 113.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 134...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (134 compressed) at 0x00008000 in 0.0 seconds (effective 768.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

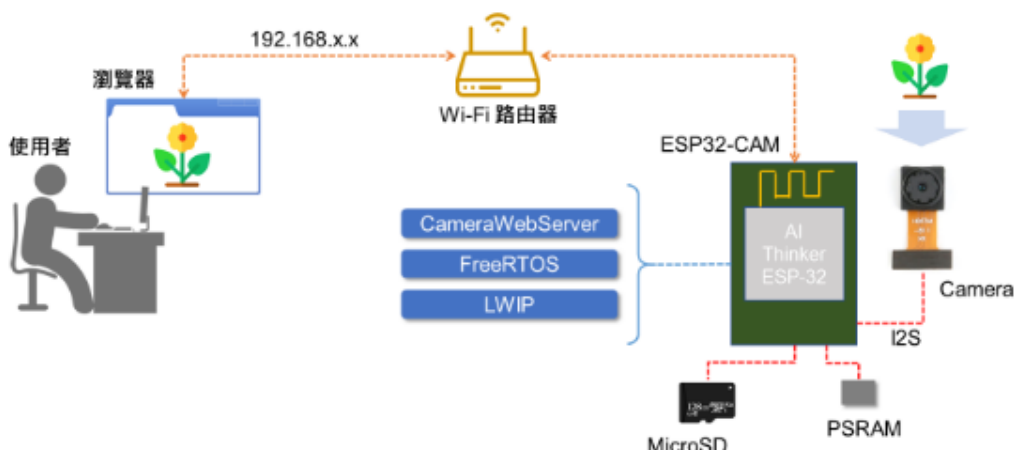
5. Wi-Fi 基地台請先開啟，右下開關往左撥，切到執行模式，按下 Reset 按鈕重新啟動開機或拔除再插入電源，然後按右上方  監視視窗，速度改 115200bps，因為 `24 Serial.begin(115200);`，若成功連接 Wi-Fi 網路會看到分配給 ESP32 CAM 的位置如 `http://192.168.X.X`，複製該位置到 Chrome 瀏覽器，按最下方的『Start Stream 開始串流』即可看到動態影片，『Get Still』是取得靜態照片，同時觀察 Arduino 監視視窗。（註：電腦需跟 ESP32 CAM 同網域方可，鏡頭方向並無統一）



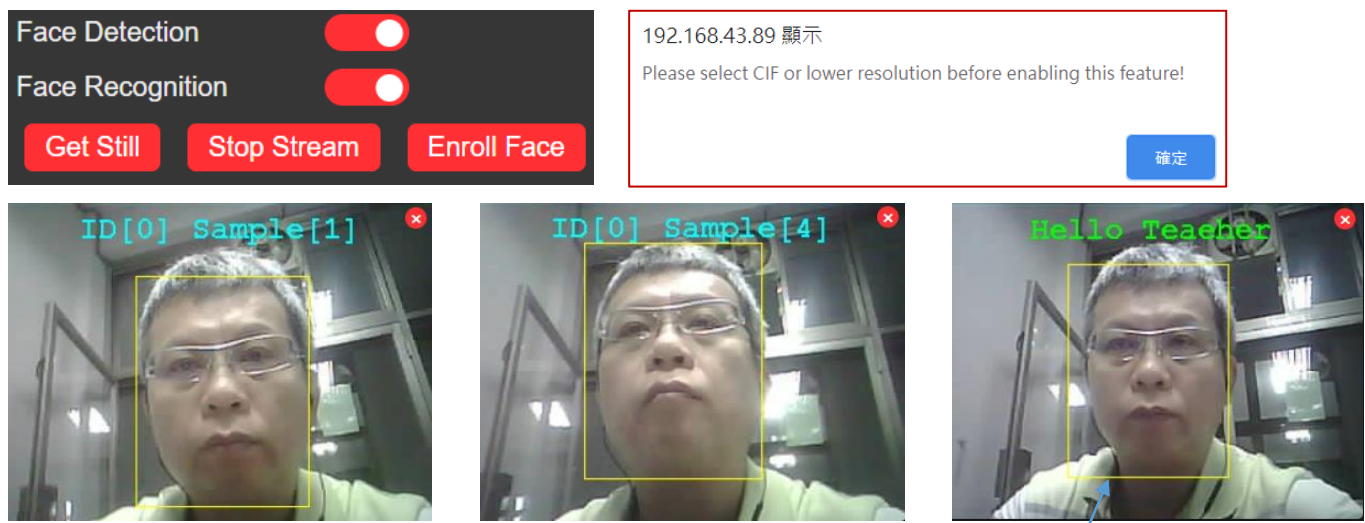
6. 若有連線但看不到影像，按瀏覽器右上『⋮』→更多工具→開發人員工具，以便觀察錯誤訊息。

7. 動態分配給 ESP32

CAM 的位置，若一直沒看到，表示尚未連接到 WiFi，有可能是因為電流供應不穩定，如桌機前面 USB 是透過跳線接出來，電流會較小，不妨改接到桌機背後的 USB，因為焊接在主機板的電流會較穩定。



8. 開啟最下方『Face Recognition 臉部辨識』，則上方『Face Detection 臉部檢測追蹤』自動開啟，此時『Enroll Face 註冊人臉』才可使用，若解析度調太高會出現右下圖的錯誤，請降低解析度，然後程式就會自動追蹤臉部，每出現方框就自動拍照，上方顯示 ID[0] 是表示第 1 位，而 Sample[4] 是表示已經取樣第 5 張照片(編號是從 0 開始)，若看到 ID[1] Sample[5] 是表示第 2 位的第 1 張照片，依此類推，其實當抓完五張照片就可以自動開始辨識，接著可關閉 Face Recognition 臉部辨識，只要剛剛人物出現，就會自動出現人名，若欲修改顯示的名稱，請切換修改『app\_httpd.cpp』檔案的第 191 行內容，若網路速度太慢，會出現右下圖錯誤訊息，提醒選擇更低的解析度(lower resolution)。



192.168.43.89 顯示  
Please select CIF or lower resolution before enabling this feature!  
確定

CameraWebServer - app\_httpd.cpp | Arduino 1.8.12  
檔案 編輯 草稿碼 工具 說明

```
173 if (align_face(net_boxes, image_matrix, aligned_face) == ESP_OK){  
174   if (is_enrolling == 1){  
175     int8_t left_sample_face = enroll_face(&id_list, aligned_face);  
176  
177     if(left_sample_face == (ENROLL_CONFIRM_TIMES - 1)){  
178       Serial.printf("Enrolling Face ID: %d\n", id_list.tail);  
179     }  
180     Serial.printf("Enrolling Face ID: %d sample %d\n", id_list.tail, ENROLL_CONFIRM_TIMES - left_sample_face);  
181     rgb_printf(image_matrix, FACE_COLOR_CYAN, "ID[%u] Sample[%u]", id_list.tail, ENROLL_CONFIRM_TIMES - left_sample_face);  
182     if (left_sample_face == 0){  
183       is_enrolling = 0;  
184       Serial.printf("Enrolled Face ID: %d\n", id_list.tail);  
185     }  
186   } else {  
187     matched_id = recognize_face(&id_list, aligned_face);  
188     if (matched_id >= 0) {  
189       Serial.printf("Match Face ID: %u\n", matched_id);  
190       if (matched_id==0)  
191         rgb_printf(image_matrix, FACE_COLOR_GREEN, "Hello %s", "Teacher");  
192       else  
193         rgb_printf(image_matrix, FACE_COLOR_GREEN, "Hello Person %u", matched_id);  
194     }  
195   }  
196 }
```

```
matched_id = recognize_face(&id_list, aligned_face);  
if (matched_id >= 0) {  
  Serial.printf("Match Face ID: %u\n", matched_id);  
  rgb_printf(image_matrix, FACE_COLOR_GREEN, "Hello Subject %u", matched_id);  
}
```

9. 影像辨識若在光線不足或背光條件時，辨識效果會很差，應盡量避免發生。

※ ESP32-CAM 網路學習資源推薦

- Randomnerdtutorials <https://randomnerdtutorials.com/esp32-cam-troubleshooting-guide/>
- Youtube [https://www.youtube.com/results?search\\_query=esp32-cam](https://www.youtube.com/results?search_query=esp32-cam)
- Github (fustyles) <https://github.com/fustyles/Arduino>
- Github <https://github.com/search?q=esp32-cam>
- 台灣 ESP32 好玩客 <https://www.facebook.com/groups/2468989666672920/>



## 實作 02 : ESP32-CAM\_http.ino 與 ESP32-CAM\_https.ino

1. 整合實作 01 的四個檔案，增加透過網頁的各式控制，可參考程式碼前面註解 1-36 行說明，若瀏覽器畫面當掉，可依序按右上『:』→更多工具→清除瀏覽資料→清除資料。
2. ESP32-CAM\_http.ino 模組可串流，無法使用 HTTPS，所謂 HTTPS 全名超文本傳輸安全協定，S 就是 Secure 安全的意思；HTTPS 一樣透過 HTTP 進行通訊，但通訊過程使用 SSL/TLS 進行加密，當要將影片上傳到雲端就不行，所以必須使用 ESP32-CAM\_https.ino，利用『偽串流』來使用 HTTPS，所謂偽串流又稱為『張流』，即將多張照片持續上傳，但缺點是會造成延遲看起來較嚴重。

```
ESP32-CAM_https
13
14 預設AP端IP: 192.168.4.1
15
16 查詢Client端IP:
17 查詢IP: http://192.168.4.1/?ip
18 重設網路: http://192.168.4.1/?resetwifi=ssid:password
19
20 自訂指令
21 http://192.168.xxx.xxx?ip
22 http://192.168.xxx.xxx?mac
23 http://192.168.xxx.xxx?restart
24 http://192.168.xxx.xxx?digitalwrite=pin;value
25 http://192.168.xxx.xxx?analogwrite=pin;value
26 http://192.168.xxx.xxx?flash=value //value= 0~255 閃光燈
27 http://192.168.xxx.xxx?getstill //取得視訊影像
28 http://192.168.xxx.xxx?framesize=size // value = 10->UXGA(1600x1200), 9->SXGA
29 http://192.168.xxx.xxx?quality=value // value = 10 to 63
30 http://192.168.xxx.xxx?brightness=value // value = -2 to 2
31 http://192.168.xxx.xxx?contrast=value // value = -2 to 2
32 http://192.168.xxx.xxx?hmirror=value // value = 0 or 1
33 http://192.168.xxx.xxx?vflip=value // value = 0 or 1
34 http://192.168.xxx.xxx?telegramimage=token:chatid
35 http://192.168.xxx.xxx?telegramtext=token:chatid;text
36 */
37
38 //輸入WiFi連線帳號密碼
39 const char* ssid = "TSSH"; //Wi-Fi帳號
40 const char* password = "12345678"; //Wi-Fi密碼
41
42 //輸入AP端連線帳號密碼
43 const char* apssid = "ESP32-CAM88"; //可自訂名稱於WiFi熱點清單中顯示名稱與區網IP
44 const char* appassword = "12345678"; //AP密碼至少要8個字元以上
```

當電源不穩定或電流太小，易有條紋波干擾，不妨外加 5V 電源

『數位 Digital』是指 0 與 1 或亮與不亮兩種狀態，屬於『不連續型』。而『類比 Analog』是指 0~255，屬於『連續型』

▲善用複製修改第 36 行以前之註解


3. 第 43 行自訂 ESP32-CAM 的 AP 熱點名稱，末兩碼建議每人用座號代替，方便觀察分配到的動態位置，不然全班都長得一樣，怎知自己是哪片板子？但若是使用自己網路則不用變更第 43 行。
4. 若桌機與 ESP32-CAM 是同個區段網路，則用 192.168.43.89 位置控制，若無對外的網路吃到飽，則將桌機連到第 43 行設定的熱點，即使 ESP32-CAM 沒有設定連上區域網路，也可以利用 192.168.4.1 進行『離線』控制，但無法將資料上傳到雲端。
5. XXX\_http.ino 視訊串流目前『僅一人』能取得影像，自訂指令可多人排隊執行，問號前面需多 control。XXX\_https.ino 視訊張流可『多人排隊』分別取得影像，自訂指令可多人排隊執行。
6. 從 ESP32-CAM\_http.ino 複製第 24 行『http://192.168.xxx.xxx?digitalwrite=pin;value』註解，網址問號後面是參數傳遞，digitalwrite 數位寫入，pin 腳位、value 數值，若是數位則為 0 或 1，若是類比

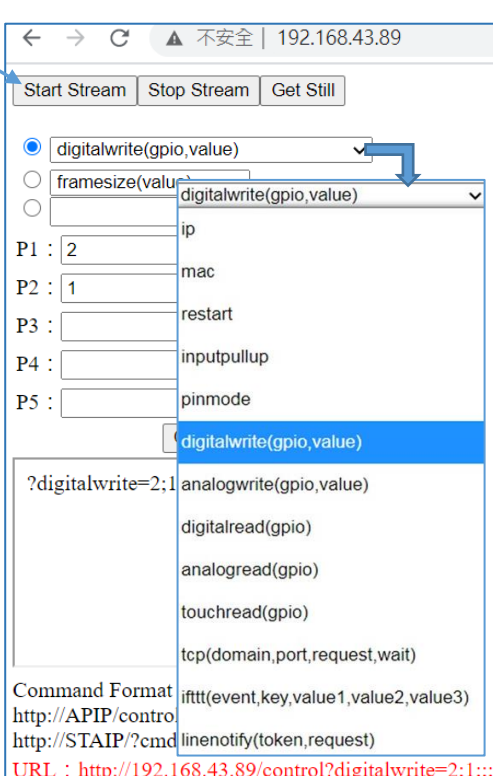
- 所謂 STA(Station) 模式是指將 ESP32 CAM 與電腦都同時連到一個現有的區域網路 TSSH，但動態分配的位置都不一定，如 192.168.43.89，每人都不同
- AP(Access Point) 模式是指將 ESP32 CAM 當成臨時 AP 分享器，電腦可直接連到 ESP32CAM 的 AP 名稱，位置固定為 192.168.4.1，可離線使用，但不可對外，如 Telegram 不可使用

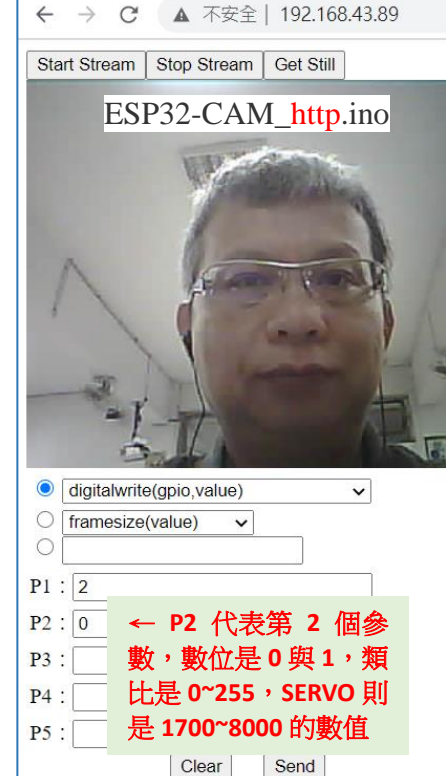


`analogwrite` 則值介於 0~255。LED 發光二極體長腳為正，接左邊 GPIO2，短腳為負，接右邊 GND，如左下圖，當瀏覽器輸入 `http://192.168.43.89?digitalwrite=2;1`，即可透過網址控制 LED 亮滅；複製修改第 26 行『`http://192.168.xxx.xxx?flash=value`』，`value` 介於 0~255，即可控制閃光燈不同亮度。

按下 Start Stream 才會看到影像，以節省網路流量







← P2 代表第 2 個參數，數位是 0 與 1，類比是 0~255，SERVO 則是 1700~8000 的數值

7. 若沒用到 Telegram 或 GoogleDrive、Linenotify，建議用法蘭斯改寫官方的 `ESP32-CAM_http.ino`，Stream 串流畫面較流暢，而且參數可以直接輸入再按『Send 送出』，不用輸入網址就可控制，管理超方便，如上面中間與右邊的圖片所示，但若要用網址控制當然也可以，只是 `ESP32-CAM_http.ino` 網址的問號前面需多 `control` 命令，因為這是官方函式庫規定，若不外接 LED，可利用內建的閃光燈 GPIO4，建議閃光燈不要開太久，怕燒掉。將 LED 移除，改接 SERVO 伺服馬達，**橙色接 GPIO2，紅色接中間 5V，棕色接 GND**，中間透過長排針連接，複製 `ESP32-CAM_http.ino` 第 41 行『`http://192.168.xxx.xxx/control?servo=pin;value`』，`value` 數值介於 1700(0 度) ~ 8000(180 度)，改為『`http://192.168.43.89/control?servo=2;4850`』，則 SERVO 會轉到 90 度中間位置，或將 P2 改為 4850。
8. 複製修改 `ESP32-CAM_http.ino` 第 16 行『`http://192.168.xxx.xxx:81/stream`』，其中 :81 是指利用 PORT 埠 81 的位置來傳遞視訊的資料，如下圖。

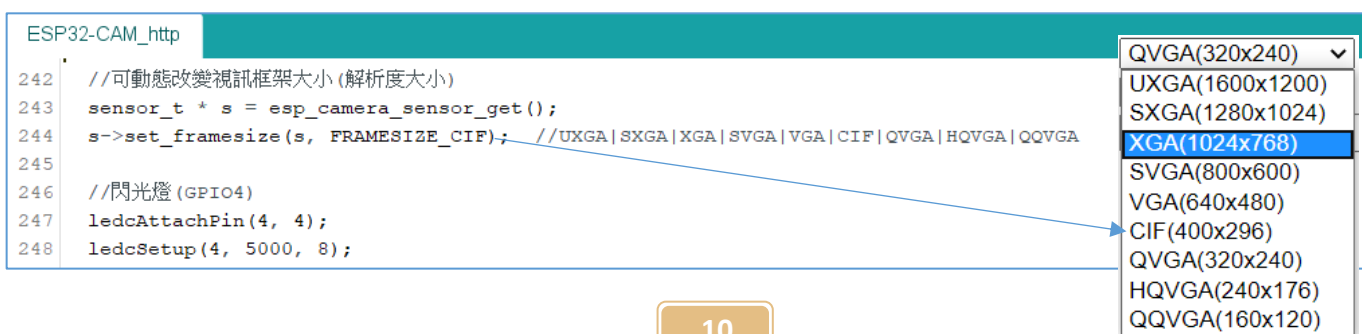
```

ESP32-CAM_http
10
11 查詢Client端IP:
12 查詢IP: http://192.168.4.1/control?ip
13 重設網路: http://192.168.4.1/control?resetwifi=ssid;password
14
15 http://192.168.xxx.xxx //網頁首頁管理介面
16 http://192.168.xxx.xxx:81/stream //取得串流影像 
<TR><TD></TD><TD></TD><TD></TD>
<TR><TD></TD><TD></TD><TD></TD>
<TR><TD></TD><TD></TD><TD></TD>
</TABLE>

```



- 若要變更監視畫質請修改第 244 行程式，第 247-248 行是設定 GPIO4 閃光燈使用第 4 通道，意謂將來其他 GPIO 若使用類比 PWM 控制，不可重複使用第 4 通道，同時 GPIO4 跟拍照電路有關，所以只要拍照完畢，一定要強制關閉閃光燈的腳位。



開機時，當連上網路，閃光燈快速閃爍，未連上慢速閃爍，藉此容易判斷是否無線網路連線中。

```
ESP32-CAM_http
275   for (int i=0;i<5;i++) {    //若連上WiFi設定閃光燈快速閃爍
276       ledcWrite(4,10);
277       delay(200);
278       ledcWrite(4,0);
279       delay(200);
280   }
281 }
282 else {
283     WiFi.softAP((WiFi.softAPIP().toString()+"_"+(String)apssid).c_str(), appassword);
284
285     for (int i=0;i<2;i++) {    //若連不上WiFi設定閃光燈慢速閃爍
286         ledcWrite(4,10);
287         delay(1000);
288         ledcWrite(4,0);
289         delay(1000);
```

註：若發現閃光燈每隔一段時間會自動閃爍點亮，應是電源不穩定，造成反覆重新啟動所造成。

12. 學習自訂指令，請複製貼上下圖選取的位置，增加控制閃光燈或其他指令，此步驟較有挑戰性，請自行斟酌實際修改程式看看！

```
ESP32-CAM_http
550 //自訂指令區 http://192.168.xxx.xxx/control?cmd=P1;P2;P3;P4;P5;P6;P7;P8;P9
551 if (cmd=="your cmd") {
552     //Feedback變數為回傳客戶端內容，可為一般文字或HTML語法
553     Feedback="<font color=\"red\">Hello World</font>";
554 }
555 else if (cmd=="ip") { //查詢IP
556     Feedback="AP IP: "+WiFi.softAPIP().toString();
557     Feedback+=", ";
558     Feedback+="STA IP: "+WiFi.localIP().toString();
559 }
560 else if (cmd=="mac") { //查詢MAC
561     Feedback="STA MAC: "+WiFi.macAddress();
562 }
```

13. ESP32-CAM\_http.ino 透過網址就可以同時控制四個伺服馬達，第 1 個數字是腳位編號，第 2 個腳位角度，例如 <http://192.168.xxx.xxx/control?servo=12;1700;13;4850;14;8000;15;4850>
14. ESP32 CAM 的 SERVO 跟 Arduino 使用方式不同，原本角度是介於 1700~8000 之間，如底程式，但可以利用  $\text{angle} \times 6300 / 180 + 1700$  這就可以將 0-180 轉換成 1700~8000。

```
41 http://192.168.xxx.xxx/control?servo=pin;value //value= 1700(0度) ~ 8000(180度)
```

```
634     else if (cmd=="servo") {
635         ledcAttachPin(P1.toInt(), 3);
636         ledcSetup(3, 50, 16);
637
638         int val = P2.toInt();
639         if (val > 8000)
640             val = 8000;
641         else if (val < 1700)
642             val = 1700;
643         val = 1700 + (8000 - val);
644         ledcWrite(3, val);
    }

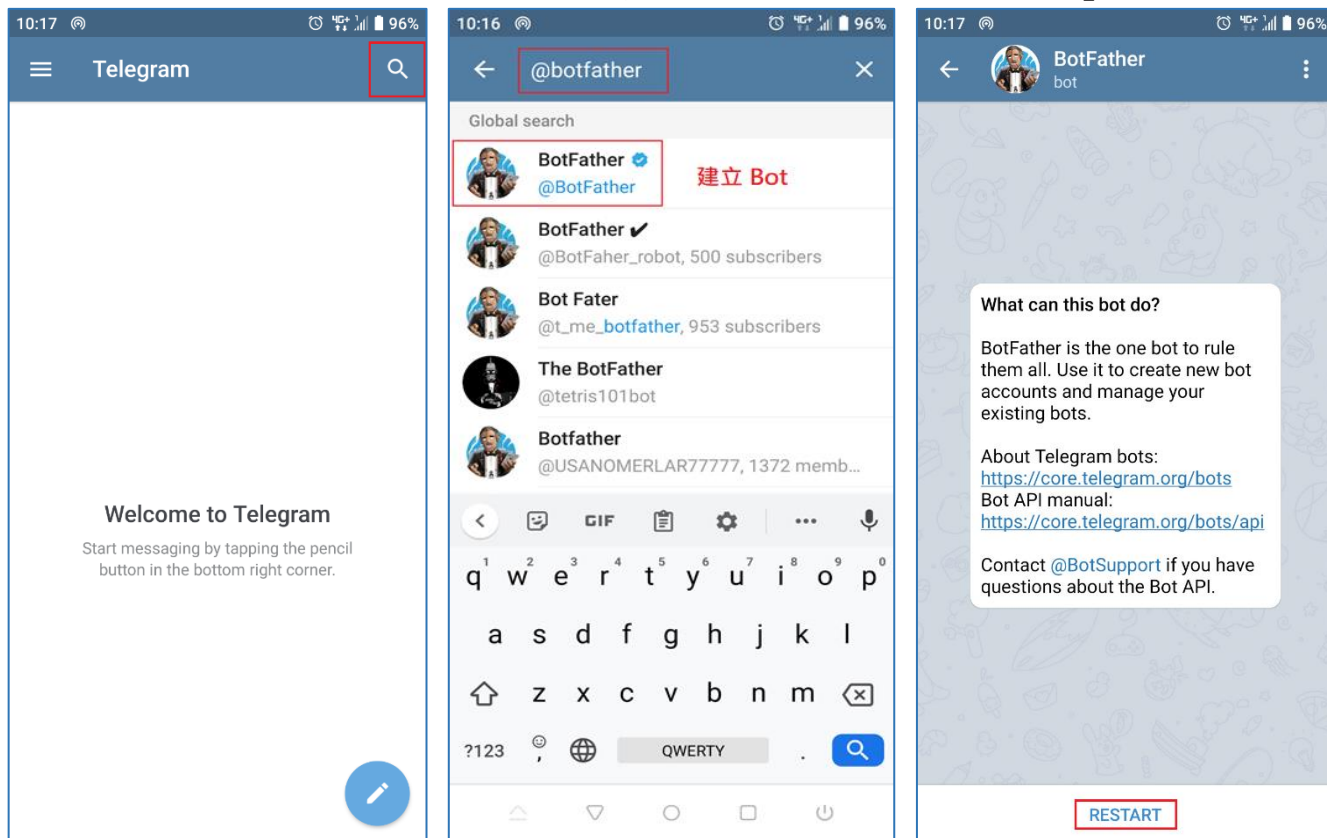
int transferAngle(int angle, String side) {
    if (angle > 180)
        angle = 180;
    else if (angle < 0)
        angle = 0;
    if (side=="right")
        angle = 180 - angle;
    return angle*6300/180+1700;
}
```

15. 伺服馬達 SERVO 分成兩種，外觀非常相似，「不連續型」伺服馬達可控制旋轉角度 0~180 度之間，如機器人關節或門禁開關門動作；而「連續型」伺服馬達旋轉角度 360 度，無法控制旋轉角度，但可以控制正轉(0 度)或反轉(180 度)以及轉速，可應用於車子前進或後退，90 度則為停止。

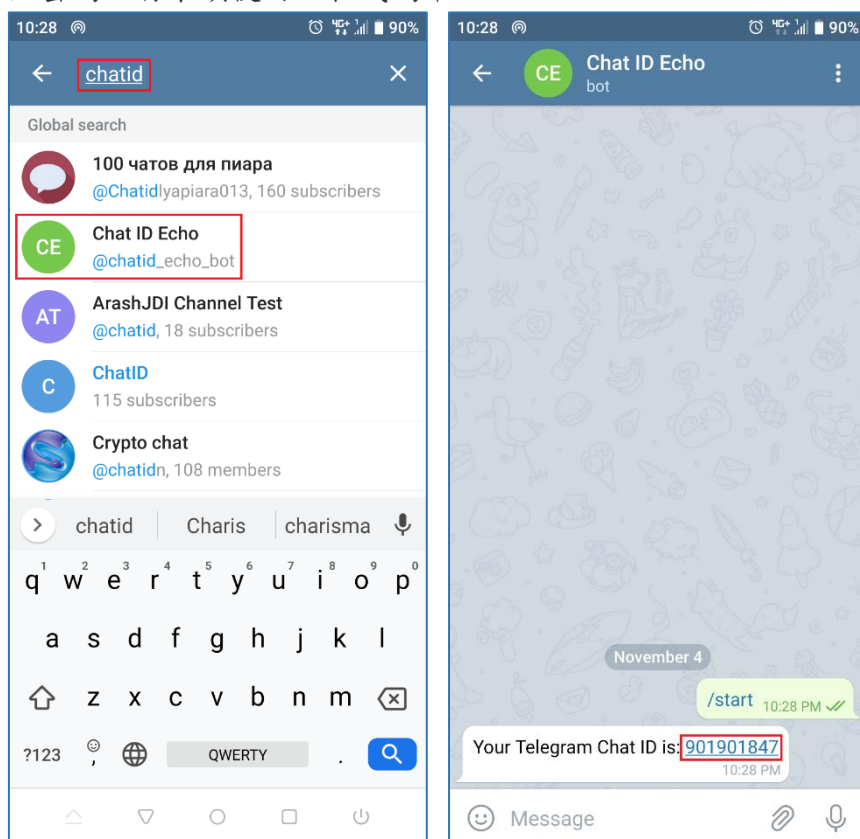


### 實作 03 : ESP32-CAM\_PIR\_Telegram.ino 當有人經過，自動將擷取照片上傳到免費 Telegram

1. 若使用 Line Notify 接受照片，一小時只能接收 **50 張** 影像的限制，使用 Telegram 傳送影像則**無限制**，進階的 Line bot 機器人建立需要用 node.js 或 google script 架設，但 telegram bot 只需要在 telegram 上操作就可以建立了，所以若玩 ESP32 CAM 就少用 Line Notify 與 Line bot，改用 Telegram，申請會用到手機當場認證，相關影片請參閱 [https://www.youtube.com/results?search\\_query=telegram+bot](https://www.youtube.com/results?search_query=telegram+bot)
2. 先用手機安裝 **Telegram** APP，安裝後按右上角的放大鏡，搜尋『**botfather** 機器人之父』，點選第一個建立 Bot 機器人，出現對話頁面後，按下最下方『**RESTART** 重新開始啟用』。



3. 接著按右上角的放大鏡，搜尋『**chatid** 聊天室』好友，選擇下圖『Chat ID Echo』，點選後可看到一組密碼，將來須使用於程式碼中。



4. 桌機網址 <https://web.telegram.org/> 官網，第一次登入須先輸入手機號碼認證，但第一個數字 0 不用輸入，按 OK，查看手機 Telegram 的通知(※要稍等)，輸入密碼後方可登入，然後按『允許』。

Telegram Web

web.telegram.org/#/login

Sign in

Please choose your country and enter your full phone number.

Country  
Taiwan

Code  
+886

Phone number  
955974622

Is this phone number correct?

+886 955974622

CANCEL OK

+886 955974622

Edit phone number

We've sent the code to the Telegram app on your other device.  
Please enter the code below.

You will be able to request SMS in 59:35

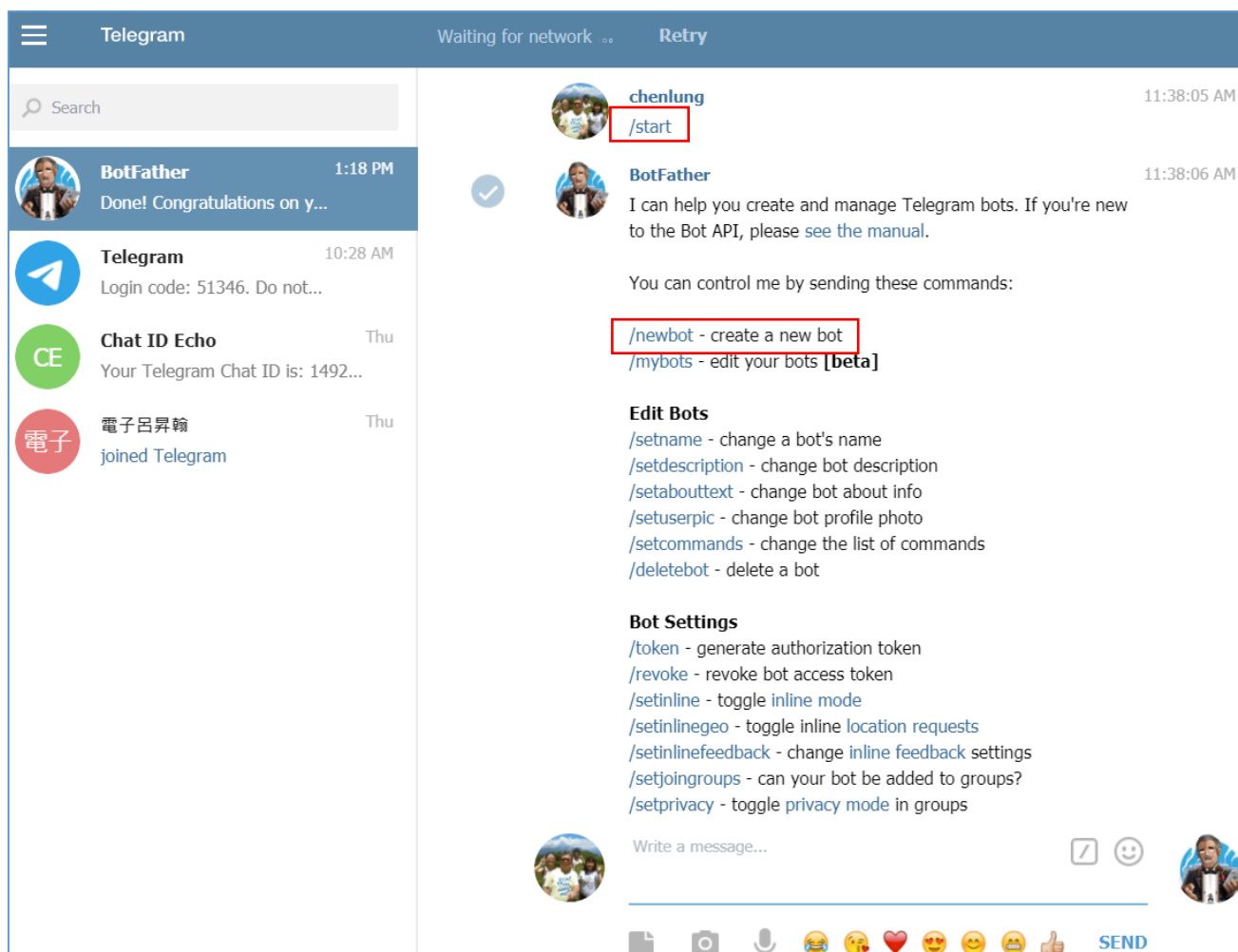
Enter your code  
5143

web.telegram.org要求下列權限：

顯示通知

允許 封鎖

5. 點選左側 Botfather，Write a message 寫入一個訊息可用鍵盤輸入『/start』開始，對話視窗會列出所有功能，然後選擇『/newbot 新機器人』。



6. 輸入『獨一無二』的機器人名稱，不可跟別人相同，接著輸入剛剛相同的名稱，但後面多加『\_bot』，不可輸入錯誤，若成功會收到恭喜建立成功與 API token 授權碼的通知，未來想查看，可利用『/mybots』我的機器人即可列出所有機器人清單，點選該機器人名稱，選擇 API token，最後步驟最重要，很容易忽略，就是點擊右下圖的『@XXXXXX\_bot』機器人，然後按 **START** 啟用，很重要<sup>3</sup>。

The screenshot shows the Telegram BotFather interface. On the left, a chat with 'chenlung' shows the bot creation process: the user enters 'chenlung0815', then 'chenlung\_bot', and BotFather provides instructions and an API token: '1455817938:AAH1xsZwNYS3VlxcZdubfoBsBTj0ch7CtDw'. On the right, a chat with 'BotFather' shows the '/mybots' menu with options like 'API Token', 'Edit Bot', 'Bot Settings', 'Payments', 'Transfer Ownership', 'Delete Bot', and 'Back to Bots List'. Below that, another chat shows the 'API Token' screen with the token '1455817938:AAH1xsZwNYS3VlxcZdubfoBsBTj0ch7CtDw' and a 'Revoke current token' button.

7. 選擇『Chat ID Echo』，複製右側的授權碼，待會要貼到 ESP32-CAM\_PIR\_Telegram.ino

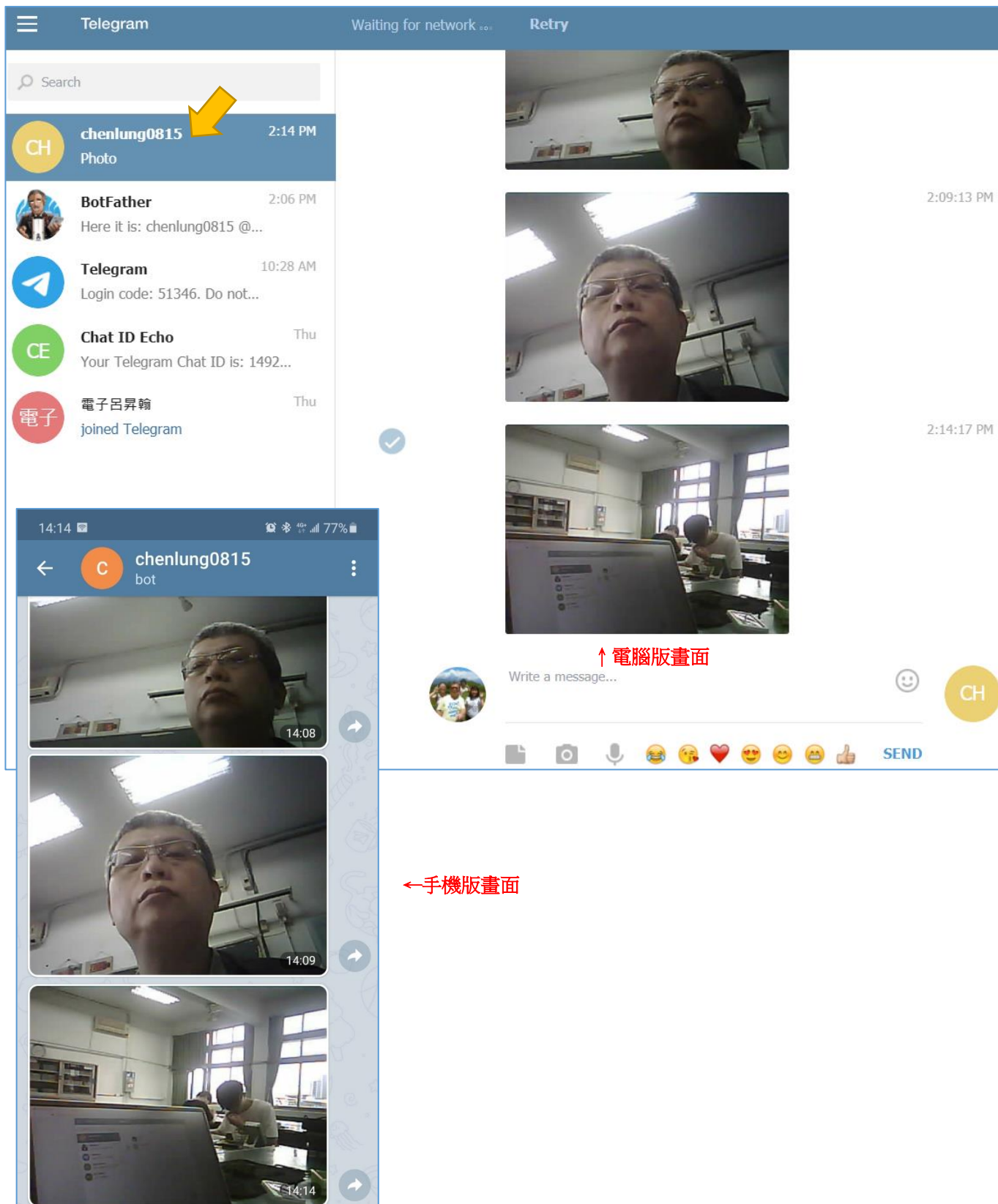
The screenshot shows a Telegram chat interface. The top bar indicates the chat is with 'Chat ID Echo bot'. The left sidebar shows a list of chats, including 'BotFather', 'Telegram', 'Chat ID Echo', and '電子呂昇翰'. The main chat area shows a message from 'Chat ID Echo' with the text 'Your Telegram Chat ID is: 1492376388'. The chat ID is highlighted with a red box. The bottom of the screen shows a 'Write a message...' input field and a 'SEND' button.



8. 將步驟 6 一長串內容貼到 ESP32-CAM\_PIR\_Telegram.ino 第 13 行，步驟 7 內容貼到第 14 行，PIR 腳位是在 GPIO15，然後燒錄上傳測試看看，但請勿使用於非法的偷拍行為喔！。

```
ESP32-CAM_PIR_Telegram $
10 const char* ssid      = "TSSH";    //Wi-Fi帳號
11 const char* password = "12345678"; //Wi-Fi密碼
12
13 String token = "1455817938:AAHlxszWnYS3VlxcZdubfoBsBTj0ch7CtDw";
14 String chat_id = "1492376388";    //搜尋 chatid_echo_bot 對話取得 chat_id
15 int pinPIR = 15;                  //人體移動感測器腳位
```

9. 手機與桌機的瀏覽器顯示結果如下，若看不到有可能是步驟 6 未啟動『@XXXXXXX\_bot』機器人。



10. 第 158 行是每隔 10 秒才去判斷 PIR，否則圖片太多佔頻寬。若欲用文字通知，節省頻寬，可將傳圖片的第 154 行變成註解，在下面插入『`sendMessage2Telegram(token, chat_id, "Alert!");`』

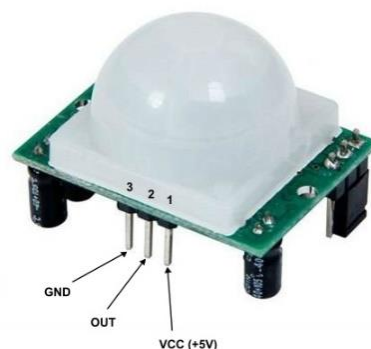
```
149 void loop()
150 {
151   int v = digitalRead(pinPIR);
152   Serial.println(v);
153   if (v==1) {
154     sendCapturedImage2Telegram(token, chat_id); //不顯示傳送結果
155     //Serial.println(sendCapturedImage2Telegram(token, chat_id)); //顯示傳送結果
156     delay(10000);
157   }
158   delay(1000);
159 }
```

11. PIR 當人體經過會產生電位的變化，預設是接到 GPIO15，若有使用到 SD 卡，建議改用 GPIO13。

## PIR 人體移動感測器

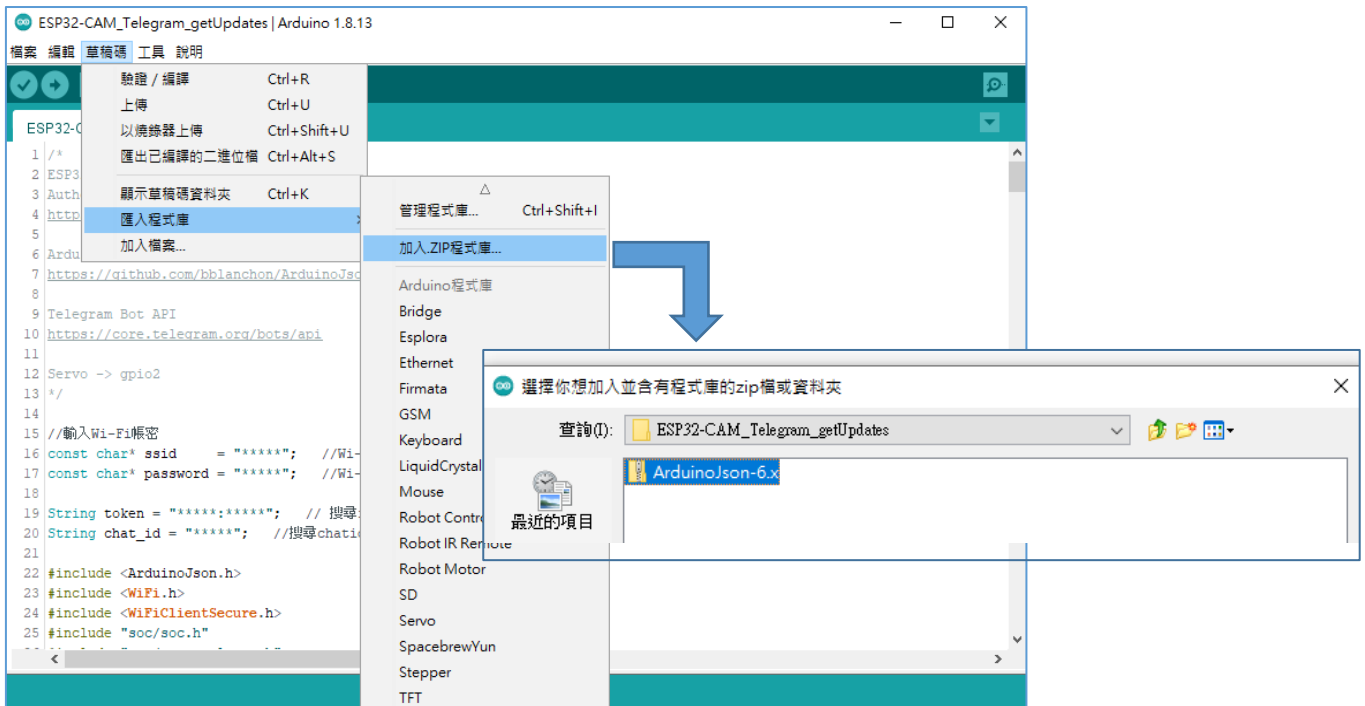


1. 可偵測距離3~7公尺
2. SX ( Sensitivity ) 旋鈕：  
靈敏度：順時鐘旋轉增加靈敏度
3. TX ( Time Delay ) 旋鈕：  
延遲偵測時間：5~300秒  
順時鐘增加延遲偵測時間
4. 初始測試時，靈敏度調至最高、  
延遲偵測時間調至最低
5. 接線：3.3V、GND、IO15

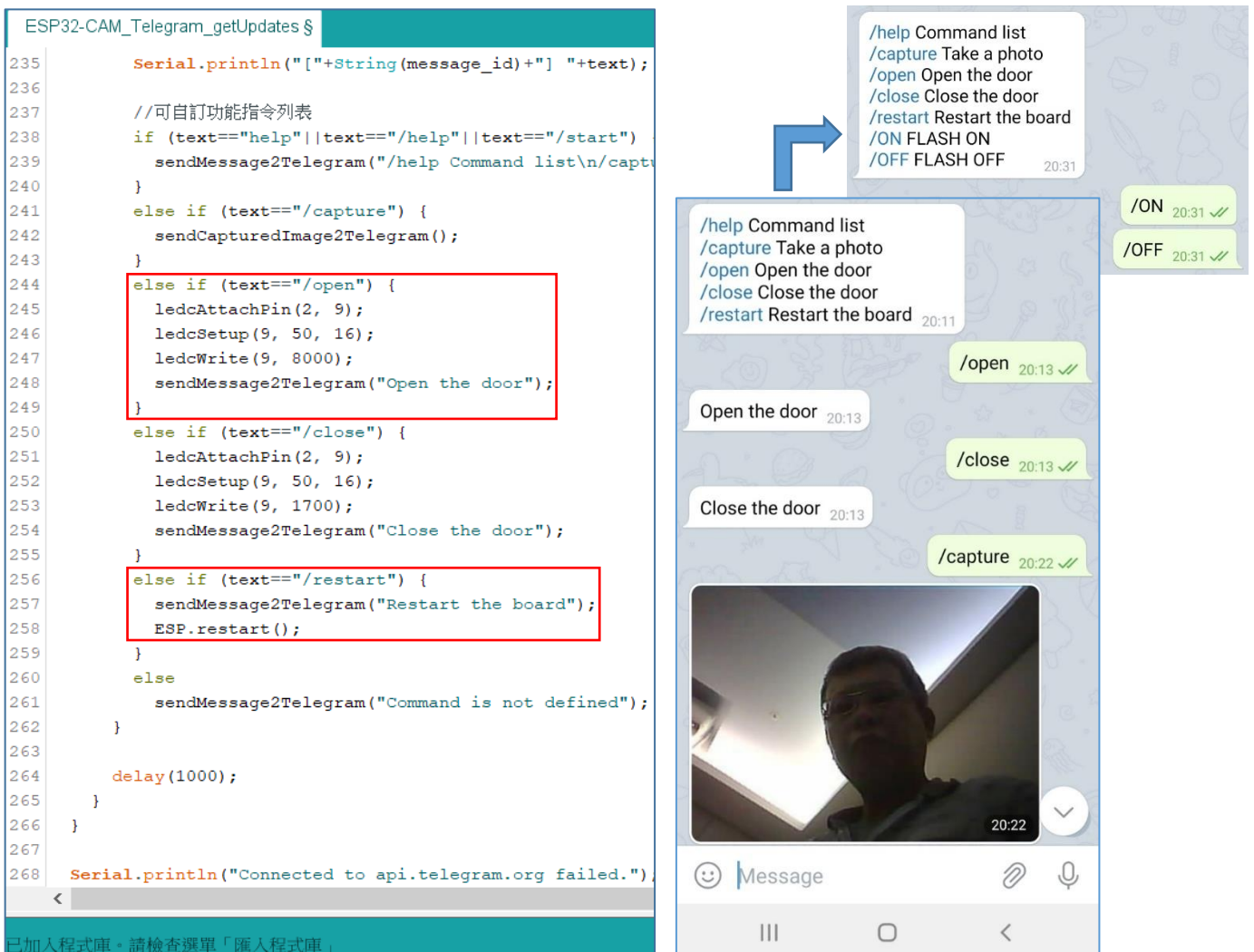


## 實作 04 : ESP32-CAM\_Telegram\_getUpdates.ino Telegram 機器人下達指令控制 GPIO 或擷取照片。

1. 依照下面步驟安裝 **ArduinoJson-6.x.zip** 程式庫，並修改 Telegram 的 **token** 與 **chat\_id** 再燒錄。



2. 自行修改第 239 行『sendMessage2Telegram("/help Command list\n/capture Take a photo\n/open Open the door\n/close Close the door\n/restart Restart the board\n/nON FLASH ON\n/nOFF FLASH OFF");』，開機時主動通知列出所有指令，點選項目即可執行，如 SERVO 接 GPIO2，當機器人輸入『/open』則會開門，可自行再增加指令，如複製貼上第 244-249 行再修改，其中『\n』是指換行。



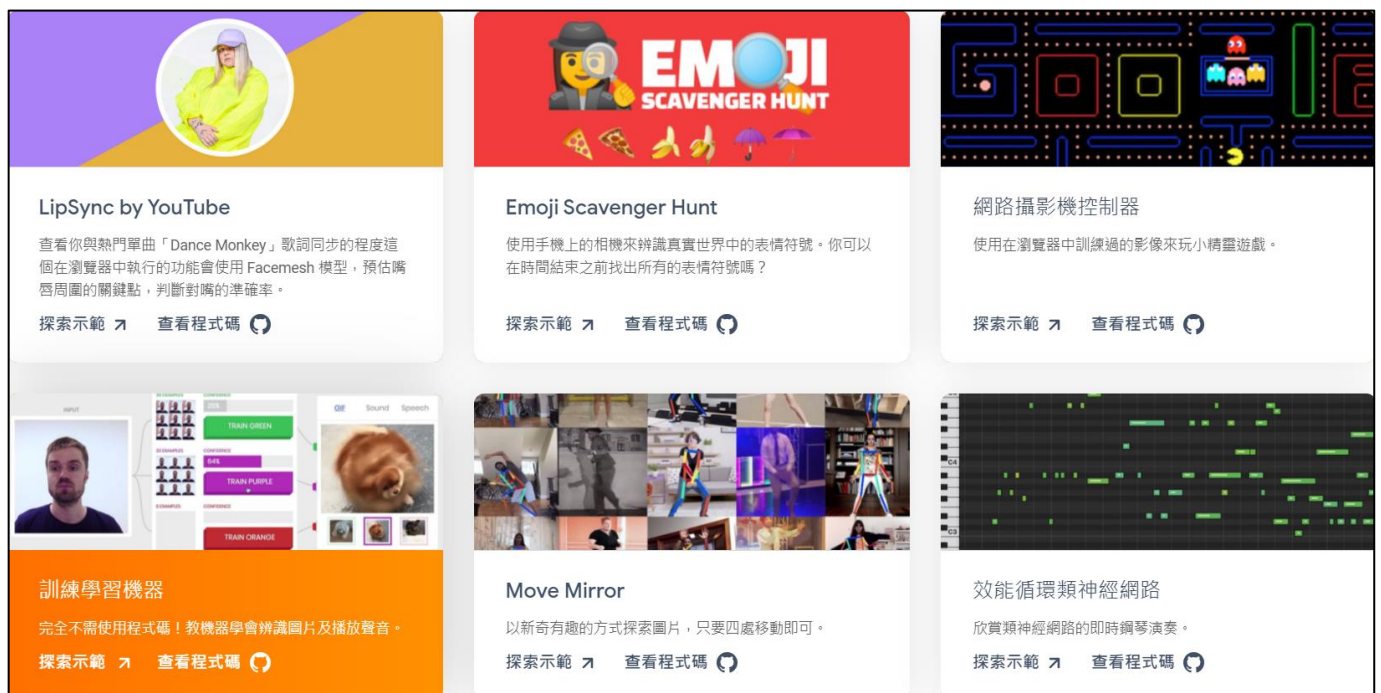


**實作 05:ESP32-CAM\_SD\_PIR.ino** 當有人經過將照片上傳到 SD 卡，但 SD 限 FAT32 格式，大小限 16GB 以下，同時建議開啟監視視窗，以觀察存檔的內容，命名方式是 0~255.jpg。

1. 由於 ESP32 CAM 使用數個 GPIO 腳位控制 Micro SD 記憶卡，而因為原有 PIR 是接 GPIO15，有時會造成無法正確使用 SD，建議改用杜邦線將 PIR 接到 **GPIO13**。
2. 每次拍照，可觀察閃光燈會自動亮一下。
3. 修改第 145 行『s->set\_framesize(s, FRAMESIZE\_CIF);』可以變更預設的畫質。

```
COM4
17:32:59.627 -> E (28733) sdmmc_req: sdmmc_host_wait_for_event returned 0x107
17:32:59.627 -> E (28733) sdmmc_cmd: sdmmc_write_sectors_dma: sdmmc_send_cmd returned 0x107
17:32:59.627 -> E (28734) diskio_sdmmc: sdmmc_write_blocks failed (263)
17:33:07.346 -> 
17:33:08.044 -> E (2137) sdmmc_io: sdmmc_io_reset: unexpected return: 0x108
17:33:08.044 -> Card Mount Failed ←Failed 失敗
17:33:10.247 -> Picture file name: /9.jpg
17:33:10.647 -> Saved file to path: /9.jpg
17:33:10.647 -> File Size = 117668
17:33:10.647 -> Success
17:33:10.647 -> 
17:33:21.716 -> Picture file name: /10.jpg
17:33:21.967 -> Saved file to path: /10.jpg ←成功 Success
17:33:22.017 -> File Size = 93737
17:33:22.017 -> Success
17:33:22.017 -> 
[ ] 自動捲動 [x] Show timestamp NL & CR 115200 baud Clear output
```

- Google Tensorflow.js(tfjs)介紹 <https://www.tensorflow.org/js/>，可按最下方查看所有相關範例。

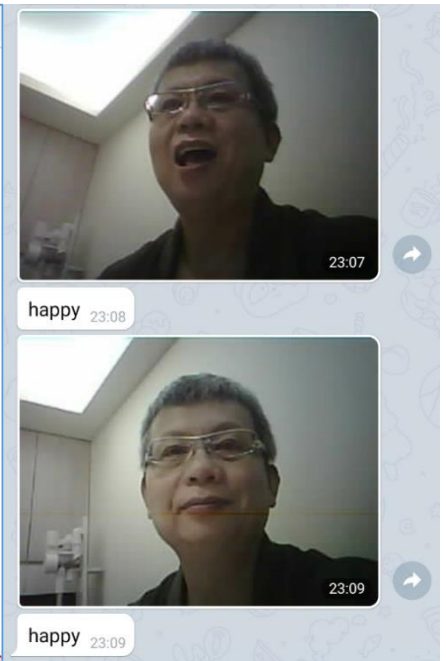
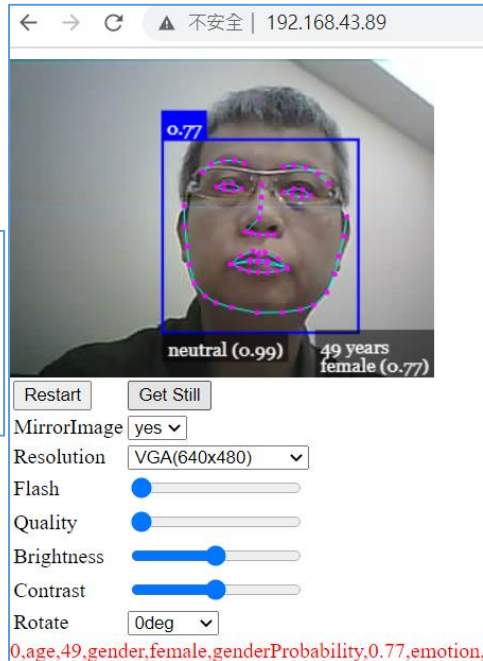


- Tensorflow.js 已訓練模型 <https://github.com/tensorflow/tfjs-models>

## 實作 06 : ESP32-CAM\_face-api\_FaceDetection.ino

1. 利用 Google 建立好的 Tensorflow 訓練模型，辨識人臉目前的表情、性別、年齡，由於資料庫模型是用西方臉孔建立，所以我們實測會較年輕...
2. 辨識人臉目前的表情是利用第 483~506 行找出各式表情(情緒)的最大值 max，如 sad 傷心、angry 憤怒、fearful 可怕、disgusted 噁心的。
3. 第 509~512 行是判斷表情感到快樂時，才會傳送圖片與文字傳到 Telegram，但表情要誇張一點才會變 happy 喔！其實智慧相機也有捕捉到笑臉自動拍照的功能，大家用力地笑吧....

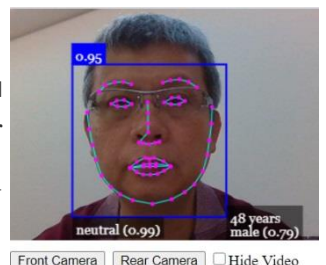
```
22:59:05.430 -> Connecting to TSSH
22:59:05.430 ->
22:59:05.430 -> STAIP address:
22:59:05.430 -> 192.168.43.89
22:59:07.419 ->
22:59:07.419 -> APIP address:
22:59:07.419 -> 192.168.4.1
```



```
483     if (expressions.happy>maxProbability) {
484         maxProbability=expressions.happy;
485         maxEmotion="happy";
486     }
487     if (expressions.sad>maxProbability) {
488         maxProbability=expressions.sad;
489         maxEmotion="sad";
490     }
491     if (expressions.angry>maxProbability) {
492         maxProbability=expressions.angry;
493         maxEmotion="angry";
494     }
495     if (expressions.fearful>maxProbability) {
496         maxProbability=expressions.fearful;
497         maxEmotion="fearful";
498     }
499     if (expressions.disgusted>maxProbability) {
500         maxProbability=expressions.disgusted;
501         maxEmotion="disgusted";
502     }
503     if (expressions.surprised>maxProbability) {
504         maxProbability=expressions.surprised;
505         maxEmotion="surprised";
506     }
507
508     //當可能性最大的表情是happy時
509     if (maxEmotion=="happy") {
510         $.ajax({url: document.location.origin+'?telegram_image', async: false}); //傳送影像
511         $.ajax({url: document.location.origin+'?telegram_text='+maxEmotion, async: false}); //傳送文字
512     }
```



4. 若不透過 ESP32 CAM 的鏡頭，亦可直接利用網頁版執行，網址為『[https://fustyles.github.io/webduino/TensorFlow/Face-api/Face-api\\_FaceDetection\\_video.html](https://fustyles.github.io/webduino/TensorFlow/Face-api/Face-api_FaceDetection_video.html)』需用手機或筆電，因為要有攝影機鏡頭方能辨識，下方出現『Please wait for loading model.』表示還在載入模型中，或用手機掃描右上 QR code，建議使用 Chrome 瀏覽器，前後鏡頭可切換，用手機要橫擺。



## 實作 07 : ESP32-CAM\_face-api\_FaceRecognition.ino

1. 實作 06 Detection 是偵測，本實作 Recognition 是指辨識，由於需要和網站的指定圖片做比對，但各位應該都無個人網站，所以請先申請 Github 免費帳號 <https://github.com/join?source=login>，無限量的免費私人儲存庫，姓名不可重複，密碼需要至少 8 個字元，並包括英文大小寫與數字，驗證是怕有駭客用機器人申請，都成功後按下最下方 **Create account** 建立帳號按鈕。

The image shows the GitHub account creation process in three stages:

- Stage 1:** Username selection. The input 'chenlung' is rejected with the message: "Username chenlung is not available. chenlung-gif, chenlung-byte, or chenlung116 are available." The input 'chenlung88' is accepted with the message: "chenlung88 is available."
- Stage 2:** A CAPTCHA challenge titled "選擇呈螺旋狀的星系" (Select the spiral galaxies). It shows a 2x3 grid of galaxy images. A red arrow points from the 'chenlung88 is available' message to this stage.
- Stage 3:** The main "Create your account" form. It includes fields for Username, Email address, and Password. Below these is an "Email preferences" section with a checkbox for "Send me occasional product updates, announcements, and offers." At the bottom is a "Verify your account" section with a text box and a "驗證" (Verify) button. A red arrow points from the CAPTCHA stage to this button.


2. 第一次進入會詢問申請者身分、寫程式的經驗、任選 3 個申請目的、興趣，最後按 **Complete setup**。

The image shows the "Welcome to GitHub" setup page with the following sections:

- What kind of work do you do, mainly?** A grid of buttons for different roles: Software Engineer, Student, Product Manager, UX & Design, Data & Analytics, Marketing & Sales, Teacher, and Other.
- How much programming experience do you have?** A grid of buttons for experience levels: None, A little, A moderate amount, and A lot.
- What do you plan to use GitHub for?** A grid of buttons for various uses: Learn to code, Learn Git and GitHub, Host a project (repository), Create a website with GitHub Pages, Collaborating with my team, Find and contribute to open source, School work and student projects, Use the GitHub API, and Other.



3. 到電子郵件認證，按下 **Verify email address** 即可進入 Github



**Please verify your email address**

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to [chenlung@tssh.ntpc.edu.tw](mailto:chenlung@tssh.ntpc.edu.tw).

[Resend verification email](#) [Change your email settings](#)

Almost done, @chenlung88! To complete your GitHub sign up, we just need to verify your email address: [chenlung@tssh.ntpc.edu.tw](mailto:chenlung@tssh.ntpc.edu.tw).

**Verify email address**



Once verified, you can start using all of GitHub's features to explore, build, and share projects.

4. 按下首頁左側 **Create repository** 建立資料庫，填入資料庫名稱，其餘都不改，再按 **Create repository**。

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

**Owner \*** **Repository name \***

 **chenlung88** / **ESP32CAM** 

Great repository names are short and memorable. Need inspiration? How about **reimagined-octo-chainsaw**?

**Description (optional)**

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.


☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)


☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)


☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

**Create repository** 

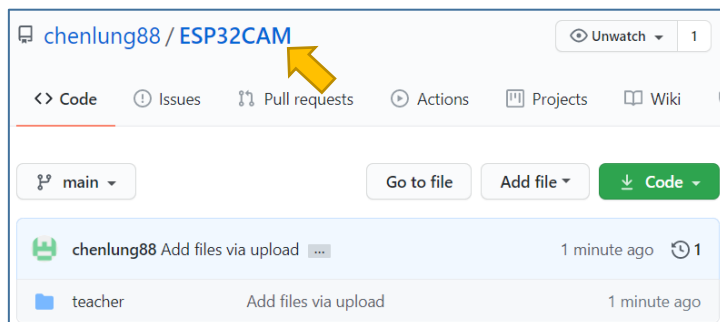
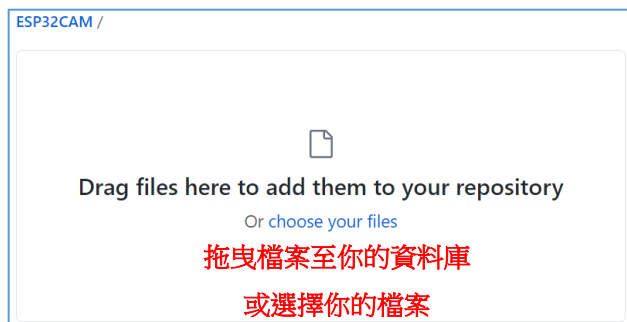
5. 點選『**uploading an existing file** 上傳一個存在的檔案』。

**Quick setup — if you've done this kind of thing before**

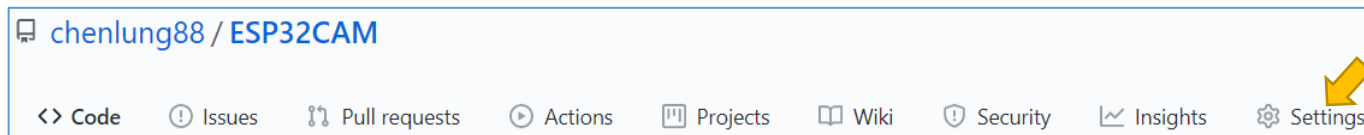
 **Set up in Desktop** or **HTTPS** **SSH** <https://github.com/chenlung88/ESP32CAM.git>

Get started by [creating a new file](#) or [uploading an existing file](#).  We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

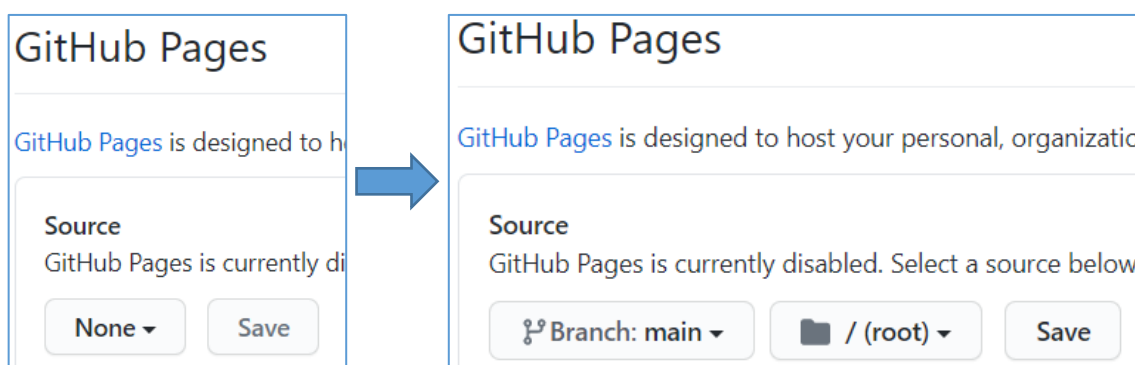
6. 在桌面先建立『teacher』的資料夾，裡面至少放2張以上的大頭照，可用小畫家調整大小並選取剪裁，背景不要有其他人，背景越單純越好，然後利用拖曳方式將整個資料夾『拖曳』到底下的方框內，再按 **Commit changes** 提交改變，然後才會開始上傳。



7. 按右上圖『ESP32CAM』資料庫名稱，然後選擇最右邊的『Settings 設定』。



8. 拉到最下面，將 None 改為『main 主要』，然後按『Save 儲存』。



9. 測試上傳的圖片是否已經成功，網址為『<https://chenlung88.github.io/ESP32CAM/teacher/1.jpg>』，其中 **chenlung88** 是帳號名稱，**ESP32CAM** 是資料庫名稱，**teacher** 是第 1 位人名資料夾。
10. 自訂指令，當偵測到認識的人，會透過 Telegram 傳回姓名文字與照片(有可能只是相似的人)，然後利用 SERVO 開門(數值介於 1700~8000 之間)

```
189 else if (cmd=="telegram_text") {
190     sendMessage2Telegram(token, chat_id, P1);
191 }
192 else if (cmd=="telegram_image") {
193     sendCapturedImage2Telegram(token, chat_id);
194 }
195 else if (cmd=="servo") {
196     int pin = P1.toInt();
197     int channel = P3.toInt();
198     ledcAttachPin(pin, channel);
199     ledcSetup(channel, 50, 16);
200
201     int val = P2.toInt();
202     if (val > 8000)
203         val = 8000;
204     else if (val < 1700)
205         val = 1700;
206     val = 1700 + (8000 - val);
207     ledcWrite(channel, val);
208 }
```

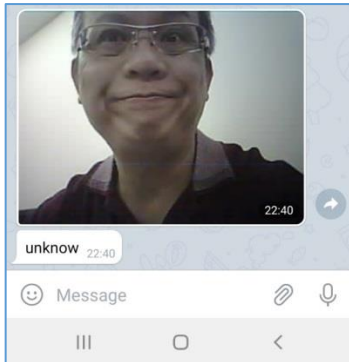
11. 第 396 行需改為自己存放雲端的絕對位置，請將步驟 9 的內容取代，不含最後人名資料夾與圖檔名稱，例如『<https://chenlung88.github.io/ESP32CAM/>』，第 397 行是該資料夾裡面的子資料夾名稱也要修改，每個名稱代表人名，雲端與程式碼名稱與數量均需一致，第 398 行代表每個人名的資料夾需辨識的照片數量，至少 2 張，裡面一定是流水號檔名，如 1.jpg、2.jpg、3.jpg，照片以大頭照為主，像素不可太高，節省判斷的時間，但也不可以很小，第 399 行是判斷的臨界值，低於預設 0.4，表示很接近預設圖片，則會顯示該人名，若高於 0.4 表示誤差很大，表示是陌生人。

```
396 const faceImagesPath = 'https://chenlung88.github.io/ESP32CAM/'; //人名命名的資料夾路徑
397 const faceLabels = ['teacher', 'SAMLU']; //人名命名的資料夾列表
398 faceImagesCount = 3; //每個人名命名的資料夾內的照片數，以流水編號命名JPG圖檔 1.jpg, 2.jpg...
399 const distanceLimit = 0.4; //距離上限，若超過此值則顯示unknown，否則顯示人名
```

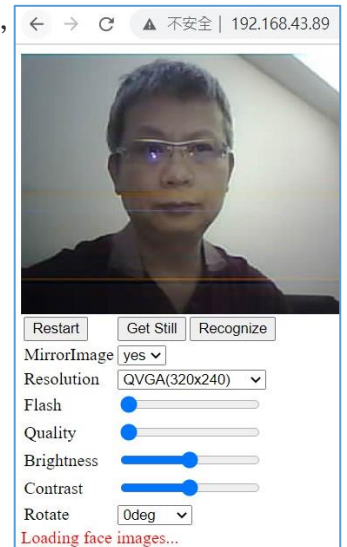
12. 當辨識到某人正確時，則會自動開門(SERVO 接 GPIO2)，請務必修改第 513 行的人名，第 514 行『servo=2;4850;9』的 2 是指用腳位 GPIO2，4850=(1700+8000)/2，約 90 度，最後 9 代表通道。

```
508 //辨識到陌生人unknown傳送至Telegram，並控制伺服馬達開門
509 if (result.label=="unknown") {
510     $.ajax({url: document.location.origin+'?telegram_image', async: false}); //傳送影像
511     $.ajax({url: document.location.origin+'?telegram_text=unknown', async: false}); //傳送文字
512 }
513 else if (result.label=="teacher") //辨識到主人人名控制伺服馬達開門
514     $.ajax({url: document.location.origin+'?servo=2;4850;9', async: false}); //控制Servo轉動
```

13. 若辨識搭配自動追蹤容易跟深度學習一樣吃記憶體，持續辨識久了就當了，所以設計需按 **Recognize** 按鈕才辨識，第一次辨識會稍微久一點，因為第一次辨識需先載入比對照片，辨識第二次就較快，若辨識成功，底下會出現『{"\_label":"teacher","\_distance":0.38921821866264056}』，distance 距離值小於第 399 行預設 0.4，此時 SERVO 會將門打開，若是陌生人則底部會出現『{"\_label":"unknown","\_distance":0.604569397201771}』，因為該值大於 0.4，代表差很多，此時才會將照片傳到 Telegram，所以 Telegram 授權碼與 ID 一定要修改，不然照片就會被我看光光。



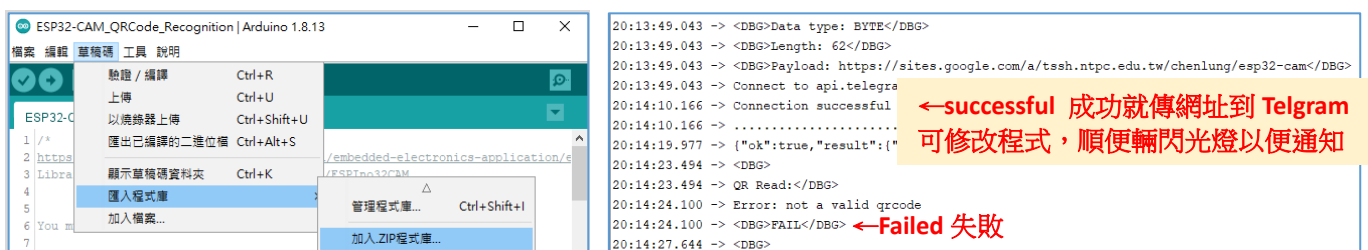
(※辨識成功不會傳到 Telegram，僅直接開門)



## 實作 08：ESP32-CAM\_QRCode\_Recognition(離線版，無顯示視窗)



1. 利用 ESP32 CAM 當作二維 QR Code 掃描鏡頭，當辨識成功會將網址傳到 Telegram 或 Line Notify。
2. 編譯前須外加 **ESPIno32CAM-master.zip** 程式庫，建議同時開啟監視視窗觀察，程式每隔一段時間自動判斷，若看到『**Error: not a valid qrcode**』，表示不是有效的 QR Code，監視視窗如右下圖。





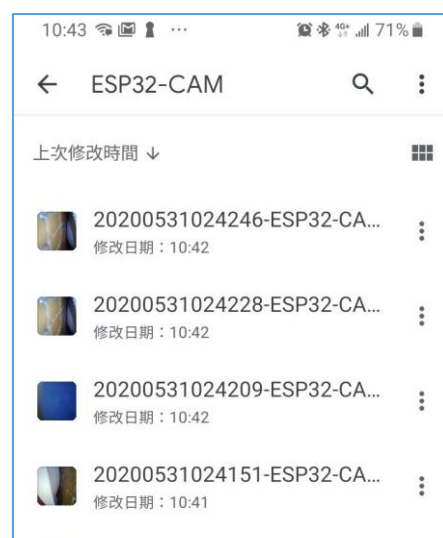
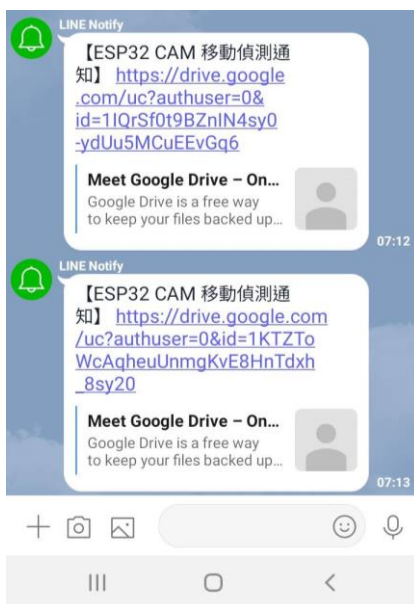
**實作 09:ESP32-CAM\_PIR\_GoogleDrive\_Linenotify.ino** 當有人經過將照片上傳到 LINE 與雲端硬碟，僅限建立 Script 程式碼人員可看到，當不想用 LINE，將第 29 行 token 刪掉即可，若要縮時攝影，刪除第 173 行判斷或改用 ESP32-CAM\_SD\_Manager\_PAG.ino，該程式還可存照片到 SD，但限 FAT32 格式。

```
27 const char* myDomain = "script.google.com";
28 String myScript = "/macros/s/AKfycbzbzYDTz3_qG7EmWoP874y10zLQsdbBW7uoo-QvhaJDsLoZwQTt/exec";
29 String myLineNotifyToken = "myToken=6go8P0etlOHierVoywAQN9gBMHTq96k6S407mQ961qB"; //Line
30 String myFoldername = "&myFoldername=ESP32-CAM";
31 String myFilename = "&myFilename=ESP32-CAM.jpg";
32 String myImage = "&myFile=";
```

```
161 void loop()
162 {
163   pinMode(gpioPIR, INPUT_PULLUP);
164   pinMode(GreenLED, OUTPUT);
165   pinMode(RedLED, OUTPUT);
166   int v = digitalRead(gpioPIR);
167   Serial.println(v);
168   ledcSetup(4, 5000, 8); //ledcSetup(LEDChannel, freq, resolution);
169   ledcAttachPin(4, 4); //ledcAttachPin(ledPin, LEDChannel);
170   digitalWrite(GreenLED, HIGH);
171   digitalWrite(RedLED, LOW);
172   ledcWrite(4, 0); //關閉閃光燈
173   if (v==1) {
174     digitalWrite(GreenLED, LOW);
175     digitalWrite(RedLED, HIGH);
176     ledcWrite(4, 10); //ESP32獨有的類比輸出指令，255太亮,10% Duty Cycle
177     SendCapturedImage();
178     digitalWrite(GreenLED, HIGH);
179     digitalWrite(RedLED, LOW);
180     ledcWrite(4, 0); //關閉閃光燈
181     delay(5000);
182   }
183   delay(1000);
184 }
```

建議 PIR 初值把延遲時間 (Delay) 先 **逆時針** 調到最底，也就是最低值，太久會持續被偵測拍照，敏感度 (Sensitivity) 先順時針調到底，也就是最高值，若莫名其妙自動拍照，就是靈敏度太高，若是反覆自動拍照，應是延遲太久。

欲修改解析度，請改第 158 行，但解析度太高可能出錯  
s->set\_framesize(s, FRAMESIZE\_VGA);

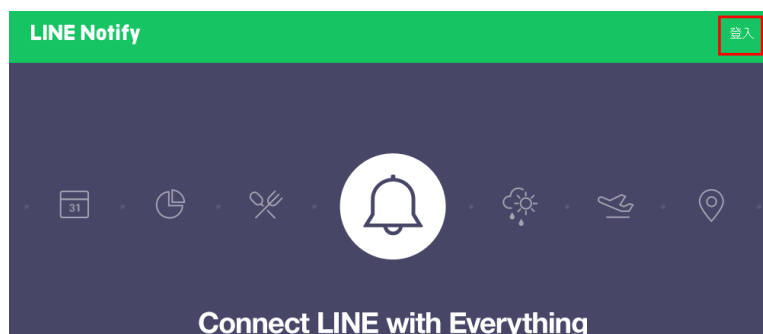


檔案名稱是用美國主機的時間，需自行 **+8 小時**，才是台灣正確時間，因為時差不同，最後 3 碼代表千分之幾秒，透過 Script 反應速度會稍慢。

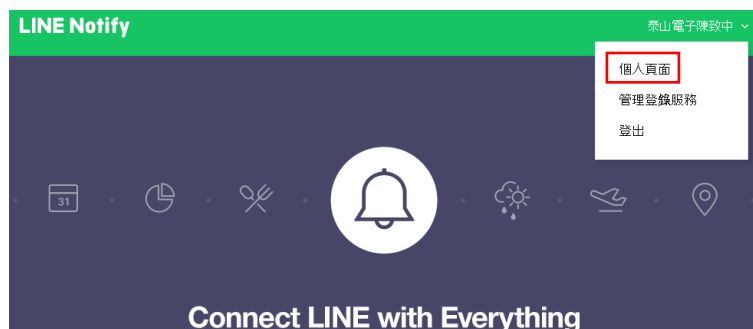
## 補充：取得 Line Notify Token 步驟如下

(註：Notify 是通知、Token 是權杖的意思)

1. 進入 [https://notify-bot.Line.me/zh\\_TW/](https://notify-bot.Line.me/zh_TW/) 官方網站，按下右上方『登入』按鈕，輸入登入的帳號與密碼，登入個人 Line 帳號，倘若帳號密碼忘記請不要問老師。



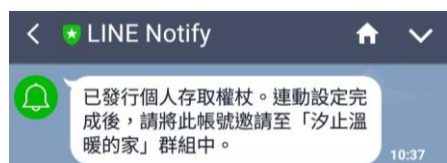
2. 右上方顯示帳號名稱，選擇名字旁邊的下拉式選單，選擇『個人頁面』。



3. 請選擇最下方『發行權杖』，第一個權杖名稱一定要填，未來在通知時，會出現【...】告知發話人是誰或通知用途，內容最多 20 個字，第二個是選擇通知的對象可以是『1 對 1』，也可以直接選擇任意一個群組，若要選擇群組，記得要將 Line Notify 加到該組群裡面，當兩個欄位都設定後，方可按下底下『發行』按鈕。



4. 此時 LINE 帳號就會收到連動的通知。



5. 不同的權杖名稱，都會隨機產生一組獨一無二的通行識別碼，如左下圖，當按下『複製』按鈕會自動將 token 紀錄起來，記得要立即貼到自訂積木裡面，因為當離開此視窗，將不再提供。如果上述 token 外流，遇到有心人士亂傳一大堆資料，只能回到首頁按『解除』，如右下圖，然後再重新建立新的 token，程式碼也要同步更新，所以盡量注意個人隱私勿外流。

已發行的權杖如下。

MjkjKyDcsN99aj5wKOI1vI8bt2TFIpPWOCwEDV;

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

泰山電子陳致中

## 已連動的服務

此為已連動的服務一覽。若想解除連動，請按「解除」鍵。



2018.10.27 11:37

從 通知測試

傳送至 汐止溫暖的家

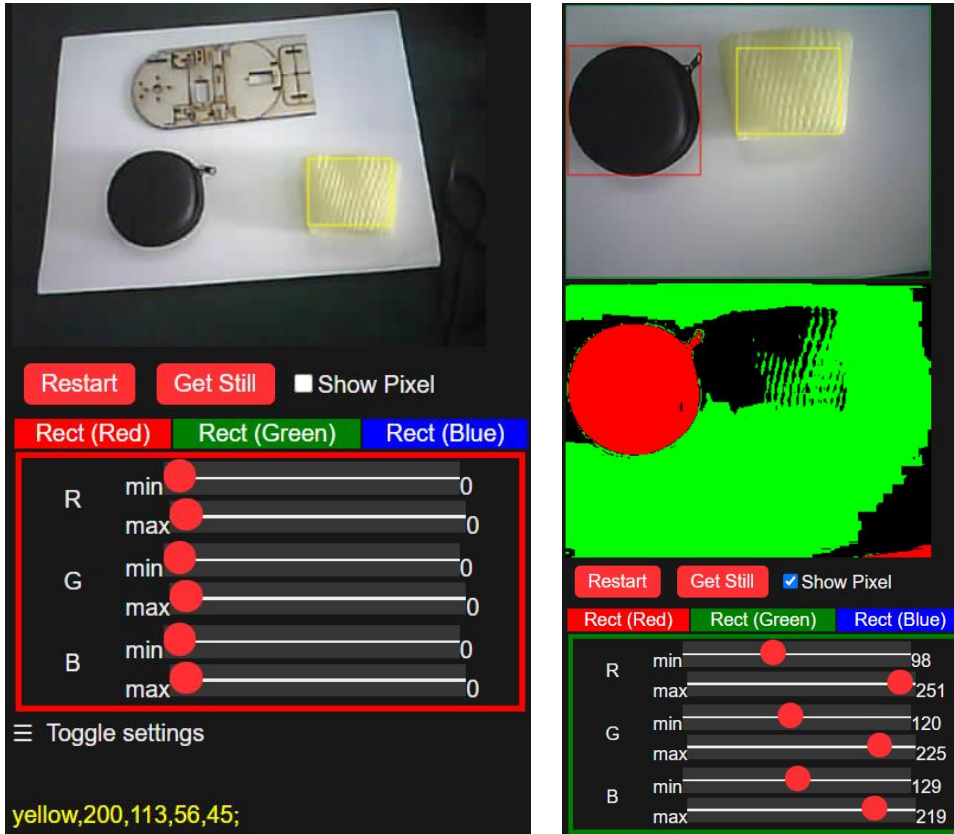
解除

```
27 const char* myDomain = "script.google.com";
28 String myScript = "/macros/s/AKfycb4bzYDTz3_qG7EmWoP874y10zLQsdbBW7uoo-QvhaJDsLoZwQTt/exec";
29 String myLineNotifyToken = "myToken=6go8P0etlOHieRVoywAQN9gBMHTq96k6S4O7mQ961qB"; //Line
30 String myFoldername = "&myFoldername=ESP32-CAM";
31 String myFilename = "&myFilename=ESP32-CAM.jpg";
32 String myImage = "&myFile=";
```

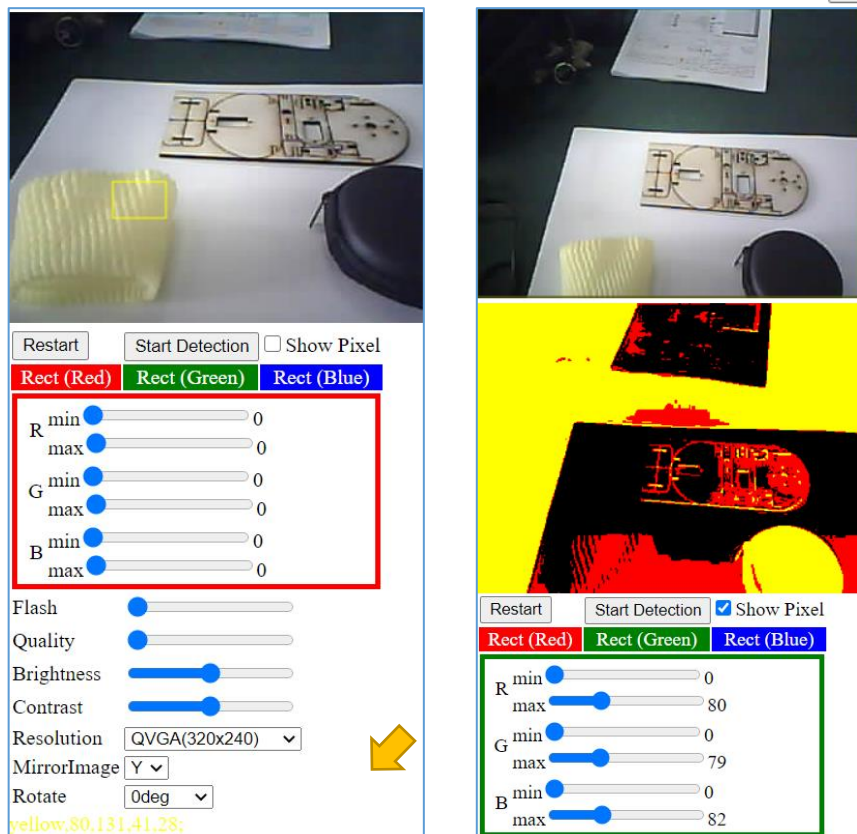


## 實作 10：ESP32-CAM\_Tracking.js\_color 顏色辨識追蹤

1. ESP32-CAM\_Tracking.js\_color.ino 是利用官方預設的串流，顯示較流暢穩定，若有追蹤到物件，會顯示顏色與位置座標(left, top, width, height)於最下方，視窗左上角座標為(0,0)，由於程式內建會抓 magenta 洋紅色、cyan 青色、yellow 黃色 這三種顏色，與自訂範圍無關，所以一開始只找到黃色物件，如左下圖，當勾選『Show Pixel 顯示像素』則增加一個視窗，下方視窗是『僅』顯示 RGB 範圍內的顏色，若在白紙上放一個黑色物件，選擇預設 Black 黑色是 R=0、G=0、B=0，但一定找不到，因為那是純黑色，不可能剛剛好，請修正各顏色 min 最小值與 max 最大值即可，所有顏色的 min  $\neq$  max，方可辨識顏色。



2. ESP32-CAM\_Tracking.js\_color1.ino 是使用張流，需先按下  才會顯示偵測的物件視窗



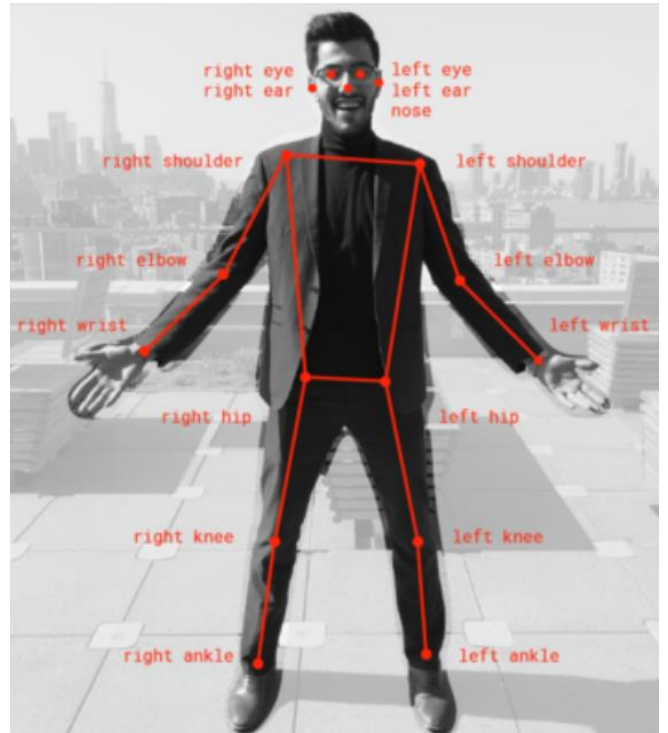
## 實作 11：ESP32-CAM\_MultiPersonPose\_ControlRelay 用肢體動作控制電燈

### 1. 姿態辨識程式撰寫要點：

- A. 若不使用鏡像，視訊前人的右手(left)對應 A I 回傳的右手參數。
- B. 若使用鏡像，視訊前人的右手對應 A I 回傳的左手(right)參數。
- C. 各部位之間的距離會因人與視訊鏡頭間的距離而改變。可利用各部位距離間的比值關係不受距離視訊鏡頭遠近影響來設定判定姿態的條件。
- D. 設定一變數紀錄目前開關狀態做判斷，避免視訊偵測快速重複執行開關而使單晶片當機。
- E. 可設定較高的信心度，避免產生異常的偵測意外開關電器，若無繼電器，可控制 GPIO4 閃光燈。

### 2. 姿態辨識要先學會找到 Nose 鼻子、Eye 眼睛、Ear 耳朵、Shoulder 肩膀、Elbow 手肘、Wrist 手腕、Hip 臀部、Knee 膝蓋、Ankle 腳踝...

### 3. 姿態辨識通靈動作練習：開燈：兩手腕分別靠近兩耳旁(左下圖)、關燈：兩手腕在雙肩下交叉(右下圖)，底下顯示身體各部位位置，也可多人一起玩。



Restart Relay:ON

Flash ☐

≡ Toggle settings

Persons  ScoreLimit

MirrorImage

0,nose,1,146,120

0,leftEye,1,156,107

0,rightEye,1,132,110

0,leftEar,0.88,171,111

0,rightEar,0.94,113,122

0,leftShoulder,0.96,207,173

0,rightShoulder,0.97,88,179

0,leftElbow,0.94,286,177

0,rightElbow,0.68,2,179

0,leftWrist,0.45,208,115

0,rightWrist,0.73,61,128

0,leftHip,0.06,217,280

0,rightHip,0.04,98,287

0,leftKnee,0.02,223,279

0,rightKnee,0.01,100,281

0,leftAnkle,0.03,218,278

0,rightAnkle,0.97,282

Restart Relay:OFF

Flash ☐

≡ Toggle settings

Persons  ScoreLimit

MirrorImage

0,nose,1,173,96

0,leftEye,1,184,86

0,rightEye,1,162,85

0,leftEar,0.79,195,97

0,rightEar,0.95,143,96

0,leftShoulder,0.99,217,159

0,rightShoulder,0.99,126,161

0,leftElbow,0.96,261,238

0,rightElbow,0.73,87,230

0,leftWrist,0.86,209,203

0,rightWrist,0.94,138,203

0,leftHip,0.1,203,271

0,rightHip,0.17,137,267

0,leftKnee,0.05,203,271

0,rightKnee,0.01,144,263

0,leftAnkle,0.04,200,277

0,rightAnkle,0.02,141,275

Restart Relay:OFF

Flash ☐

≡ Toggle settings

Persons  ScoreLimit

MirrorImage

Please wait for loading model.



4. 第 1322 行『/control?flash=10』是指閃光燈亮度 10(※最大 255 易燒毀)，若想設計動作，請先觀察人體各部位相對關係，如第 1319 與 1324 行的判斷句，『&&』是而且 AND 的意思。

```
1315 //可以在這區塊做姿態判定控制開關或繼電器
1316 if (rightWristLeft>=0&&rightWristTop>=0&&leftWristLeft>=0&&leftWristTop>=0&&rightShoulderLeft>=0&&rightShoulderTop>=0&&leftShoulderLeft>=0&&leftShoulderTop>=0) {
1317     if (mirrorimage==1||mirrorimage==0) { //通靈姿態左右位置條件相同，因此使用鏡像與否皆相同
1318
1319         if (((rightWristLeft-rightEarLeft)/(rightEarLeft-noseLeft)>=0.5) && ((rightWristLeft-rightEarLeft)/(rightEarLeft-noseLeft)<=3) && ((leftWristLeft-rightEarLeft)/(leftEarLeft-noseLeft)>=0.5) && ((leftWristLeft-rightEarLeft)/(leftEarLeft-noseLeft)<=3) && (rightWristTop > rightShoulderTop && leftWristTop > leftShoulderTop)) {
1320             if (document.getElementById("state").innerHTML == "OFF") {
1321                 document.getElementById("state").innerHTML = "ON";
1322                 $.ajax({url: document.location.origin+"/control?flash=10", async: false}); //控制閃光燈來代表繼電器
1323             }
1324         } else if (Math.abs(rightWristLeft - leftWristLeft) * 2 <= Math.abs(rightShoulderLeft - leftShoulderLeft) && (rightWristTop > rightShoulderTop && leftWristTop > leftShoulderTop)) {
1325             if (document.getElementById("state").innerHTML == "ON") {
1326                 document.getElementById("state").innerHTML = "OFF";
1327                 $.ajax({url: document.location.origin+"/control?flash=0", async: false}); //控制閃光燈來代表繼電器
1328             }
1329         }
1330     }
1331 }
1332 }
```

若要改用繼電器控制，將第 1322 行與第 1327 行改為 HIGH 與 LOW  
pinMode(XX, OUTPUT);  
digitalWrite(XX, LOW);

5. ESP32-CAM\_MultiPersonPose\_ControlRelay02.ino 則是控制開關動作不同，打開是『拉弓動作，右手腕在肩膀右上，左手腕在左胸，兩手腕連線傾斜角度 20~60 度』，關閉是『兩手腕垂直高舉』，請自行比較各種姿態判斷語法的撰寫。

```
1313 //可以在這區塊做姿態判定控制開關或繼電器
1314 if (leftShoulderTop>=0&&rightShoulderTop>=0&&rightWristLeft>=0&&rightWristTop>=0&&leftWristLeft>=0&&leftWristTop>=0&&noseLeft>=0&&noseTop>=0) {
1315     //當控制的姿態是左右不對稱時，鏡像與非鏡像判別條件不同
1316
1317     if (mirrorimage==1) { //鏡像時
1318         if (rightWristLeft>noseLeft&&rightWristTop<noseTop&&(leftWristTop-leftShoulderTop)/(leftShoulderTop-noseTop)>=-0.5&&(leftWristTop-leftShoulderTop)/(leftShoulderTop-noseTop)<=3) {
1319             if (document.getElementById("state").innerHTML == "OFF") {
1320                 document.getElementById("state").innerHTML = "ON";
1321                 document.getElementById("ifr").src = document.location.origin+"/control?flash=10";
1322             }
1323         }
1324     } else if (rightWristTop<noseTop&&leftWristTop<noseTop) {
1325         if (document.getElementById("state").innerHTML == "ON") {
1326             document.getElementById("state").innerHTML = "OFF";
1327             document.getElementById("ifr").src = document.location.origin+"/control?flash=0"; //控制閃光燈來代表繼電器
1328         }
1329     }
1330 }
```

## 實作 12 : ESP32-CAM\_teachablemachine.ino 機械學習

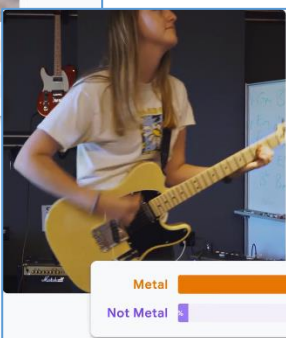
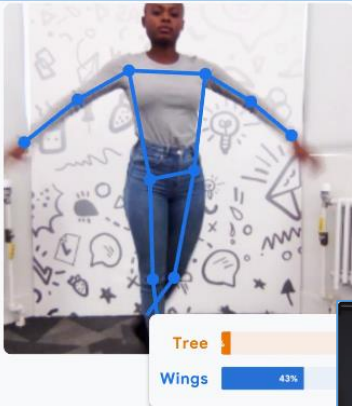
- 影片介紹(英文版) <https://www.youtube.com/watch?v=T2qQGqZxkD0>  
法蘭斯教學介紹 <https://www.youtube.com/watch?v=Sn7AwpGFXLw>
- 官方功能介紹 <https://teachablemachine.withgoogle.com/>，可機器學習再辨識影像、聲音、姿態，例如下圖 wing 翅膀姿態的機率有 43%，Metal 重金屬搖滾機率有 93%。

# Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

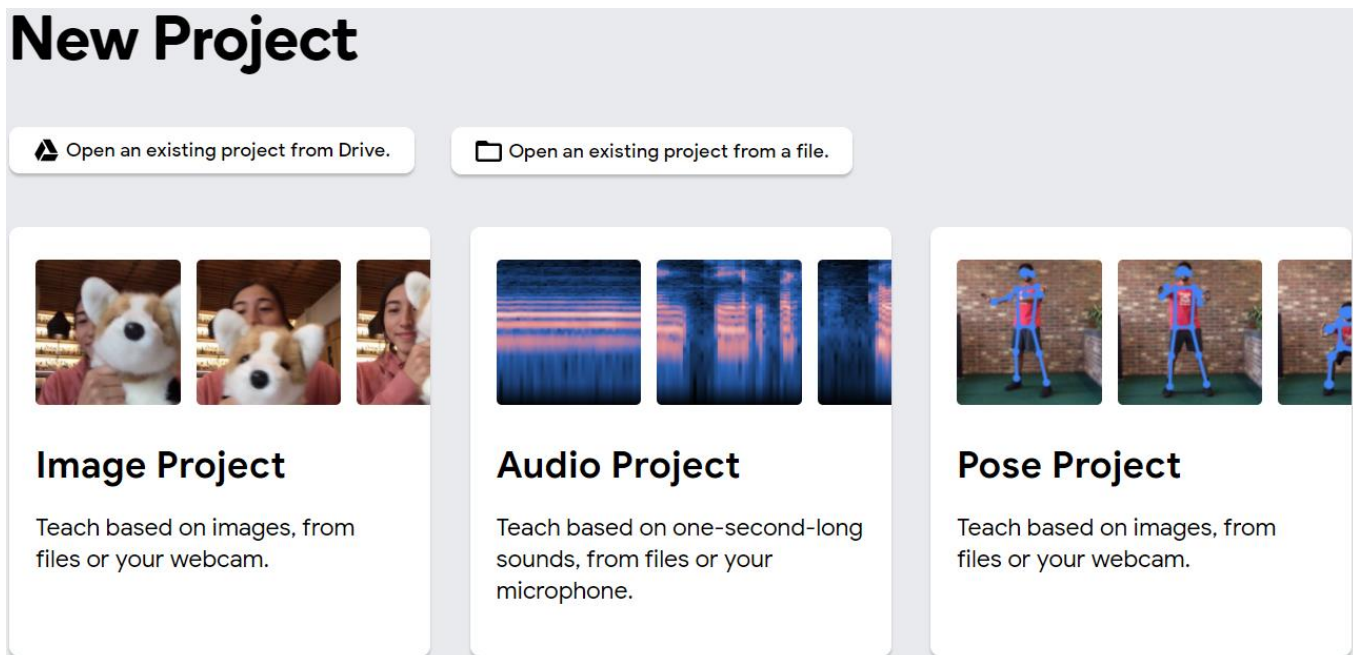
A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

[Get Started](#)

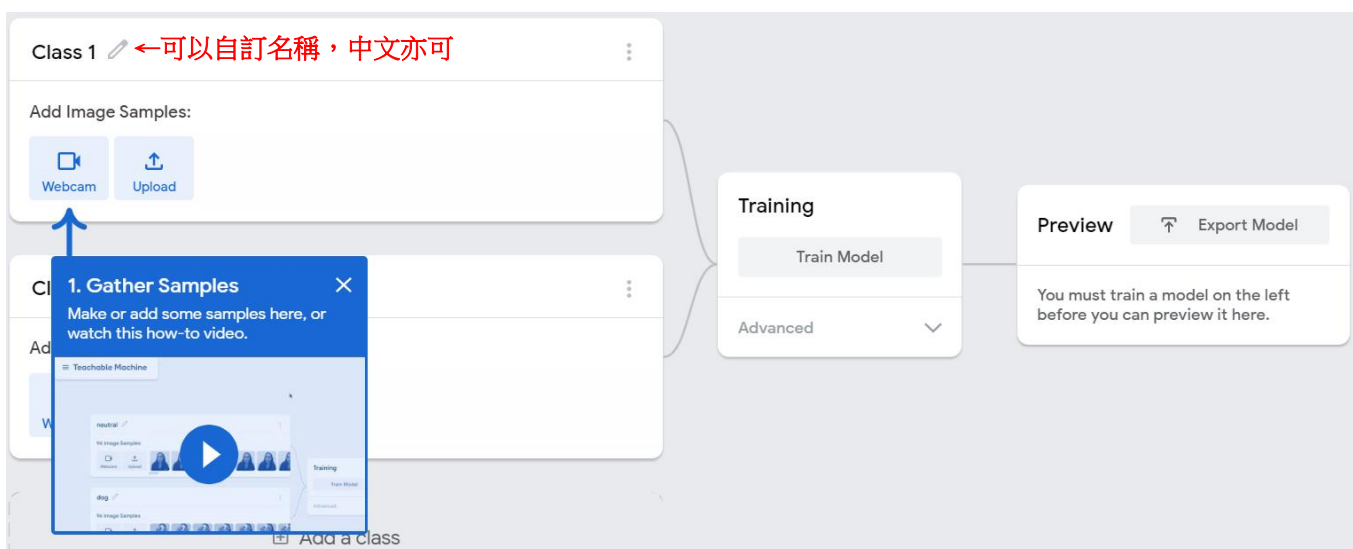




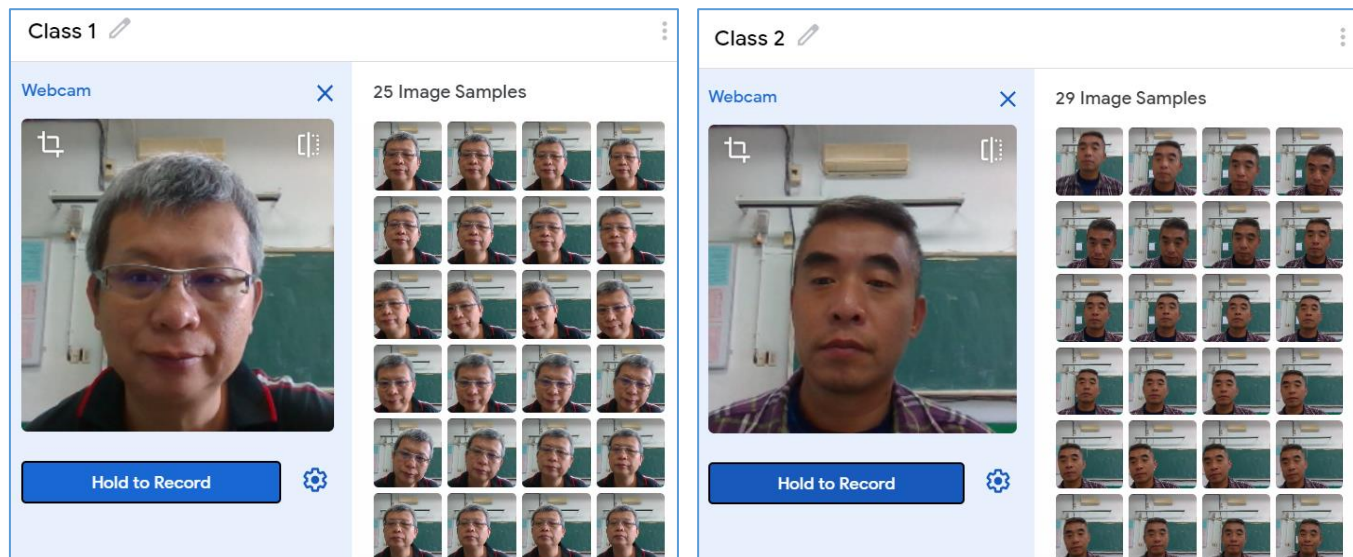
3. 按下 **Get Started** 就可以線上開始訓練，Image Project 是訓練圖像、Audio 是聲音、Pose 是姿態。



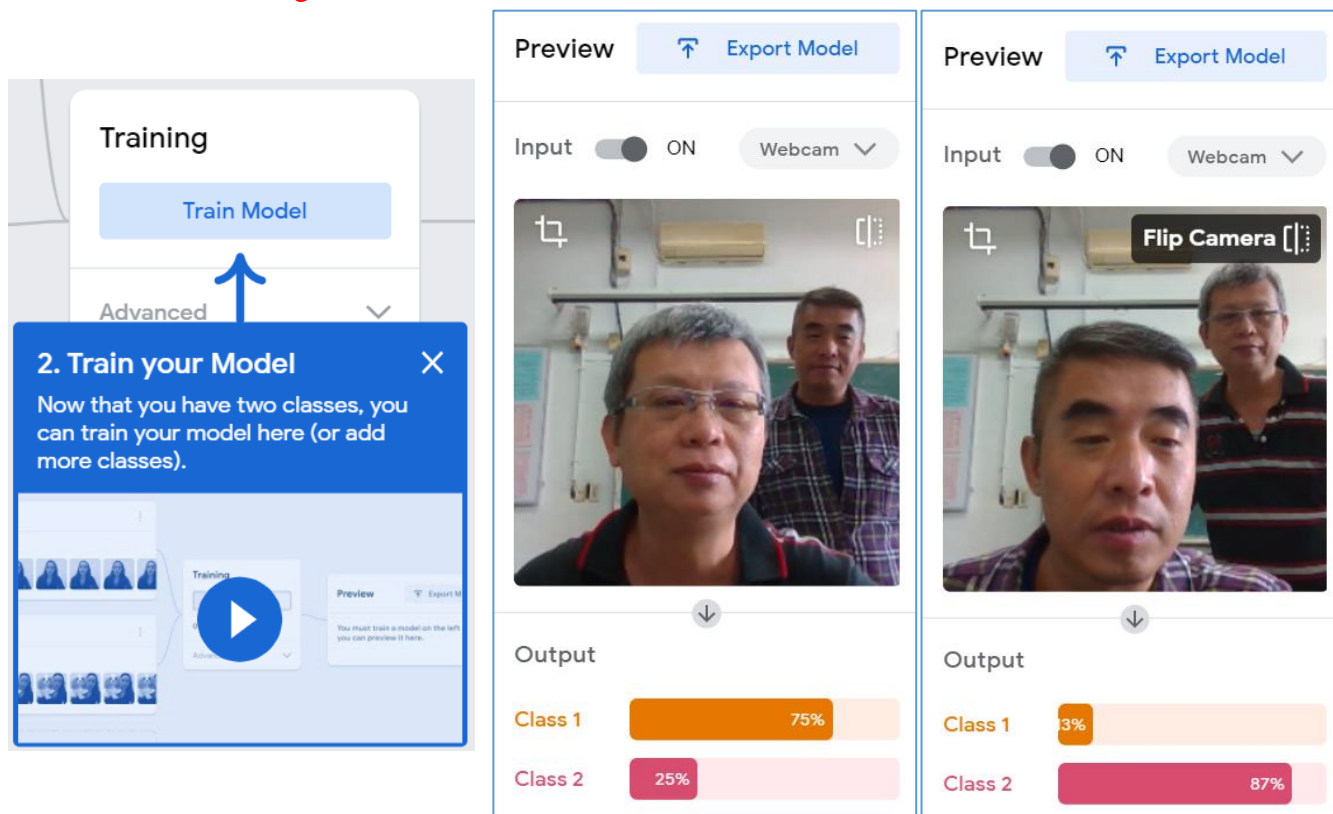
4. 若電腦有視訊鏡頭方可按『Webcam』直接拍照，若沒有可到自己 FB 或其他方式找自拍照，再用『Upload 上傳』。



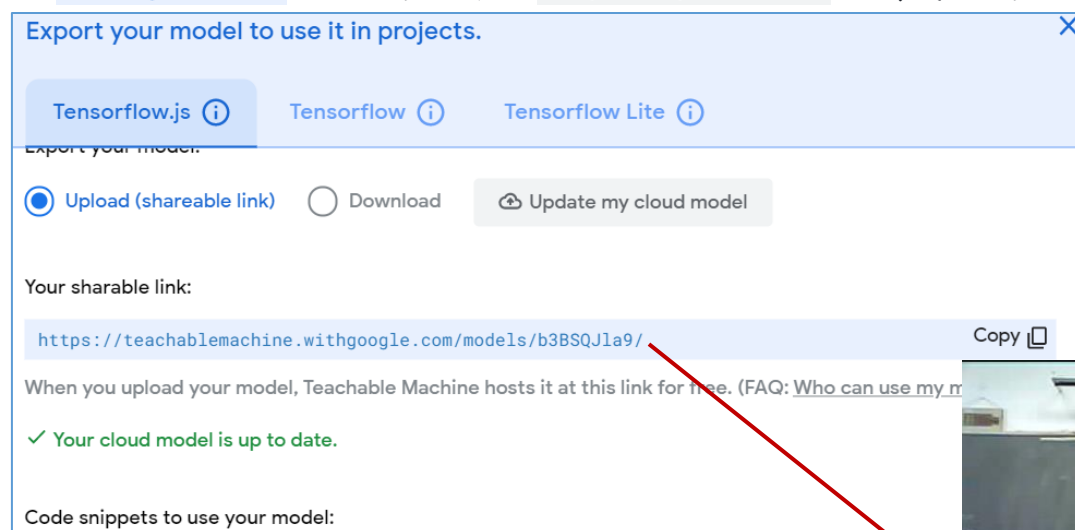
5. 步驟一按著『Hold to Record』不放會自動連續拍照，選擇類別 Class2 繼續，至少兩位才能比較。



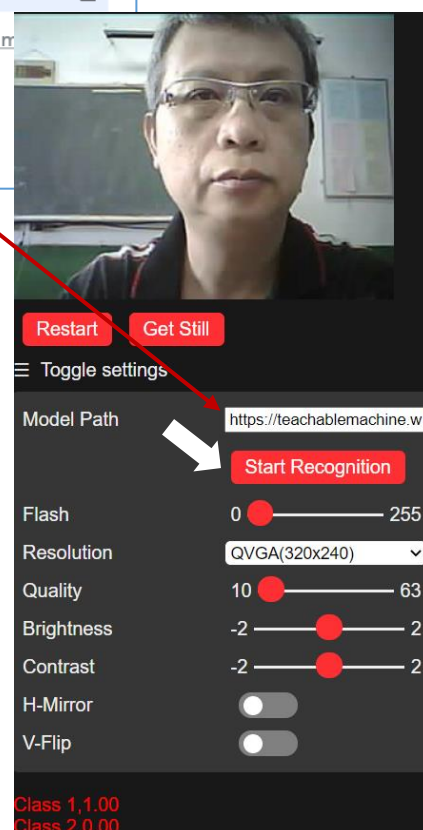
6. 步驟二按下『**Training**』開始訓練，訓練好就自動進入步驟三，直接辨識比較誰的機率高。



7. 按 **Export Model** 匯出模型，再按 **Update my cloud model** 上傳我的模型，按『**Copy**』複製連結。

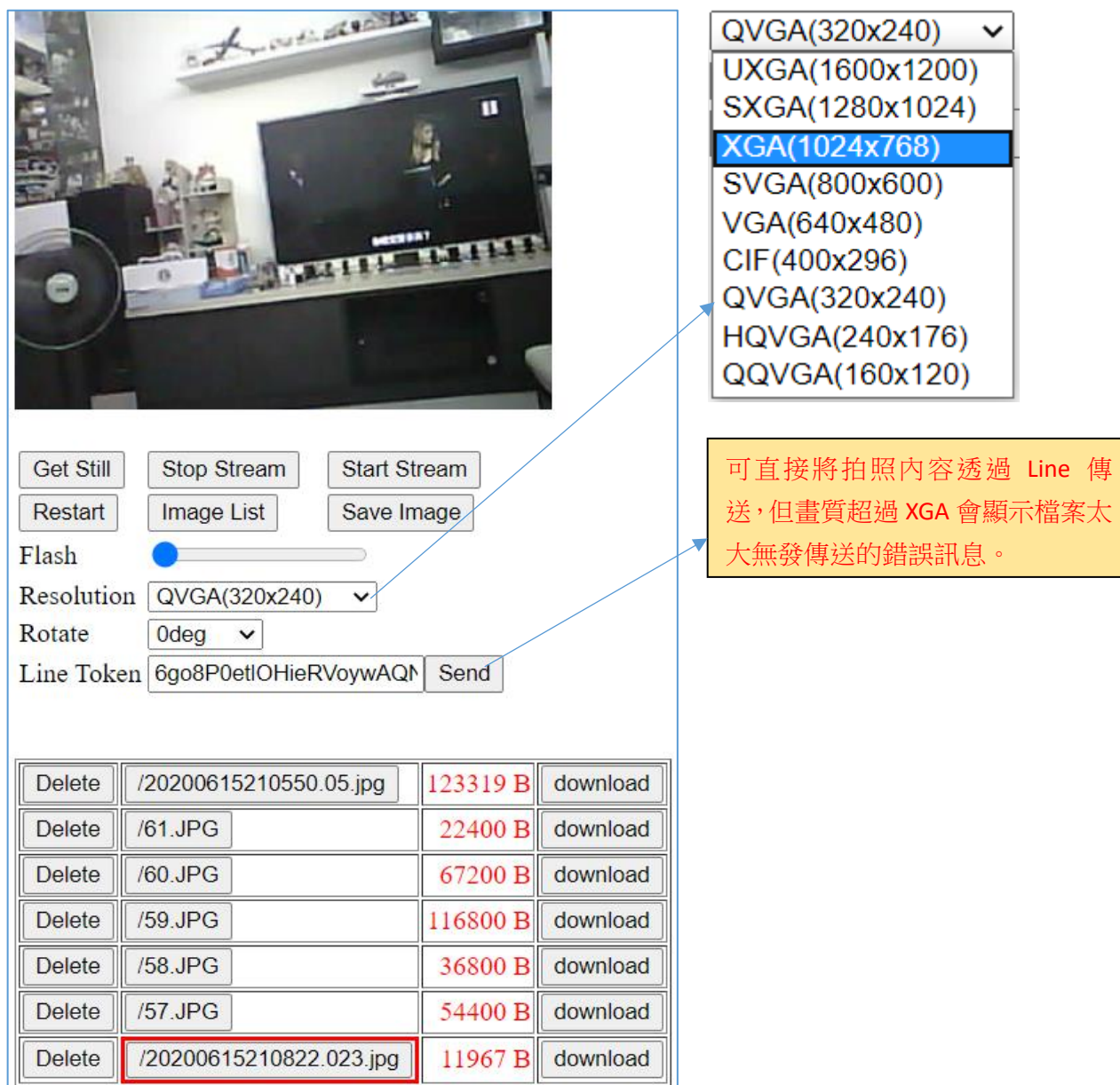


8. 開啟瀏覽器，需先輸入上面步驟複製的網址  
<https://teachablemachine.withgoogle.com/models/b3BSQJla9/>於 Model Path  
模型路徑文字方塊內，然後按下 **Start Recognition** 開始辨識，等待  
模型下載好才會看到視訊，下方則會出現機率結果。  
註：複製網址最後若為[...], 是因為還沒有按 **Update my cloud model**  
上傳到我的雲端模型。
9. 所訓練的模型是一直存在雲端，只要知道最後的代碼即可。
10. 請嘗試練習『剪刀、石頭、布』或姿態動作的識別練習。
11. 可使用 ESP32-CAM 畫面截圖，先安裝實作 02:ESP32-CAM\_http.ino  
，用『<http://192.168.xxx.xxx/capture>』網址截圖，或點 get-still 按鈕  
截圖，再另存新檔，訓練的照片越多越準確，然後回到步驟 4『**Upload**  
上傳』到這網站上訓練。



## 實作 13 : ESP32-CAM\_SD\_Manager\_PAGE.ino

1. 按下『Start Stream』可看到鏡頭內容，手動按『Save Image』將照片儲存到 SD，限 **FAT32** 格式，不支援 NTFS 格式，當未插入 SD 會一直重新啟動偵測。



The screenshot displays the ESP32-CAM web interface. At the top left is a live camera feed showing a desk with a monitor and various items. Below the feed are control buttons: 'Get Still', 'Stop Stream', 'Start Stream', 'Restart', 'Image List', and 'Save Image'. A 'Flash' slider is set to a low level. The 'Resolution' dropdown menu is open, showing a list of resolutions: QVGA(320x240), UXGA(1600x1200), SXGA(1280x1024), **XGA(1024x768)** (highlighted), SVGA(800x600), VGA(640x480), CIF(400x296), QVGA(320x240), HQVGA(240x176), and QQVGA(160x120). Below the resolution menu is a 'Rotate' dropdown set to '0deg'. The 'Line Token' field contains '6go8P0etlOHieRVoywAQN' and a 'Send' button. At the bottom is a table listing saved images on the SD card.

Delete	/20200615210550.05.jpg	123319 B	download
Delete	/61.JPG	22400 B	download
Delete	/60.JPG	67200 B	download
Delete	/59.JPG	116800 B	download
Delete	/58.JPG	36800 B	download
Delete	/57.JPG	54400 B	download
Delete	/20200615210822.023.jpg	11967 B	download

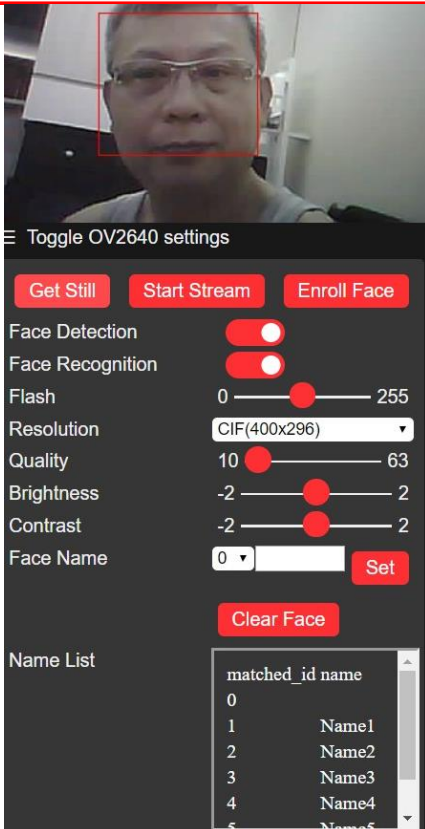
可直接將拍照內容透過 Line 傳送，但畫質超過 XGA 會顯示檔案太大無發傳送的錯誤訊息。



## 實作 14：ESP32-CAM 讀取 SD 卡照片註冊人臉做辨識

1. ESP32 需為 1.0.4 以上版本，否則編譯會錯，ESP32-CAM\_FaceRecognition\_SD\_EnrollFace\_PAGE.ino 為網頁版，ESP32-CAM\_FaceRecognition\_SD\_EnrollFace.ino 為離線版。
2. 高雄法蘭斯老師範例下載網址 <https://github.com/fustyles/Arduino>，影片教學 <https://www.youtube.com/watch?v=BDQxH3KYNpc&lc=UgxyUjfuzOSulU6RA4h4AaABAg.98NZPXJIVuV98NnIoHu4yx>
3. 第 34 行『#define ENROLL\_CONFIRM\_TIMES 5』是指人臉辨識同一人臉註冊影像數訓練，而第 43 行『#define FACE\_ID\_SAVE\_NUMBER 7』是指人臉辨識註冊人的臉數，所以  $5 \times 7 = 35$ ，按『Get Still 取得靜態照片』（先關閉臉部偵測），再以手動方式另存 1~35.JPG 檔案到 SD 根目錄，為 CIF 解析度的人臉照程式會自動切割，若改了後 String filename[5]、String recognize\_face\_matched\_name[7] 這兩行內容也要跟著改，然後在 Arduino IDE 序列埠監看視窗觀看確認每張影像都能辨識到人臉！

```
23 //輸入WiFi連線帳號密碼
24 const char* ssid = "xxxxx";
25 const char* password = "xxxxx";
26
27 //輸入AP端連線帳號密碼
28 const char* apssid = "ESP32-CAM";
29 const char* appassword = "12345678"; //AP密碼至少要8個字元以上
30
31 boolean streamState = false;
32
33 //人臉辨識同一人臉註冊影像數
34 #define ENROLL_CONFIRM_TIMES 5
35 //Save the captured still (FRAMESIZE_CIF) with your face to the SD card.
36 //Filename: 1.jpg, 2.jpg, 3.jpg, 4.jpg, 5.jpg
37 String filename[5] = {"1.jpg", "2.jpg", "3.jpg", "4.jpg", "5.jpg"}; //1.jpg, 2.jpg, ..., 35.jpg
38 int image_width = 400;
39 int image_height = 296;
40 //UXGA(1600x1200), SXGA(1280x1024), XGA(1024x768), SVGA(800x600), VGA(640x480), CIF(400x296), QVGA(320x240)
41
42 //人臉辨識註冊人為臉數
43 #define FACE_ID_SAVE_NUMBER 7
44 //設定人臉辨識顯示的人名
45 String recognize_face_matched_name[7] = {"Name0", "Name1", "Name2", "Name3", "Name4", "Name5", "Name6"};
```



```
1669 void FaceMatched(int faceid) { //辨識到註冊人臉執行指令控制
1670     if (faceid==0) {
1671         pinMode(p, OUTPUT);
1672         digitalWrite(p, 1);
1673         delay(10000);
1674         digitalWrite(p, 0);
1675     }
1676     else if (faceid==1) {
1677     }
1678     else if (faceid==2) {
1679     }
1680     else if (faceid==3) {
1681     }
1682     else if (faceid==4) {
1683     }
1684     else if (faceid==5) {
1685     }
1686     else if (faceid==6) {
1687     }
1688     else {
1689     }
1690 }
1691
1692 void FaceNoMatched() { //辨識為陌生人臉執行指令控制
1693
1694 }
```

註：如果只要是名單內的人，都可以讓某一腳位高電位 10 秒後變回低電位，就可以真正當門禁系統。

## 實作 15 : ESP32-CAM\_coco-ssd\_PeopleTracking.ino 雙雲台追蹤物件

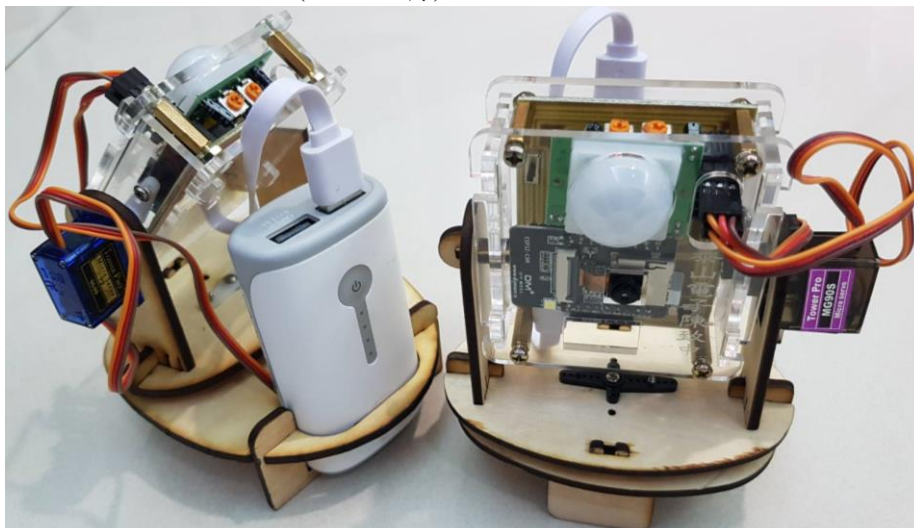
1. **COCO** (Common Object in Context) Dataset 資料庫是 Microsoft 所提供的是一個完整龐大的開源影像資料庫模型；而 **SSD** 演算法全名是 Single Shot MultiBox Detector (單一鏡頭多框偵測)。

Dataset	Category	Image quantity	BBox quantity
PASCAL VOC (07++12)	20	21,503	62,199
MSCOCO (2014trainval)	80	123,287	886,266
ImageNet Det (2017train)	200	349,319	478,806

Dataset	Total	Small	Middle	Large
PASCAL VOC (07++12)	62,199	6,983(11.2%)	19,677(31.6%)	35,539(67.2%)
MSCOCO (2014trainval)	886,266	278,651(31.4%)	311,999(35.2%)	295,616(33.4%)
ImageNet Det (2017train)	478,806	22,677(4.7%)	86,439(18.1%)	369,690(77.2%)



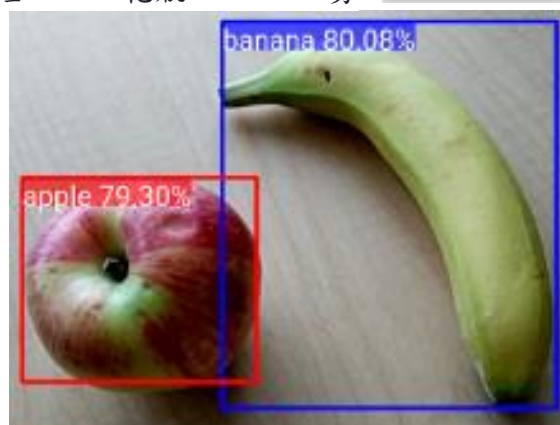
2. 利用兩顆不連續 SERVO 做影像追蹤，SERVO 建議使用 GPIO2 與 GPIO13，其中 SERVO1(水平旋轉)接 GPIO2，SERVO2 (垂直旋轉)接 GPIO13。



3. Tensorflow 的 ImageAI 提供 Video object detection 使用模型為 RetinaNet，可偵測 80 種物件類別如下，還可順便學習英語單字：

person 人、bicycle 自行車、car 汽車、motorcycle 摩托車、airplane 飛機、bus 公共汽車、train 火車、truck 卡車、boat 船、traffic light 交通號誌、fire hydrant 消火栓、stop\_sign 停止標誌、parking meter 停車收費表、bench 長凳、bird 鳥、cat 貓、dog 狗、horse 馬、sheep 羊、cow 牛、elephant 大象、bear 熊、zebra 斑馬、giraffe 長頸鹿、backpack 背包、umbrella 雨傘、handbag 手提袋、tie 領帶、suitcase 手提箱、frisbee 飛盤、skis 滑雪板、snowboard 滑雪板、sports ball 運動球、kite 風箏、baseball bat 棒球棍、baseball glove 棒球手套、skateboard 滑板、surfboard 衝浪板、tennis racket 網球拍、bottle 瓶、wine glass 酒杯、cup 杯、fork 叉、knife 刀、spoon 勺、bowl 碗、banana 香蕉、apple 蘋果、sandwich 三明治、orange 橘子、broccoli 西蘭花、carrot 胡蘿蔔、hot dog 熱狗、pizza 比薩、donut 甜甜圈、cake 蛋糕、chair 椅子、couch 沙發、potted plant 盆栽、bed 床、dining table 餐桌、toilet 廁所、tv 電視、laptop 筆電、mouse 滑鼠、remote 遙控器、keyboard 鍵盤、cell phone 手機、microwave 微波爐、oven 烤箱、toaster 烤麵包機、sink 水槽、refrigerator 冰箱、book 書、clock 時鐘、vase 花瓶、scissors 剪刀、teddy bear 泰迪熊、hair dryer 吹風機、toothbrush 牙刷。關於 Tensorflow 物件(object)辨識偵測說明 [https://www.tensorflow.org/lite/models/object\\_detection/overview](https://www.tensorflow.org/lite/models/object_detection/overview)

person  
dining table  
toilet  
tv  
laptop  
mouse  
remote  
keyboard  
cell phone  
microwave  
oven  
toaster  
sink  
refrigerator  
book  
clock  
vase  
scissors  
teddy bear  
hair drier  
toothbrush









8. 第 217-226 行是兩顆 SERVO 初始狀態，統一定位在 90 度，第 220 行的 `angle1Value1` 變數是在第 1011 行設為 4850，相當於中間 90 度，因為 SERVO 控制是在 1700-8000 之間。
9. 由於 ESP32 CAM 無類比輸出指令，若要使用 PWM 類比輸出控制，請詳閱下面說明。
10. 第 218 行的『`ledcAttachPin(2, 5);`』，第一個參數 2 代表 GPIO2 腳位，第二個參數 5 代表通道編號，6 是 Servo2、4 是內建高亮度 LED，通道不可重複使用。
11. 第 219 行的『`ledcSetup(5, 50, 16);`』，第一個參數 5 是通道編號，第二個參數 50 是頻率，傳輸影像不宜更新過快，第三個參數代表解析度分為  $2^{16}$  等份。
12. 第 220 行相當於『`ledcWrite(11, 4850);`』是將值寫入對應的通道。
13. 第 221 行是轉動後延遲 1 秒，若省略，會明顯抖動。(註：建議底座加重、加大與止滑，可避免晃動)
14. 下面程式碼是用來追蹤物件，以便控制 SERVO 轉向，若垂直馬達是裝在左側，則第 1105、1107 與 1115、1117 的加減需相反，經過計算之後，然後將修正後的角度透過網頁進行控制。

```

217 //Servo
218 ledcAttachPin(2, 5);
219 ledcSetup(5, 50, 16);
220 ledcWrite(5, angle1Value1); //90度
221 delay(1000);
222
223 ledcAttachPin(13, 6);
224 ledcSetup(6, 50, 16);
225 ledcWrite(6, angle1Value2); //90度
226 delay(1000);
227
228 //閃光燈
229 ledcAttachPin(4, 4);
230 ledcSetup(4, 5000, 8);

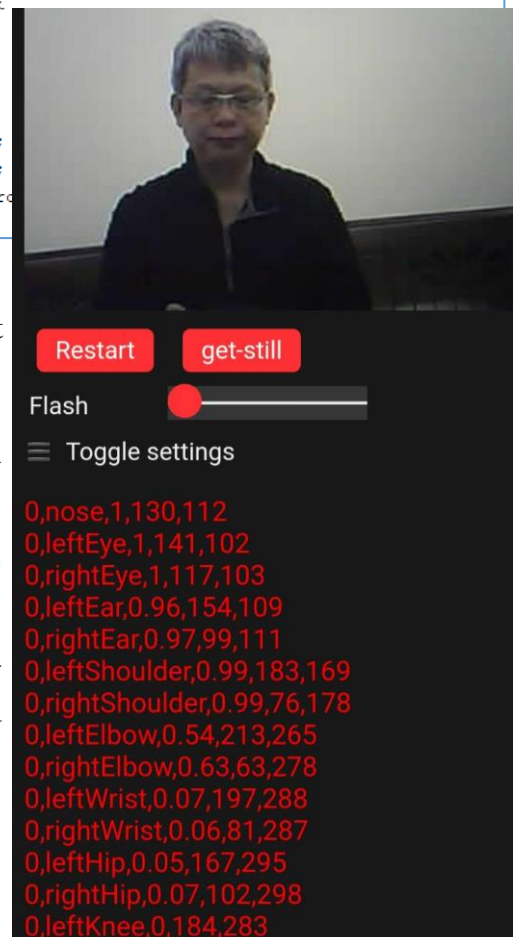
```

```

1102 var midY = Math.round(y)+Math.round(height)/2; //取偵測到第一人的區域垂直中點Y值
1103 if (midY > y_Bottom) { //畫面中心自訂小區域中即使偏下不轉動
1104     if (midY > (y_Bottom+imageHeight)/2) { //區分距離畫面中點偏下程度
1105         angle1Value2+=300;
1106     } else {
1107         angle1Value2+=150;
1108     }
1109     if (angle1Value2 > 7650) angle1Value2 = 7650;
1110     if (angle1Value2 < 2050) angle1Value2 = 2050;
1111     $.ajax({url: document.location.origin+'/control?servo2='+angle1Value2, async: false});
1112 }
1113 else if (midY < y_Top) { //畫面中心自訂小區域中即使偏上不轉動
1114     if (midY < y_Top/2) { //區分距離畫面中點偏上程度
1115         angle1Value2-=300;
1116     } else {
1117         angle1Value2-=150;
1118     }
1119     if (angle1Value2 > 7650) angle1Value2 = 7650;
1120     if (angle1Value2 < 2050) angle1Value2 = 2050;
1121     $.ajax({url: document.location.origin+'/contro
1122 }

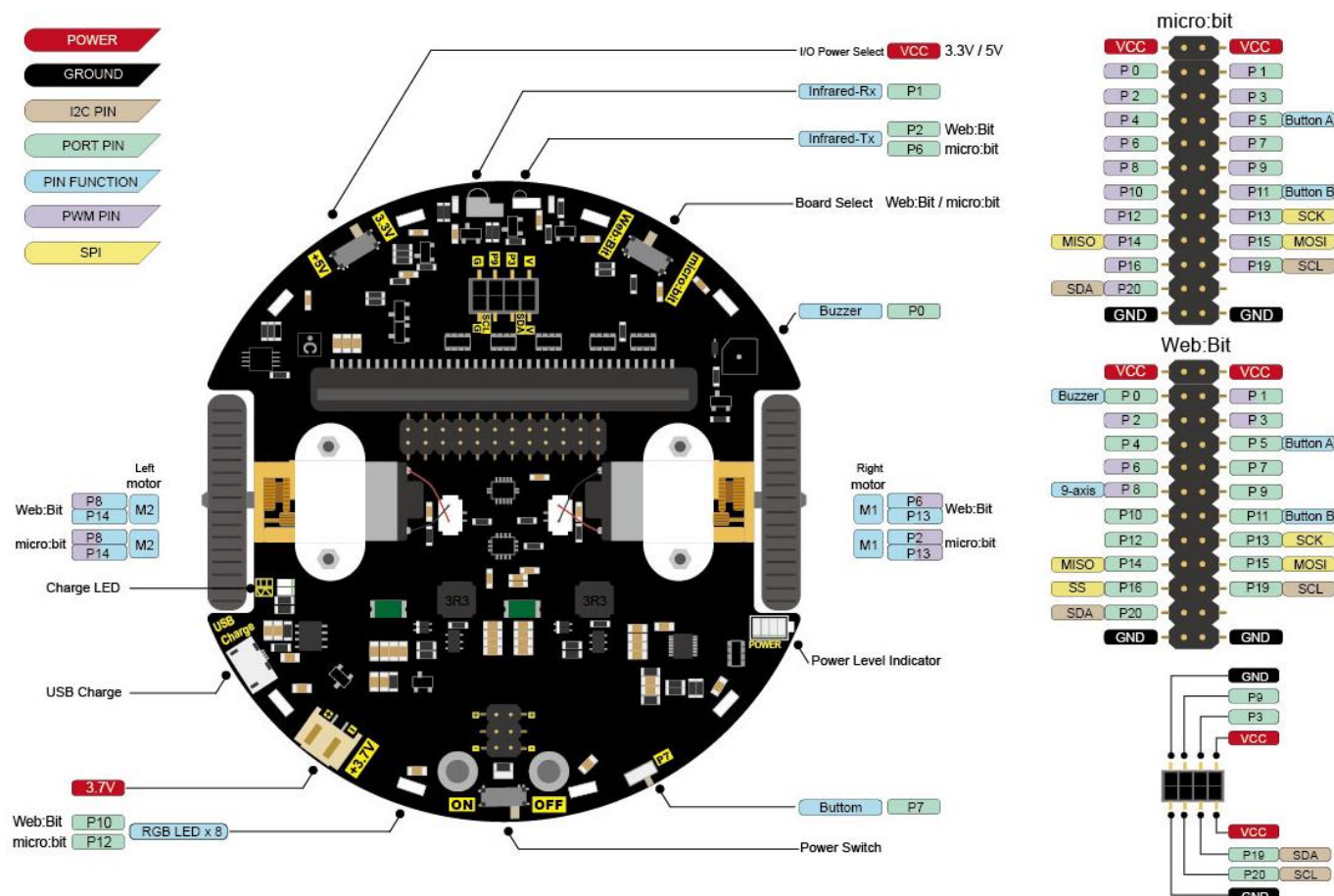
```

15. 上述實作是自動追蹤指定的物件，而 `ESP32-CAM_MultiPersonPose_FaceTracking.ino` 是利用 `posenet` 追蹤人的鼻子座標，『追蹤人臉』位置，硬體連接方式不動，網頁執行畫面如右圖，臉部不會出現框框，但鏡頭會跟著臉部的鼻子移動，底部會顯示身體各特徵的位置，0 代表是第一位人員。
16. 若用手機當 AP 基地台，亦可直接利用手機進行測試，因為都在同區段，建議使用 Chrome 瀏覽器。
17. 若底部一直沒顯示『Please wait for loading model 載入模型請稍後』或等很久還是未出現視窗，應是無線網路的『上傳』速度太過於龜速造成。
18. 當外接電源的電流不夠時，在啟動時，SERVO 會不停顫抖，請更換較大的電流電源。



## 實作 16 : ESP32-CAM\_CAR\_2pwm\_coco-ssd\_TrackingCar.ino 追蹤物件車

- 利用登月小車進行物件追蹤，水平 SERVO 省略，因為可利用輪子左右轉彎，外觀與接腳電路如下：

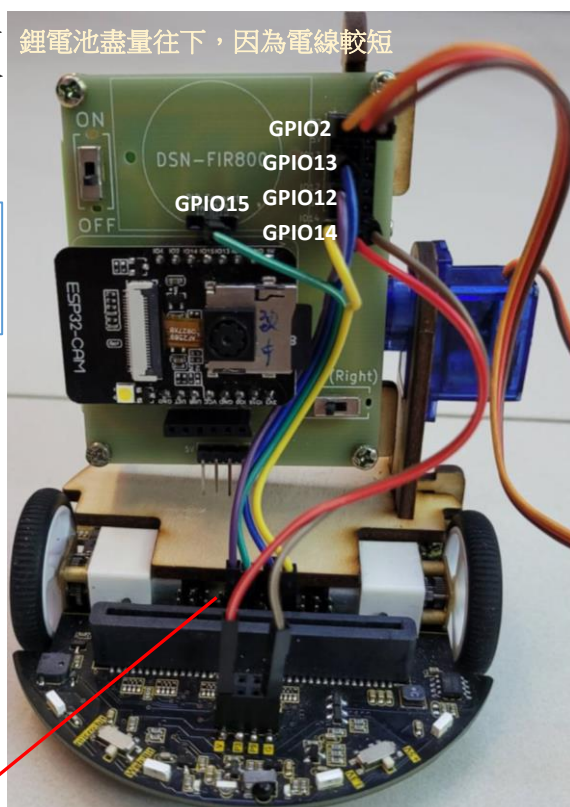
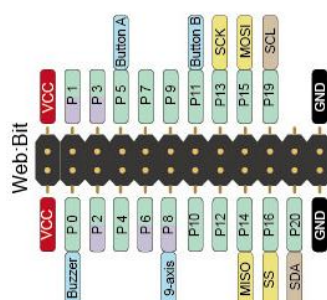


- 透過 10cm 公對公杜邦線將下面的硬體腳位連接，5V 與 GND 可接原本超音波的電源，原本 PIR 需先取下，將登月小車的 pin8 連接中間 GPIO15 的腳位，啟動時要小心，因為在網路尚未連接前，右輪會短暫的轉動，小心夾手。

```

59 int speedR = 160; //紀錄右輪初始轉速 (gpio12, gpio13)
60 int speedL = 160; //紀錄左輪初始轉速 (gpio14, gpio15)
61 double decelerate = 60; //紀錄轉彎減速為 原速*百分比
62 double servoPin = 2; //Servo Pin
    
```

ESP32 CAM	BBC Micro:bit	Web:bit
GPIO12	2	6
GPIO13	13	13
GPIO15	8	8
GPIO14	14	14
GPIO2	垂直 SERVER	
GPIO4	高亮度閃光燈	





```

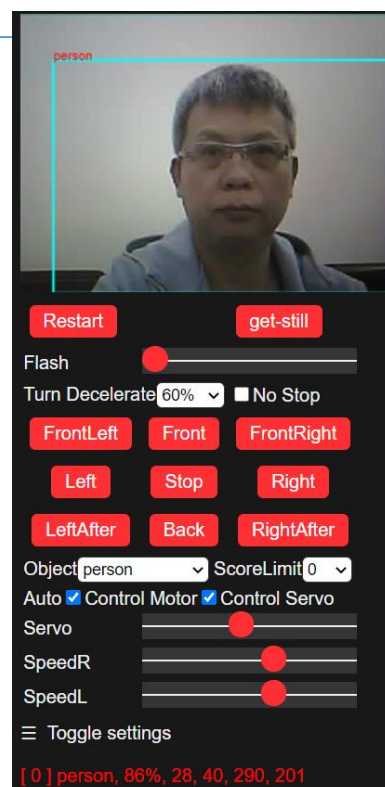
1233 var midY = Math.round(y)+Math.round(height)/2; //第一個偵測物件中心點y值
1234 if (servoState.checked) {
1235     if (midY>(240/2+30)) {
1236         if (midY>195) { //物件中心點偏下程度
1237             angle1Value+=5;
1238         } else {
1239             angle1Value+=3;
1240         }
1241         if (angle1Value > 90) angle1Value2 = 0;
1242         if (angle1Value < 4850) angle1Value2 = 4850;
1243         $.ajax({url: document.location.origin+'/control?servo1='+angle1Value+';stop', async: false});
1244     }
1245     else if (midY<(240/2)-30) {
1246         if (midY<45) { //物件中心點偏上程度
1247             angle1Value-=5;
1248         } else {
1249             angle1Value-=3;
1250         }
1251         if (angle1Value > 90) angle1Value2 = 90;
1252         if (angle1Value < 0) angle1Value2 = 0;
1253         $.ajax({url: document.location.origin+'/control?servo1='+angle1Value+';stop', async: false});
1254     }
}

1207 var midX = Math.round(x)+Math.round(width)/2; //第一個偵測物件中心點x值
1208 if (motorState.checked) {
1209     if (midX<(320/2)-40) {
1210         if (midX<60) //物件中心點偏左程度
1211             delay=200;
1212         else
1213             delay=150;
1214         $.ajax({url: document.location.origin+'/control?car=7;'+delay, async: false}); //左前進
1215         lastValue = 3;
1216     }
1217     else if (midX>(320/2+40)) {
1218         var delay=0;
1219         if (midX>260) //物件中心點偏右程度
1220             delay=200;
1221         else
1222             delay=150;
1223         $.ajax({url: document.location.origin+'/control?car=6;'+delay, async: false}); //右前進
1224         lastValue = 3;
1225     }
1226     else if (midX>=(320/2-40)&&midX<=(320/2+40)) { //物件中心點在正中心自訂區域120~200中則前進
1227         if (lastValue!=1)
1228             $.ajax({url: document.location.origin+'/control?car=1;;stop', async: false});
1229         lastValue = 1;
1230     }
}

```

-30~+30 是表示若在中央區域，則保持往前，除非偏差很大，以避免左右晃動

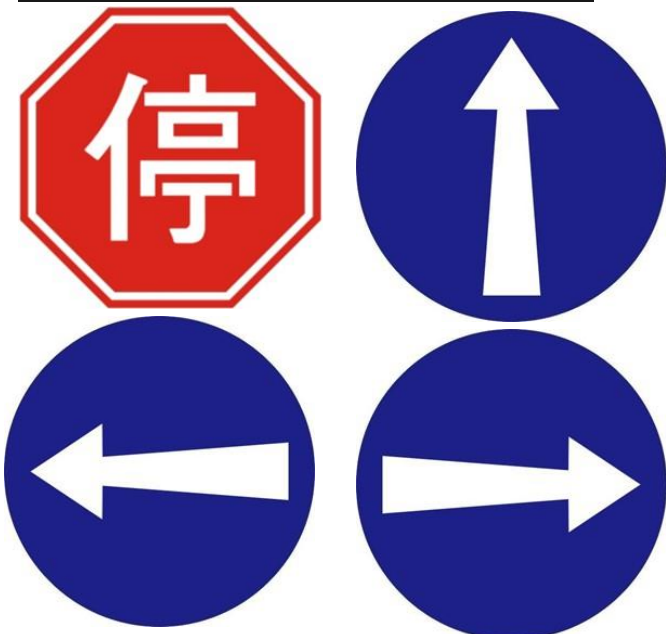
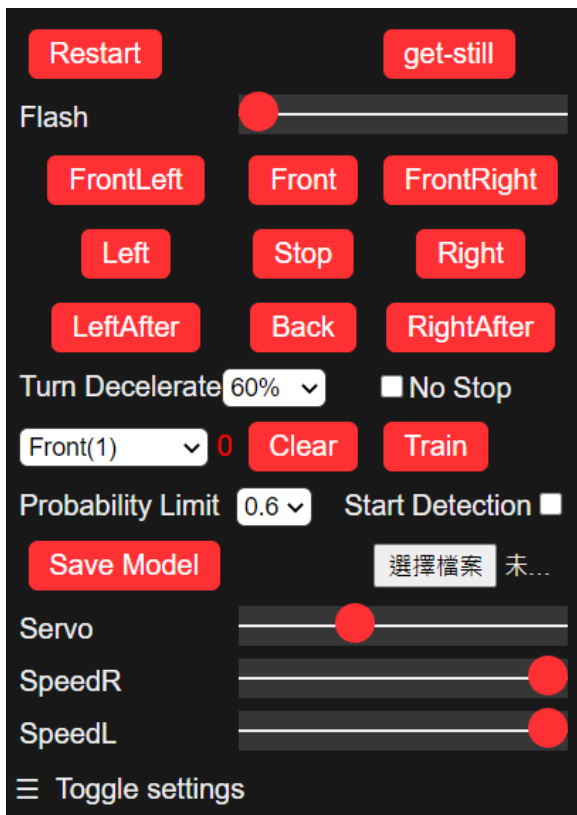
- 如果垂直 SERVO 在左側，上述第 1237、1239、1247、1249 行的加減符號需相反。
- 右圖為網頁執行畫面，可以直接控制登月小車前進的方向，『Turn Declerate 偏轉』是設定轉彎的幅度，選取『No Stop』是按下『Front 前進』就一直前進不會停止，直到按下下個按鈕動作才改變，若不選取，則按一下只會前進一下而已。
- 可以設定追蹤物件(Object)的種類，當追蹤到該物件，登月小車會自動轉彎前進，但不要移動速度太快，因為鏡頭辨識速度怕會跟不上。
- 視訊視窗會顯示 coco-ssd 所有內建的 80 種物件名稱，但是僅會根據選擇的物件 Object 動作。
- 下方 Servo 水平卷軸可以直接控制垂直方向的 SERVO，而 SpeedR 與 SpeedL 左、右輪速度預設值是 160(0~255 之間)，若網路速度較慢，會影響辨識物件的速度，所以車子前進速度就不可以太快。
- 當載入模型之後，若手機 4G 網路關閉，亦可正常繼續使用。
- 手動控制車子，同時可看到影片，適於主動觀察寵物或危險環境。





### 實作 17：ESP32-CAM\_CAR\_2pwm\_knn-classifier.ino 利用機器學習 KNN 分類控制車子

1. **Machine Learning** 機器學習的 kNN 分類演算法全名為「k Nearest Neighbor」，翻成中文意思就是「k 個最近的鄰居」。
2. 當網頁開啟時，底部顯示『**Please wait for loading model** 載入模型請稍後』消失後，才有視訊出現。
3. 與實作 16 一樣都可直接控制登月小車，實作 16 是利用追蹤物件來控制車子，而本實作 17 需先 **Train** 訓練幾個動作，未來便可辨識這些動作圖片來控制車子。
4. 選擇 1-9 各種方向，鏡頭盡量朝下拍到地面，當背景比較單純，辨識率就會提高率較高，然後按下 **Train** 訓練便會開始照相記錄，可多拍不同角度的內容，每按一次 **Train** 右邊數字會遞增，建議先訓練前進與停止兩種類別(class)，最後勾選『**Start Detection** 開始偵測』，便可開始使用不同手勢(剪刀、石頭、布)或自製圖片控制車子方向，若成功之後再多加幾個指令動作。
5. 訓練完畢之後，可按 **Save Model** 儲存模型，預設檔名為『model.json』，未來可以直接匯入就可開始辨識，不用每次使用前都要先訓練，若要清除訓練的紀錄則按 **Clear** 清除。
6. 最下方會顯示各個訓練模型的可能機率，取出最大值，若超過 **Probability Limit 0.6** 的臨界值，車子就會根據判斷結果動作。



註 1：所有訓練機率的總和必為 1，所以一定要將無照片或無手勢也當作 **STOP** 訓練。

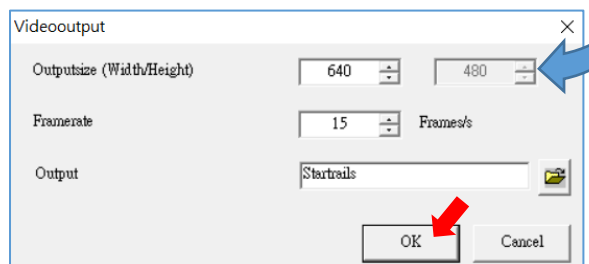
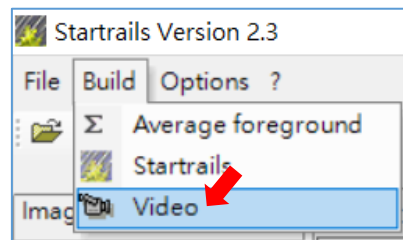
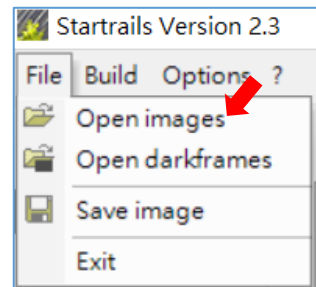
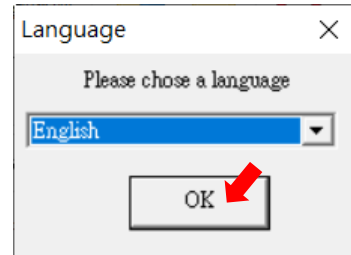
註 2：辨識背景越單純，辨識內容越大越佳。

## 實作 18：ESP32-CAM\_SD\_TimeLapse\_PAGE 縮時攝影

1. Time-lapse photography 縮時攝影，是一種將畫面拍攝頻率設定在遠低於一般觀看連續畫面所需頻率的攝影技術。一些常以縮時攝影拍攝的題材包括：雲與天空的變化、植物生長及花卉開花、水果腐敗、建築物的建造過程、城市中的人們、交通狀況等。以此技術拍攝變化緩慢、難以察覺的題材，會產生一種流暢的視覺感受。若用以拍攝變化快速的題材，則會呈現出猛烈變化的效果，若有興趣可參閱 [https://www.youtube.com/results?search\\_query=%E7%B8%AE%E6%99%82%E6%94%9D%E5%BD%B1](https://www.youtube.com/results?search_query=%E7%B8%AE%E6%99%82%E6%94%9D%E5%BD%B1)。




2. 執行時須連上網路並插入 SD，先設定拍攝畫質解析度 Resolution QVGA(320x240)、Interval(ms) 100 — 60000 間隔時間，再開啟 Start TimeLapse 便開始自動縮時攝影，並將照片存於 SD 卡內，SD 命名則是 1.jpg → 2.jpg.....，Reset Filename Start with 1 目的則是將檔案名稱又從 1.jpg 開始，檔案名稱會顯示在網頁最下方，每次拍照閃光燈都會亮一下。

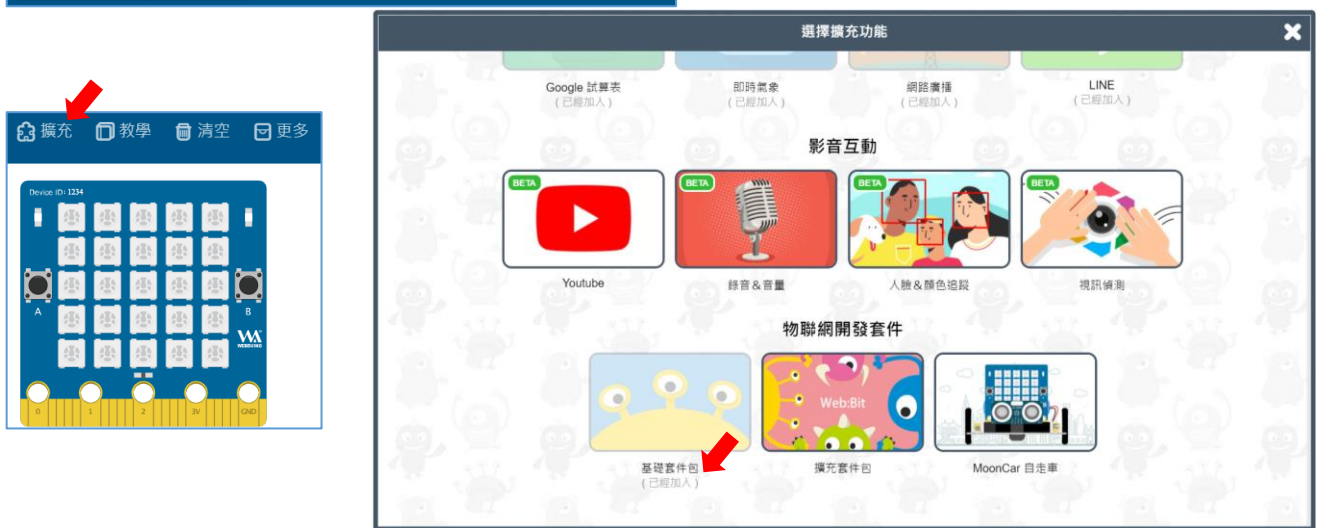


3. 縮時影片製作軟體建議使用免安裝『Startrails.exe』，可參閱 <https://www.kocpc.com.tw/archives/180783>，執行後先選擇 English 語言，但沒有中文，直接按 OK，利用『File 檔案 → Open images 開啟照片』匯入照片，利用『Build 建立 → Video 影片』，設定 Outputsize 輸出解析度大小、Framerate 是指每秒要播幾張照片(圖框)，Output 輸出路徑請指定，才知影片轉到哪裡，按下 OK 即可轉檔輸出，輸出格式為 AVI 未壓縮檔案，接著可將影片上傳到 YouTube 後，再使用內建的配樂功能。

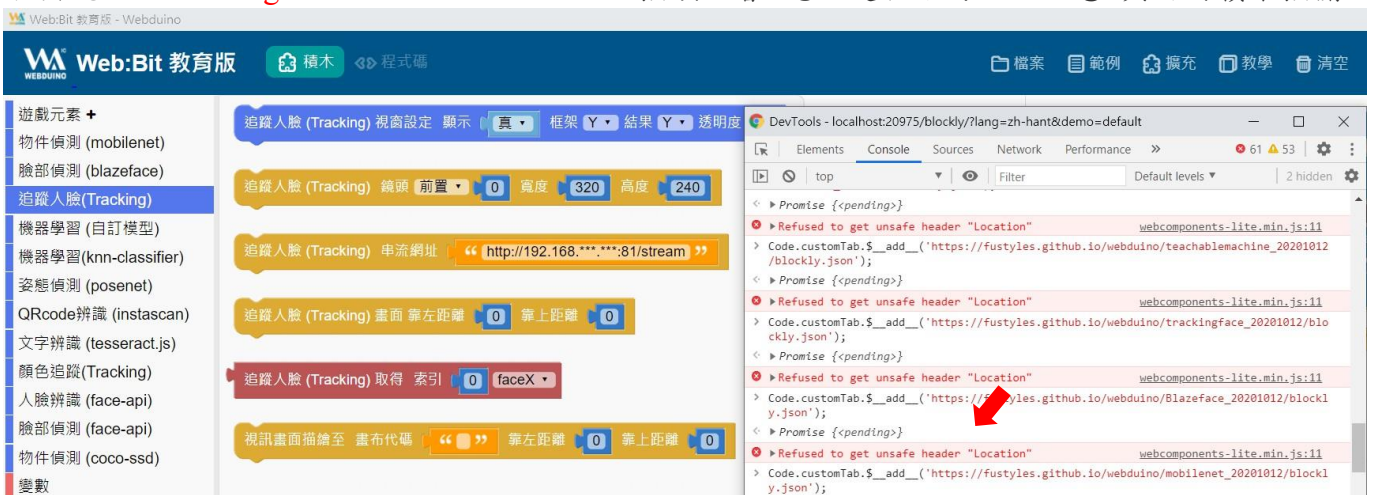


## 實作 19：結合 Web:Bit 應用

1. 利用 Web:bit 教育版軟體模擬器，先自訂影像辨識積木，可以練習影像辨識應用的邏輯思考，將積木程式轉換成 Javascript 語法置入 ESP32-CAM 使用或是直接搭配使用。
2. 軟體下載安裝 <https://ota.webduino.io/WebBitInstaller/WebBitSetup.exe>。
3. 安裝後先不要開啟軟體，需先外掛自訂積木，將資料夾 "實作 19\_Webbit 教育版\外掛自訂積木\basic-package"，將整個資料夾內容覆蓋到 "C:\Webduino\WebBit\package.nw\blockly\custom-blocks\basic-package"。
4. 開啟桌面 Webbit 軟體捷徑，按『系統 → 以瀏覽器開啟』，以利於自訂積木撰寫與除錯，且 AI 影像辨識積木才可正常執行，若按下右上方  可看到基礎套件包(已經加入)，表示上述步驟 3 已經完成。



5. 新增自訂積木方法：瀏覽器視窗空白處按右鍵 → 檢查 → Console 主控台，新增積木『Code.customTab.\$\_\_add\_\_('積木連結');』，移除積木『Code.customTab.\$\_\_remove\_\_('積木連結');』，若出現『Refused to get unsafe header "Location"』無須理會，應該是官方的錯誤訊息，與自訂積木無關。



請逐一複製貼上底下內容，以安裝或移除法蘭斯自訂積木，所有自訂積木清單請參閱 <https://github.com/fustyles/webduino/blob/master/CustomBlock.txt>

1. 遊戲元素(舊版需手動安裝)

```
Code.customTab.$__add__('https://fustyles.github.io/webduino/GameElements_20190131/blockly.json');
Code.customTab.$__remove__('https://fustyles.github.io/webduino/GameElements_20190131/blockly.json');
```



## 2. ESP32-CAM(舊版需手動安裝)

`Code.customTab.$__add__('https://fustyles.github.io/webduino/ESP32-CAM_20191201/blockly.json');`

`Code.customTab.$__remove__('https://fustyles.github.io/webduino/ESP32-CAM_20191201/blockly.json');`

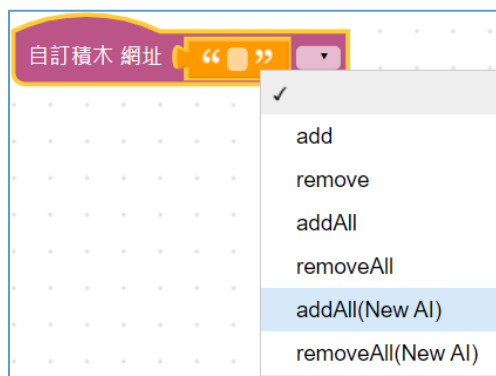
註：網站設定"不安全內容"須更改為"允許"，因為 ESP32 CAM 是走 http，會被阻擋才需要改安全性)，在 <https://webbit.webduino.io/blockly> 鎖頭圖示按右鍵→網站設定→不安全的內容改為允許



## 3. (新版) AI 積木(全)

`Code.customTab.$__add__('https://fustyles.github.io/webduino/EDU_addCustomBlocks/blockly.json');`

然後視窗左側會增加 自訂積木 →



機器學習 (自訂模型)  
物件偵測 (mobilenet)  
物件偵測 (coco-ssd)  
機器學習(knn-classifier)  
臉部偵測 (blazeface)  
姿態偵測 (posenet)  
人臉辨識 (face-api)  
顏色追蹤(Tracking)  
臉部偵測 (face-api)  
追蹤人臉(Tracking)  
文字辨識 (tesseract.js)  
QRcode辨識 (instascan)



註：ESP32 CAM 需先燒錄 `ESP32-CAM_MyBlockly_JSON.ino`，然後利用上述串流位置，就可以將 ESP2 CAM 與 Web:bit 結合，擴展更多邏輯思考應用的功能。

6. 利用『檔案 → 開啟 → 智慧開關 MQTT 控制.json → 執行』，同時開啟法蘭斯 MQTT 網頁 [https://fustyles.github.io/webduino/webduino\\_mqtt.html](https://fustyles.github.io/webduino/webduino_mqtt.html)，其中 Topic 主題名稱可自訂，但程式碼須和網頁內容的 topic 主題一致，下達指令後需按『send 送出』，執行結果如下：



註：網路廣播就是指 MQTT，是 Message Queueing Telemetry Transport 訊息佇列遙測傳輸的意思，許多智慧家電就是用 MQTT 訂閱同一個 Topic，便可以經由 MQTT Broker 發佈或接收訊息發佈，MQTT 不需要開發板實際連線，使用官方主機 1 對多控制超簡單，細節可參考網路文章 <https://swf.com.tw/?p=1002>。

