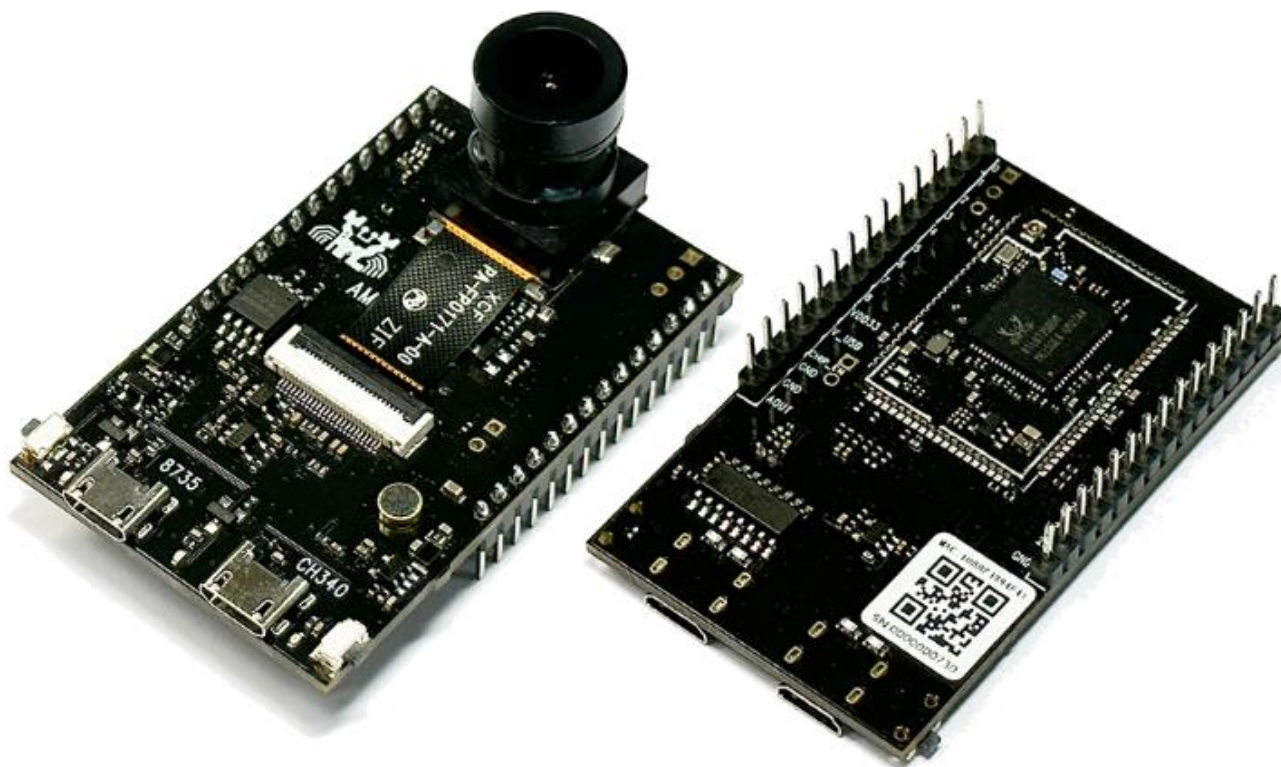


SMART AI CAM模組應用

講師: 楊頌煜 (Malo)

AMB82-mini

- 國產IC
- RTL8735B晶片

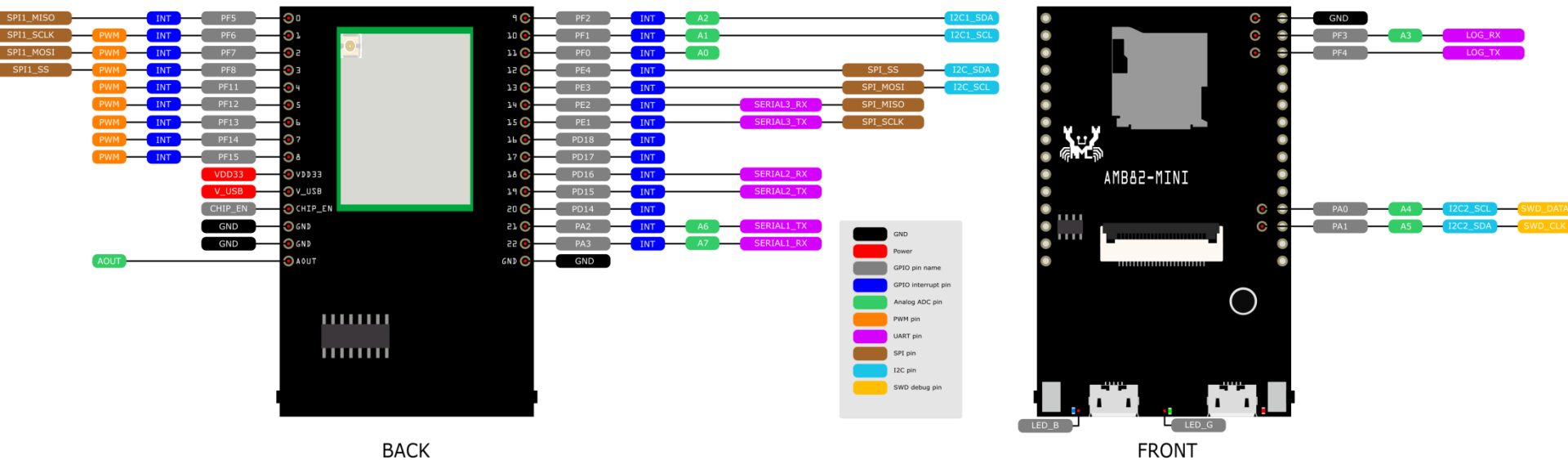


AMB82-mini

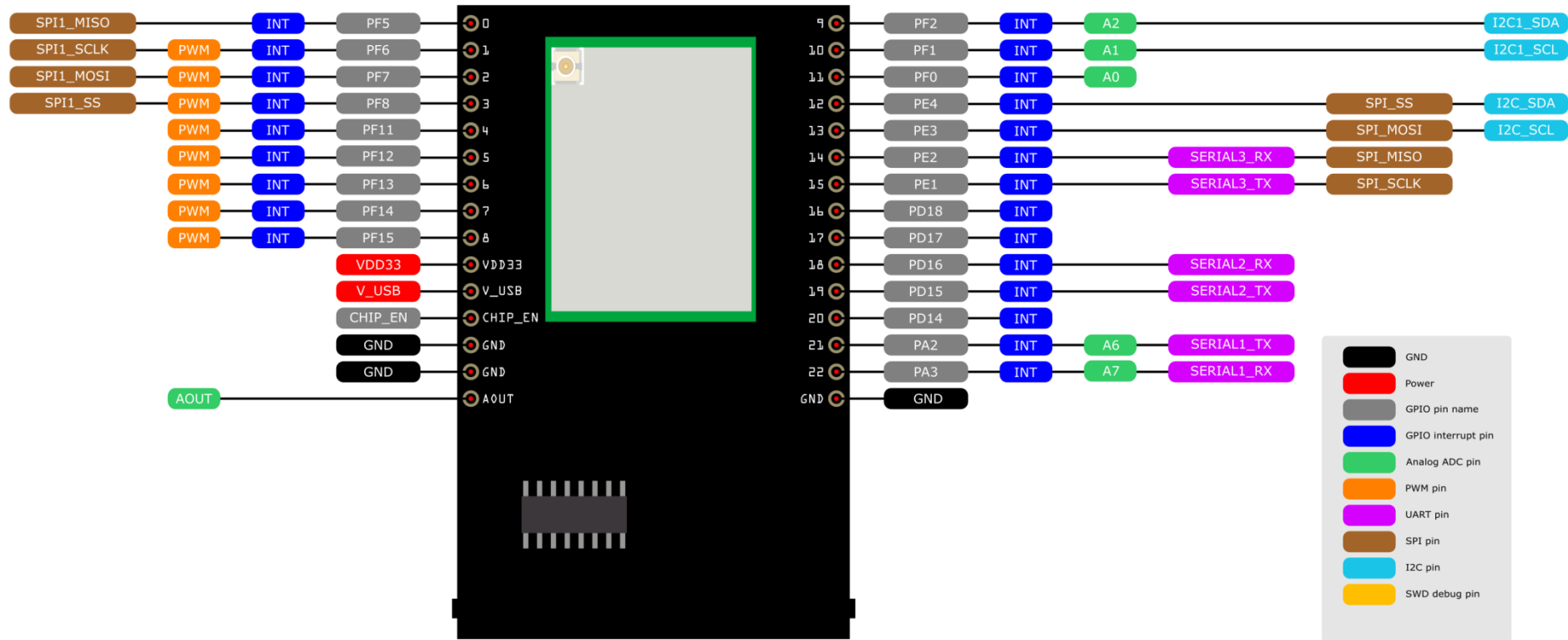
- 國產IC
- RTL8735B晶片
- 內建豐富介面: WiFi、BLE、GPIO INT、I2C、UART、SPI、PWM、ADC
- 使用最廣泛的開發環境之一 Arduino
- Ameba Arduino [介紹](#)。
- SDK及範例 [連結](#)。



AMB82-mini



AMB82-mini



BACK

AMB82-mini



編號	名稱	說明
1	8735 USB	連接至 RTL8735 原生 USB 介面。
2	CH340 USB	連接至 CH340，上傳韌體燒錄使用。
3	Upload Button	按住不放重置，會進入到燒錄模式。
4	Reset Button	系統重置按鈕。
5	LED 1 (Blue)	連接到 pin 23(PF9)，也是範例程式中的 LED_BUILTIN。
6	LED 2 (Green)	連接到 pin 24(PE6)。
7	Power LED (Red)	電源指示燈，開發板通電時點亮。



AMB82-mini

- 市場上少數具備AI算力IoT模組
- 市場上少數結合Camera的 IoT模組
- 國產網通晶片，導入產品沒有疑慮
- 支持Arduino原生開發環境
- 透過雲端載入不同AI 模型
- AI功能無限擴充，第一階段釋出AI功能為物件辨識、人臉辨識
- 聲音種類辨識 -> 進行中

腳位對應表

	GPIO pin	GPIO INT	ADC	PWM	UART	SPI	I2C	SWD	LED
0	PF5	✓				SPI1_MISO			
1	PF6	✓		✓		SPI1_SCLK			
2	PF7	✓		✓		SPI1_MOSI			
3	PF8	✓		✓		SPI1_SS			
4	PF11	✓		✓					
5	PF12	✓		✓					
6	PF13	✓		✓					
7	PF14	✓		✓					
8	PF15	✓		✓					
9	PF2	✓	A2				I2C1_SDA		
10	PF1	✓	A1				I2C1_SCL		
11	PF0	✓	A0						
12	PE4	✓				SPI_SS	I2C_SDA		
13	PE3	✓				SPI_MOSI	I2C_SCL		
14	PE2	✓			SERIAL3_RX	SPI_MISO			

腳位對應表

	GPIO pin	GPIO INT	ADC	PWM	UART	SPI	I2C	SWD	LED
15	PE1	✓			SERIAL3_TX	SPI_SCLK			
16	PD18	✓							
17	PD17	✓							
18	PD16	✓			SERIAL2_RX				
19	PD15	✓			SERIAL2_TX				
20	PD14	✓							
21	PA2	✓	A6		SERIAL1_TX				
22	PA3	✓	A7		SERIAL1_RX				
23	PF9	✓		✓					LED_BUILTIN / LEB_B (blue)
24	PE6	✓							LED_G (green)
25	PF4	✓			LOG_TX				
26	PF3	✓	*A3		LOG_RX				
27	PA1	✓	A5				I2C2_SDA	SWD_CLK	
28	PA0	✓	A4				I2C2_SCL	SWD_DATA	
29	PF10	✓							

開發環境的選擇

開發環境選擇：



透過GCC building on Linux or windows environment。
提供Edge AI、AWS-IoT、Amazon Kinesis Video streams的應用範例。



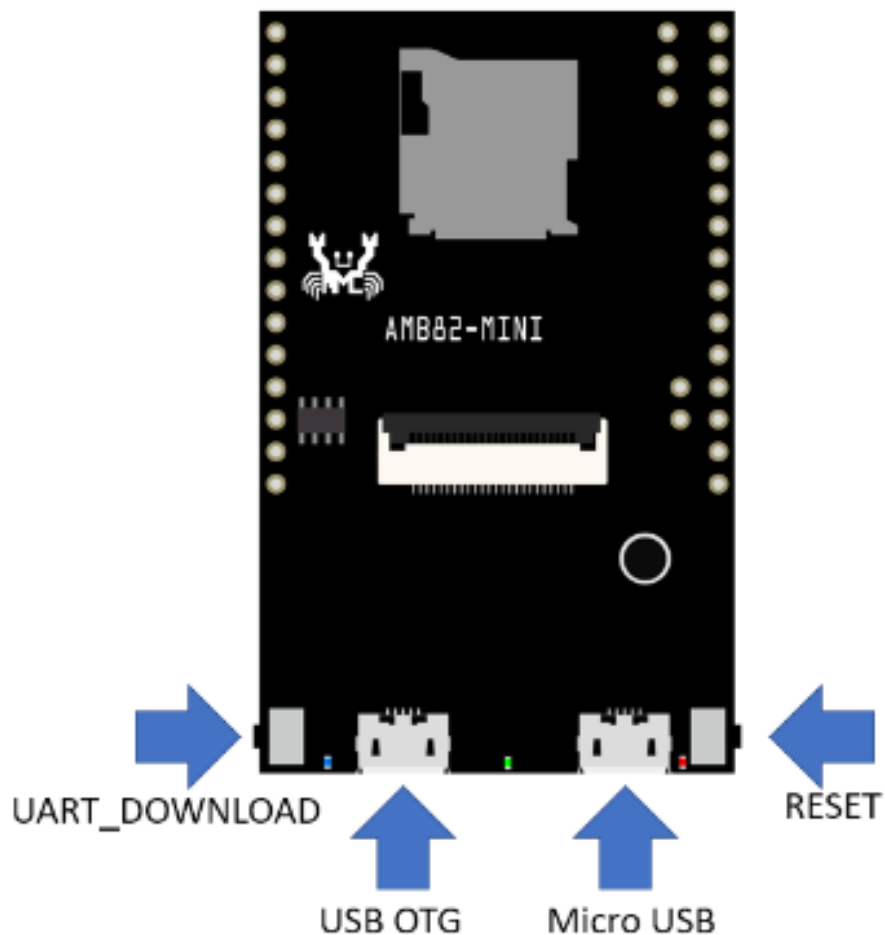
透過Arduino平台，提供 HUB 8735 各種功能範例，也可透過Github同步原廠的Release。



無須進行編譯。
透過外部Serial Port下AT Commands指令控制。

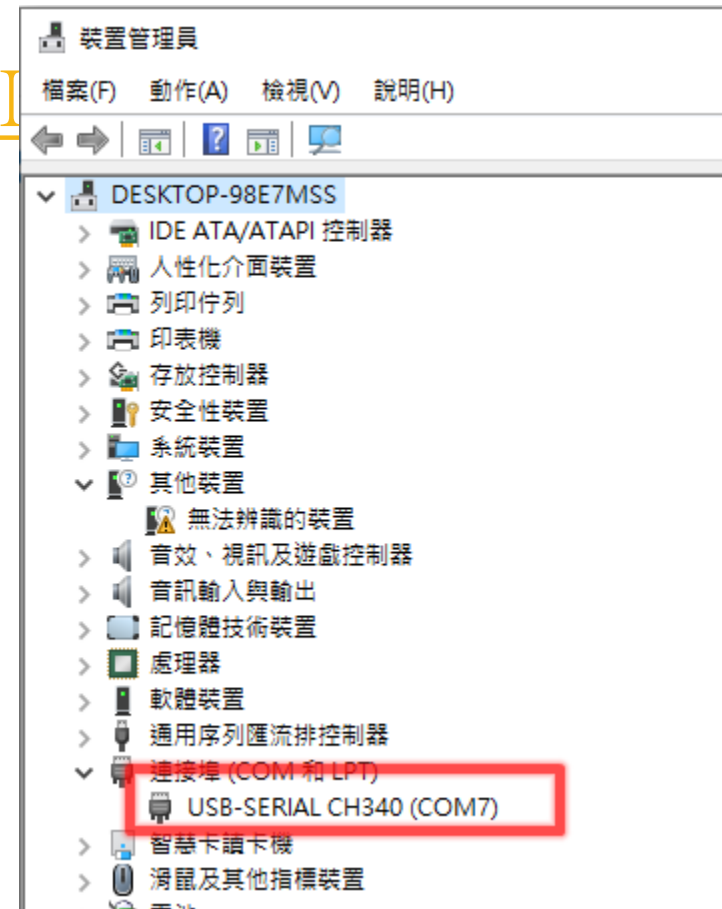
Arduino安裝說明

□ 燒錄介面說明




Arduino安裝說明

- USB驅動程式:
- http://www.wch-ic.com/downloads/CH341SER_ZI



Arduino安裝說明

- ❑ 盡量使用安裝版
- ❑ 使用1.8.19以上版本(講師使用1.8.19)

 [HARDWARE](#) [SOFTWARE](#) [CLOUD](#) [DOCUMENTATION](#) [COMMUNITY](#) [BLOG](#) [ABOUT](#)

Legacy IDE (1.8.X)



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 [Get](#) 

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

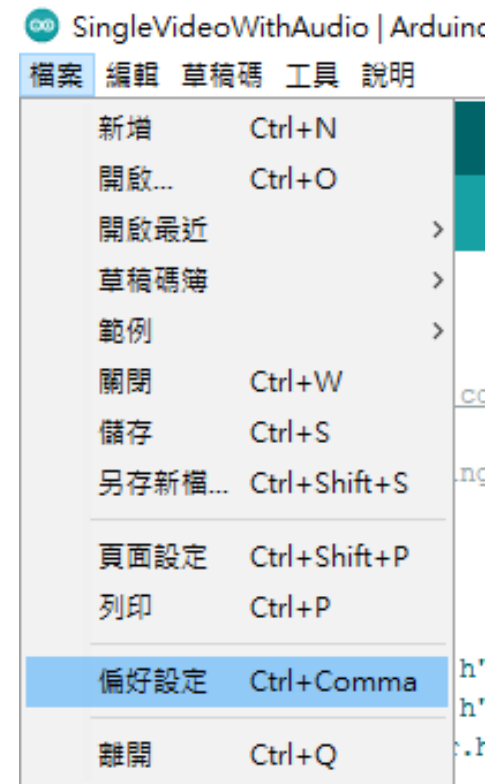
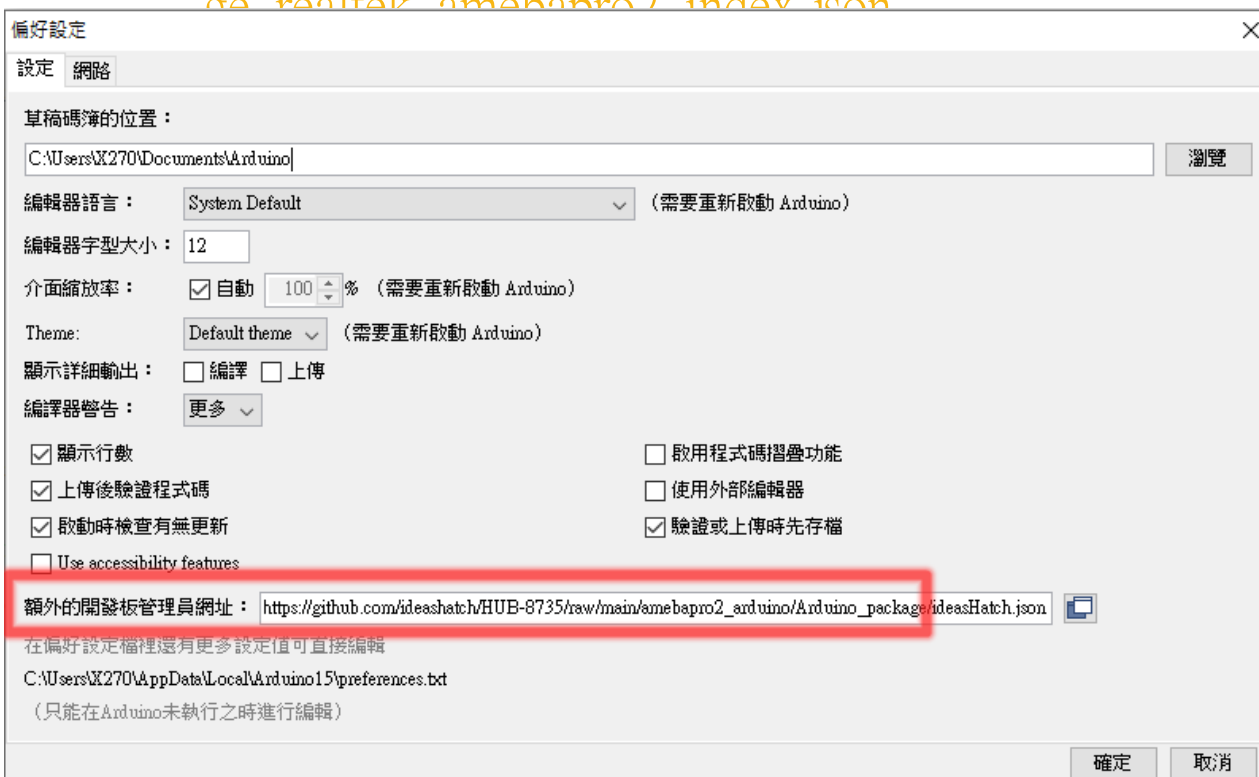
[Checksums \(sha512\)](#)

Arduino安裝說明

- 開啟Arduino IDE，打開「檔案」/「偏好設定」
(英文: “File” -> “Preferences”)

- 在「開發板管理員網址」填入網址：

https://github.com/ambiot/ambpro2_arduino/raw/main/Arduino_package/package_realtek_amebapro2_index.json



Arduino安裝說明

□ 選擇「工具」/「開發板管理員」

SingleVideoWithAudio | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and '開發板管理員...' (Board Manager...) is selected. The menu items are as follows:

- 自動格式化 (Ctrl+T)
- 封存草稿碼
- 修正編碼並重新載入
- 管理程式庫... (Ctrl+Shift+I)
- 序列埠監控視窗 (Ctrl+Shift+M)
- 序列繪圖家 (Ctrl+Shift+L)
- WiFi101 / Wi-Fi NINA Firmware Updater
- 開發板: "HUB-8735"**
- * Erase All Flash Memory (16MB): "Disable"
- * Auto Flash Mode: "Disable"
- * Standard Lib: "Disable"
- * Upload Speed: "921600"
- 序列埠
- 取得開發板資訊

The '開發板: "HUB-8735"' item is expanded, showing a list of boards:

- 開發板管理員...**
- AmebaPro2 ARM (32-bits) Boards - ideasHatch >
- Arduino AVR Boards >

The background code in the editor is:

```
1 /*  
2  
3 Example gu  
4 https://www.arduino.cc/en/tutorial/serial  
5  
6 For recom  
7 */  
8  
9 #include "V  
10 #include "S  
11 #include "V  
12 #include "Z  
13 #include "Z  
14 #include "P  
15
```

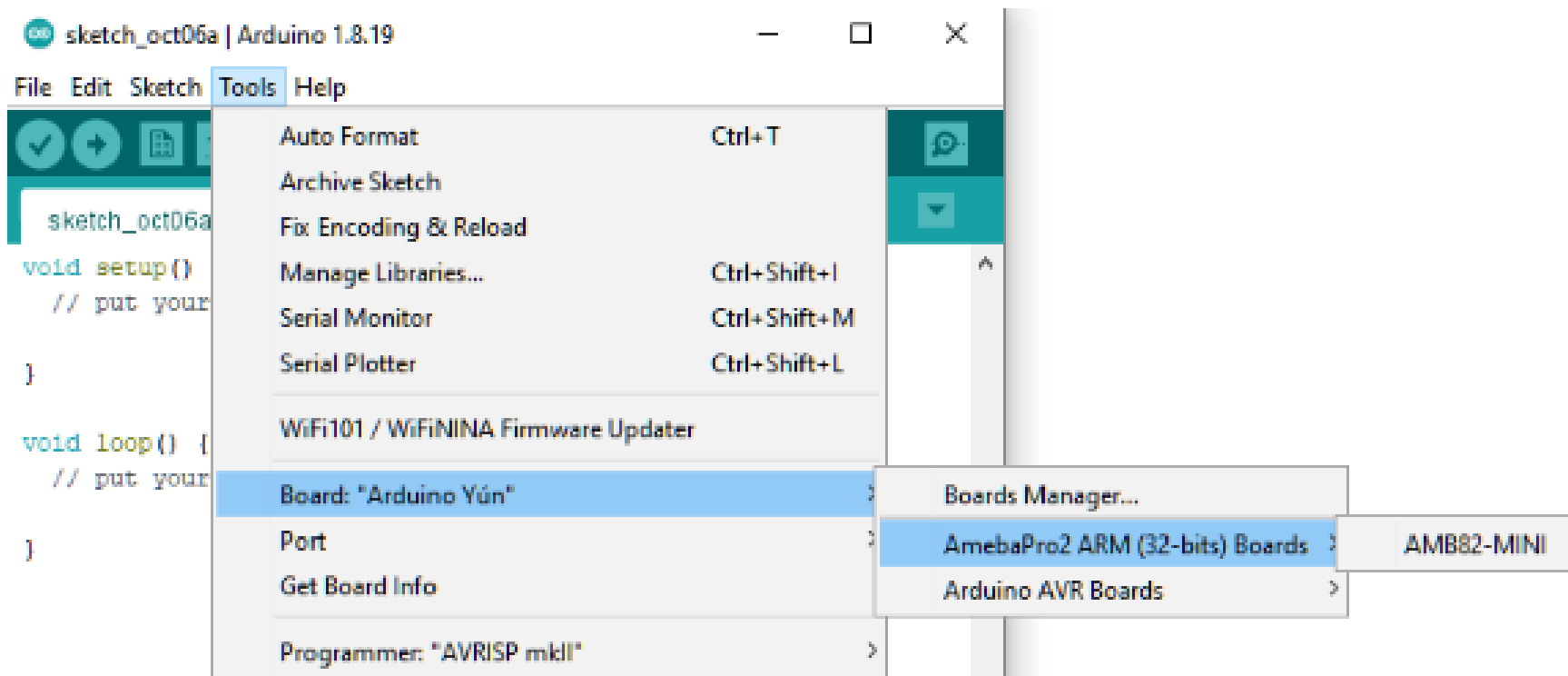
Arduino安裝說明

- 填入【AMB82 MINI】，如下圖畫面，選擇最新版本4.0.7



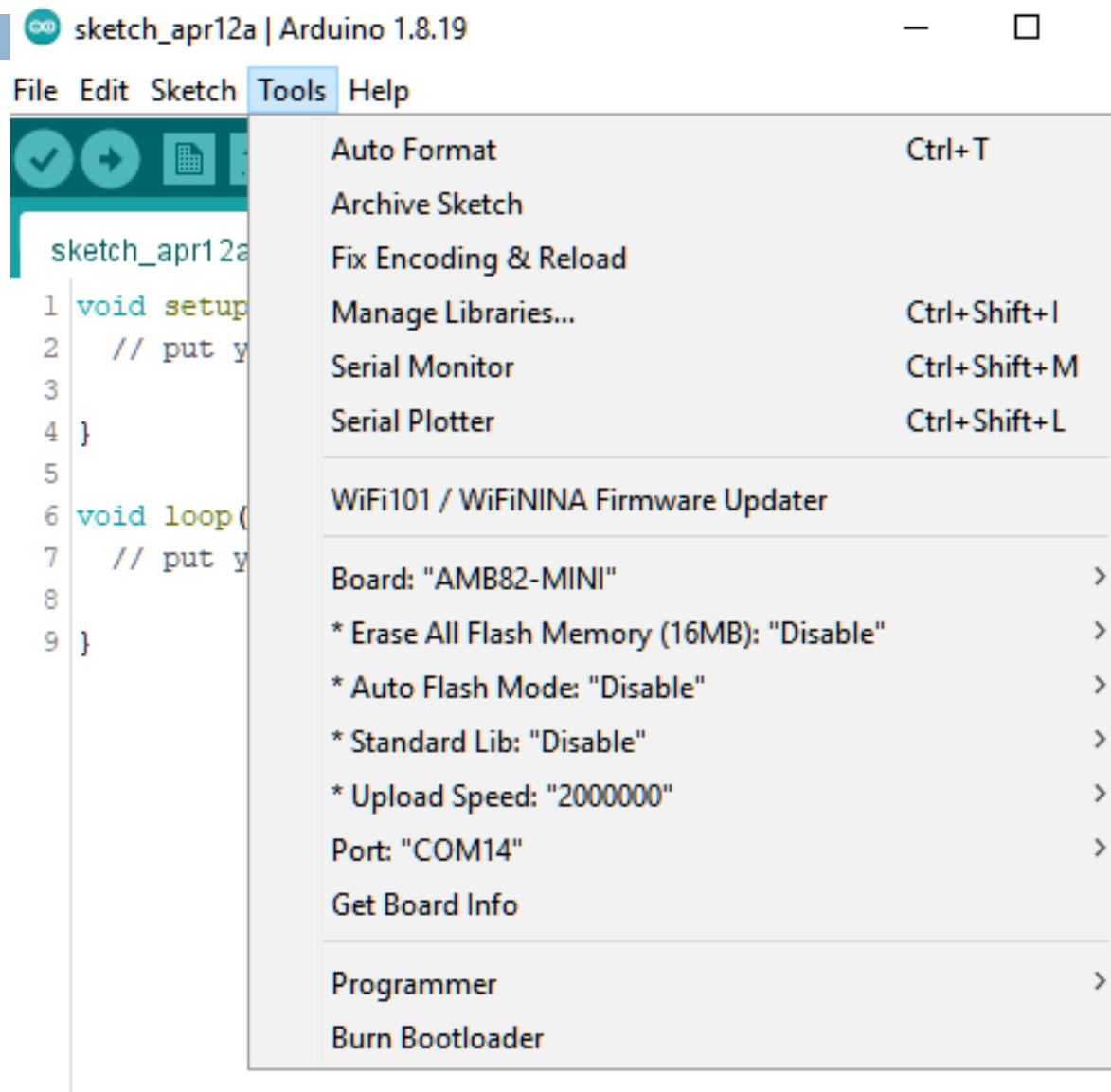
Arduino安裝說明

- 安裝好開發板套件後，再選擇 “Tools” -> “Board” -> “AmebaPro2 ARM (32-bits) Boards” -> “AMB82-MINI” 作為當前連接的開發板



Arduino安裝說明

- 設定參數
如右所示



Arduino安裝說

- 選擇後，等一些時間後，範例就會被引入
- 這時可以看到如右圖中，已經有不少可以直接套用的demo了

SingleVideoWithAudio | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

新增 Ctrl+N

開啟... Ctrl+O

開啟最近

草稿碼簿

範例

關閉 Ctrl+W

儲存 Ctrl+S

另存新檔... Ctrl+Shift+S

頁面設定 Ctrl+Shift+P

列印 Ctrl+P

偏好設定 Ctrl+Comma

離開 Ctrl+Q

Firmata

LiquidCrystal

SD

Stepper

Temboo

已廢棄

HUB-8735的範例

AmebaAnalog

AmebaBLE

AmebaFileSystem

AmebaGPIO

AmebaGTimer

AmebaHttp

AmebaMQTTClient

AmebaMultimedia

AmebaNN

AmebaNotify

AmebaPowerMode

AmebaRTC

AmebaSDDownload

AmebaSPI

AmebaWatchdog

AmebaWire

WiFi

Audio

CaptureJPEG

MotionDetection

RecordMP4

StreamRTSP

```
14 #include <Arduino.h>
15
16 #define CHANNEL 0
17
18 // Default preset config
19 // Channel 0 : 1920 x 1080
20 // Channel 1 : 1280 x 720
21 // Channel 2 : 1280 x 720
22
23 // Default audio preset
24 // 0 : 8kHz Mono Analog
25 // 1 : 16kHz Mono Analog
26 // 2 : 8kHz Mono Digital
27 // 3 : 16kHz Mono Digital
28
29 VideoSetting configV(
```

```
at java.awt.EventQueue.  
at java.awt.EventQueue.  
at java.awt.EventQueue.
```

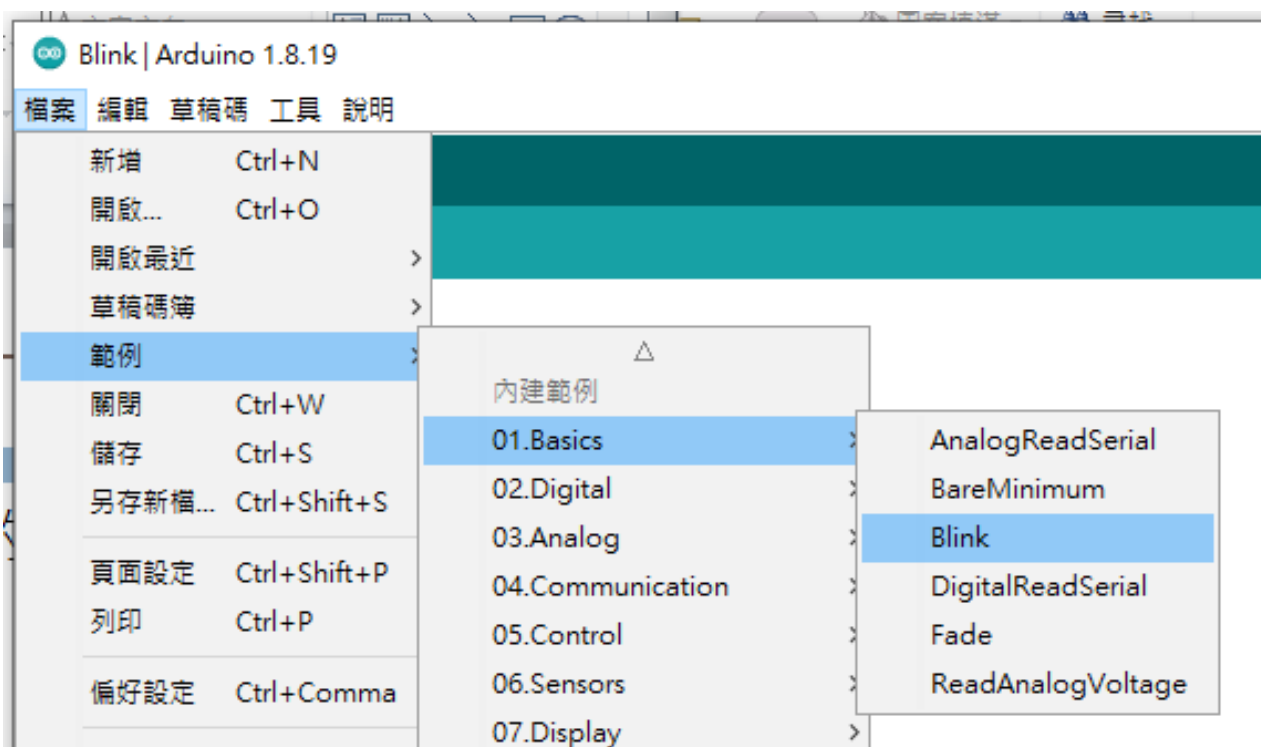
arduino-video-rtsp/
quality, please refer to

annel:

```
eventDispatchThread.java:  
eventDispatchThread.java:  
atchThread.java:82)
```

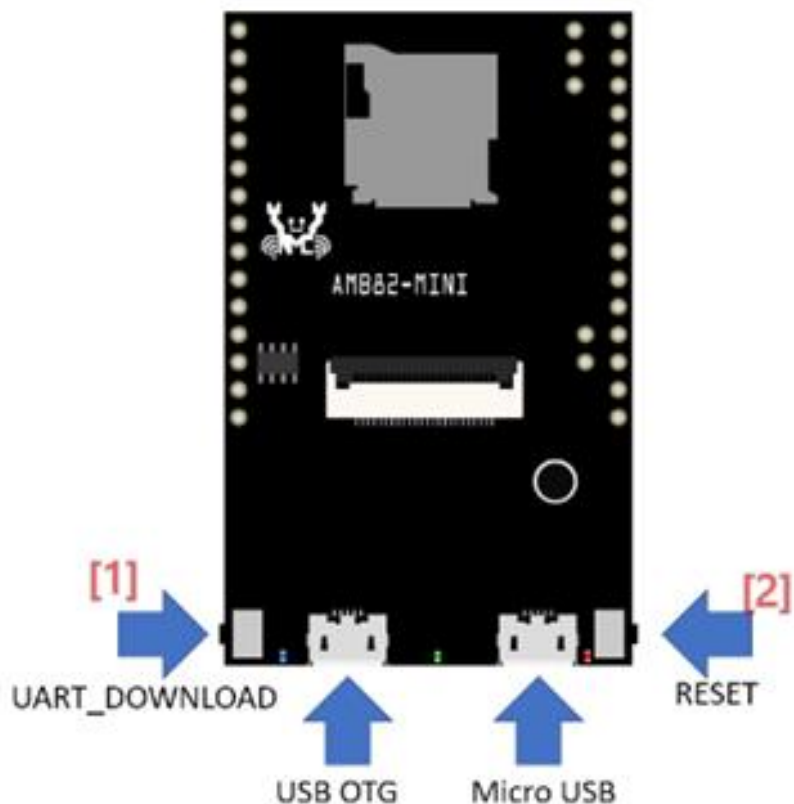
進行第一個程式燒錄: Blinky

- 點亮其上的GPIO所控制的Led燈



進行第一個程式燒錄: Blinky

- 先按住[1]，再按一下[2]放開，再放開[1]，即可進入燒錄模式。



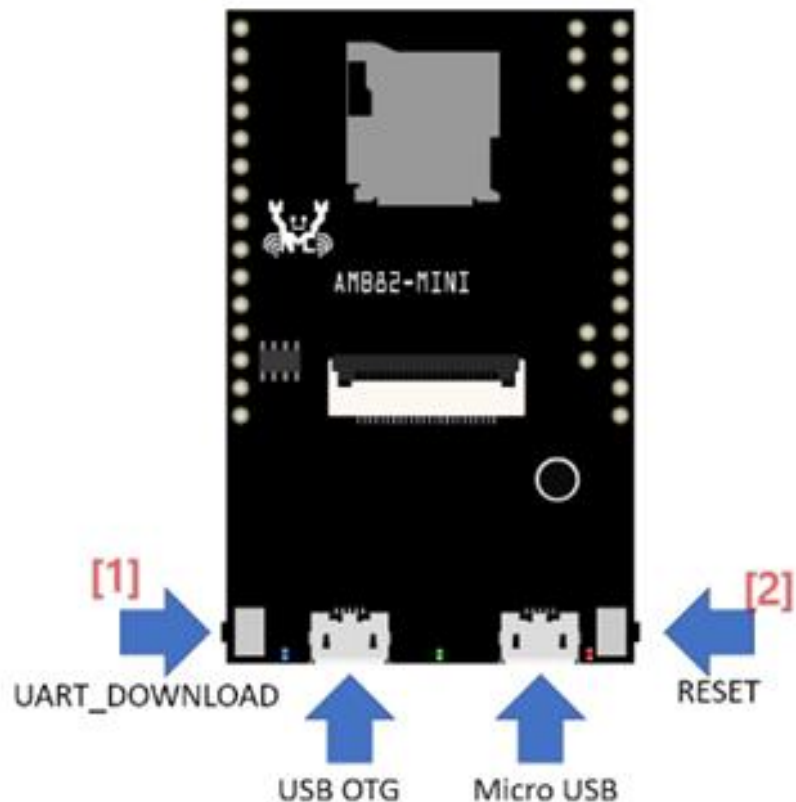
進行第一個程式燒錄: Blinky

- 進入燒錄模式後按[3]燒錄
- 完成可以看到[4]的訊息



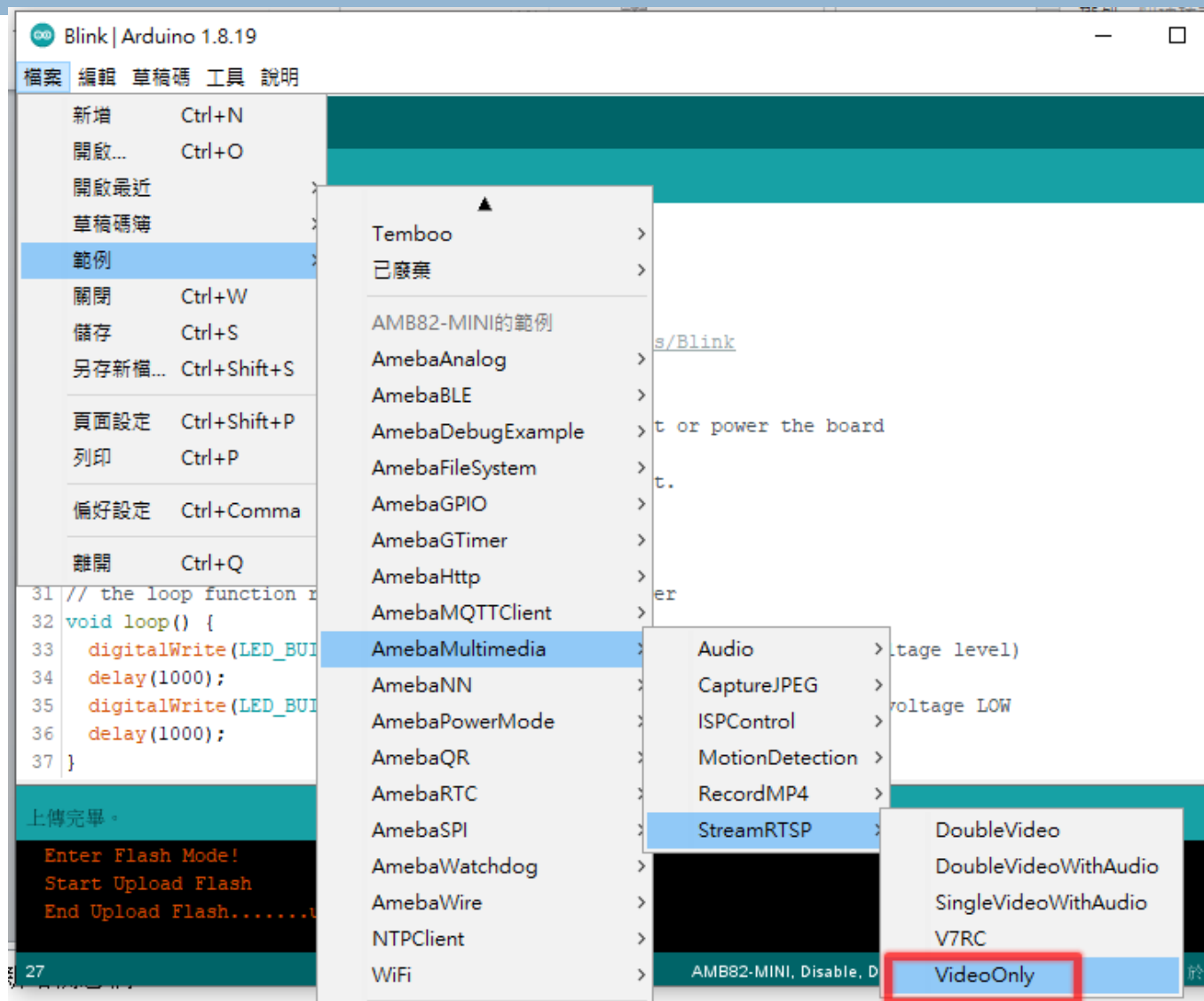
進行第一個程式燒錄: Blinky

- 再按一下[2]reset放開，即可執行程式
- 我們將會看到藍色Led燈閃爍



影像測試範例: VideoOnly

□ 選擇範例



測試範例: VideoOnly

- 修改ssid, pass
- 點「上傳」鈕



The screenshot shows the Arduino IDE interface. At the top, the project name is 'VideoOnly' and the board is 'Arduino 1.8.19'. Below the menu bar, there are icons for checking, uploading, and downloading. The 'Upload' icon (a right-pointing arrow) is highlighted with a red box and a red circle with the number '2'. The code editor shows the following code:

```
16 // Default preset configurations for each video channel.
17 // Channel 0 : 1920 x 1080 30FPS H264
18 // Channel 1 : 1280 x 720 30FPS H264
19 // Channel 2 : 1280 x 720 30FPS MJPEG
20
21 VideoSetting config(CHANNEL);
22 RTSP rtsp;
23 StreamIO videoStreamer(1, 1); // 1 Input Video -> 1 Output RTSP
24
25 char ssid[] = "Malo7a"; // your network SSID (name)
26 char pass[] = "12345678"; // your network password
27 int status = WL_IDLE_STATUS;
28
29 void setup() {
30     Serial.begin(115200);
31 }
```

The code lines 25, 26, and 27 are highlighted with a red box. Below the code editor, the status bar shows '上傳完畢。' (Upload complete). At the bottom, the command prompt shows the following output:

```
c:/users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameb
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限為 16777216 bytes。
Enter Flash Mode!
Start Upload Flash
Uploading.....upload success
```

測試範例: VideoOnly

- 等一段時間後，如果出現以下畫面，就代表燒錄成功了

上傳完畢。

```
c:/users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameba_pro2_toolchain/1.0.1-pl/bin/./lib/gcc  
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限為 16777216 bytes。
```

```
Enter Flash Mode!
```

```
Start Upload Flash
```

```
Uploading.....upload success
```

```
End Upload Flash
```

測試範例: VideoOnly

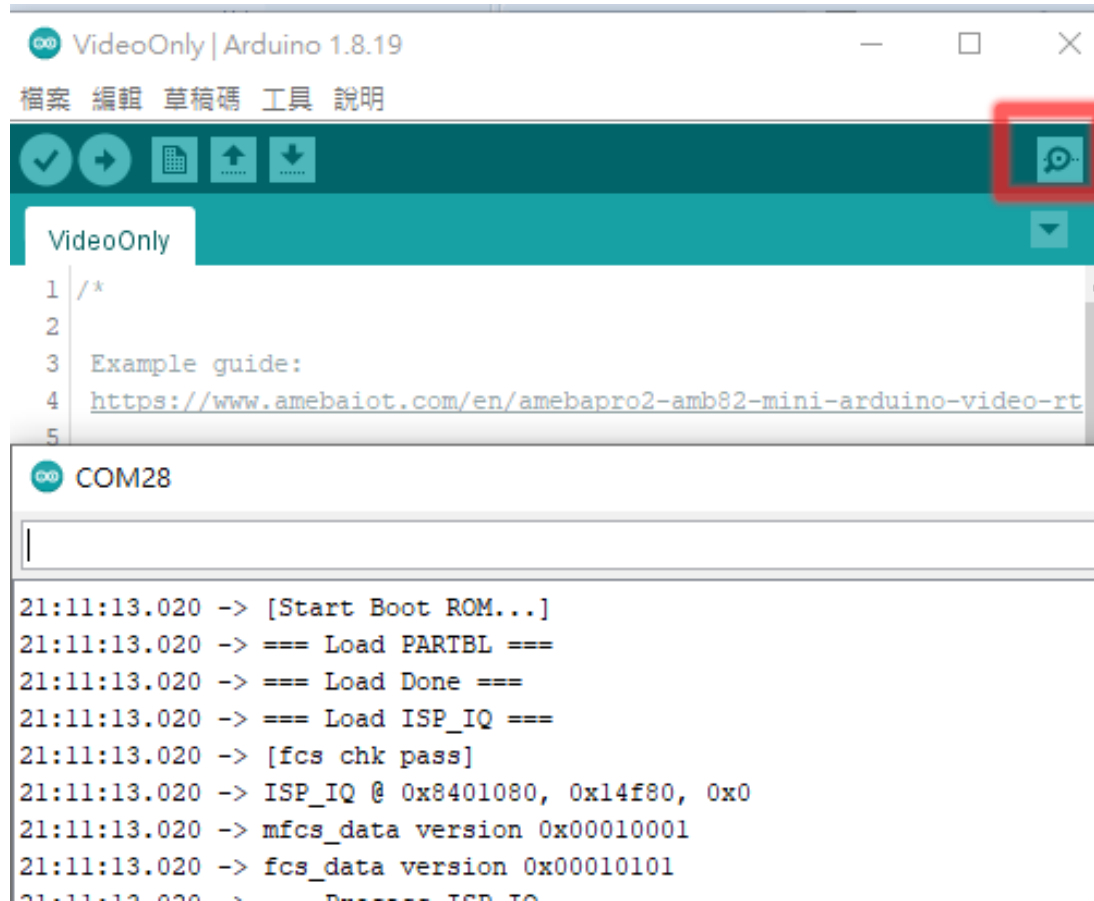
- 如果失敗。請確認一下把專案放在有中文名稱的資料夾，有可能編譯失敗，再請移到arduino專案資料夾，或是D槽底下

上傳完畢。

```
c:/users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameba_pro2_toolchain/1.0.1-pl/bin/./lib/gcc
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限為 16777216 bytes。
Enter Flash Mode!
Start Upload Flash
Uploading.....upload success
End Upload Flash
```

測試範例: VideoOnly

- 這時可以在點選如下，打開序列埠查看訊息
- 如果訊息已跑完，可以再按一下「reset」



測試範例: VideoOnly

- 在訊息中可以看到板子已經連上WiFi AP，它的IP是什麼

```
COM28

21:20:47.229 -> hal_voe_send2voe too long 102et ate cx0t0012 6 plqp 00 OpRan e I[25, 08000
21:20:47.229 -> B[25, 48] 2097152 bps pic 1 skip 0 hrd 0 cpbSize 1000000 bitrateWindow 3
21:20:47.275 -> [VOE]Set PreP: input 1920x1080 : offset 0x0 : format 1 : rotation 0cc 0 :
21:20:47.275 -> [VOE]vcenc_set_ref_pic_set() NULL
21:20:47.321 -> [VOE]etgain 14400 th 10000: day2night timer start
21:20:47.321 -> [VOE]etgain 1879445856 th 6658 :day2night timer stop
21:20:47.552 -> -----
21:20:48.248 -> - Summary of Streaming -
21:20:48.248 -> -----
21:20:48.248 -> Channel: 0
21:20:48.248 -> Encoder type: H264
21:20:48.248 -> Resolution: VIDEO_FHD
21:20:48.248 -> Video width: 1920
21:20:48.248 -> Video height: 1080
21:20:48.248 -> fps: 30
21:20:48.248 -> bps: 4194304
21:20:48.248 ->
21:20:48.248 -> - RTSP -
21:20:48.248 -> rtsp://192.168.1.144:554
21:20:48.248 ->
```

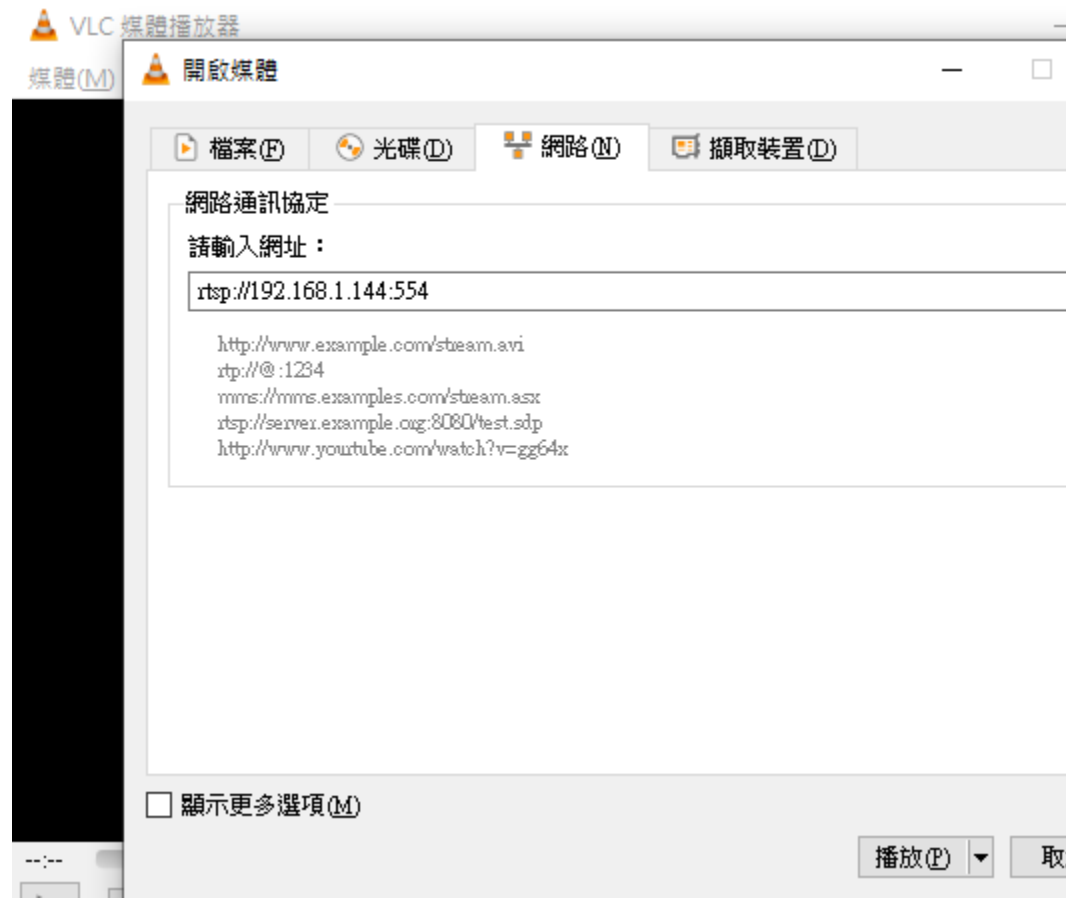
測試範例: VideoOnly

- 確認好ip後，我們可以使用VLC程式來看影像
- 在「開啟網路串流」中設定IP



測試範例: VideoOnly

- 輸入好連結後，點一下「播放」即可看到影像
- 例：rtsp://192.168.1.144:554



測試範例: VideoOnly

- 輸入好連結後，點一下「播放」即可看到影像
- 例

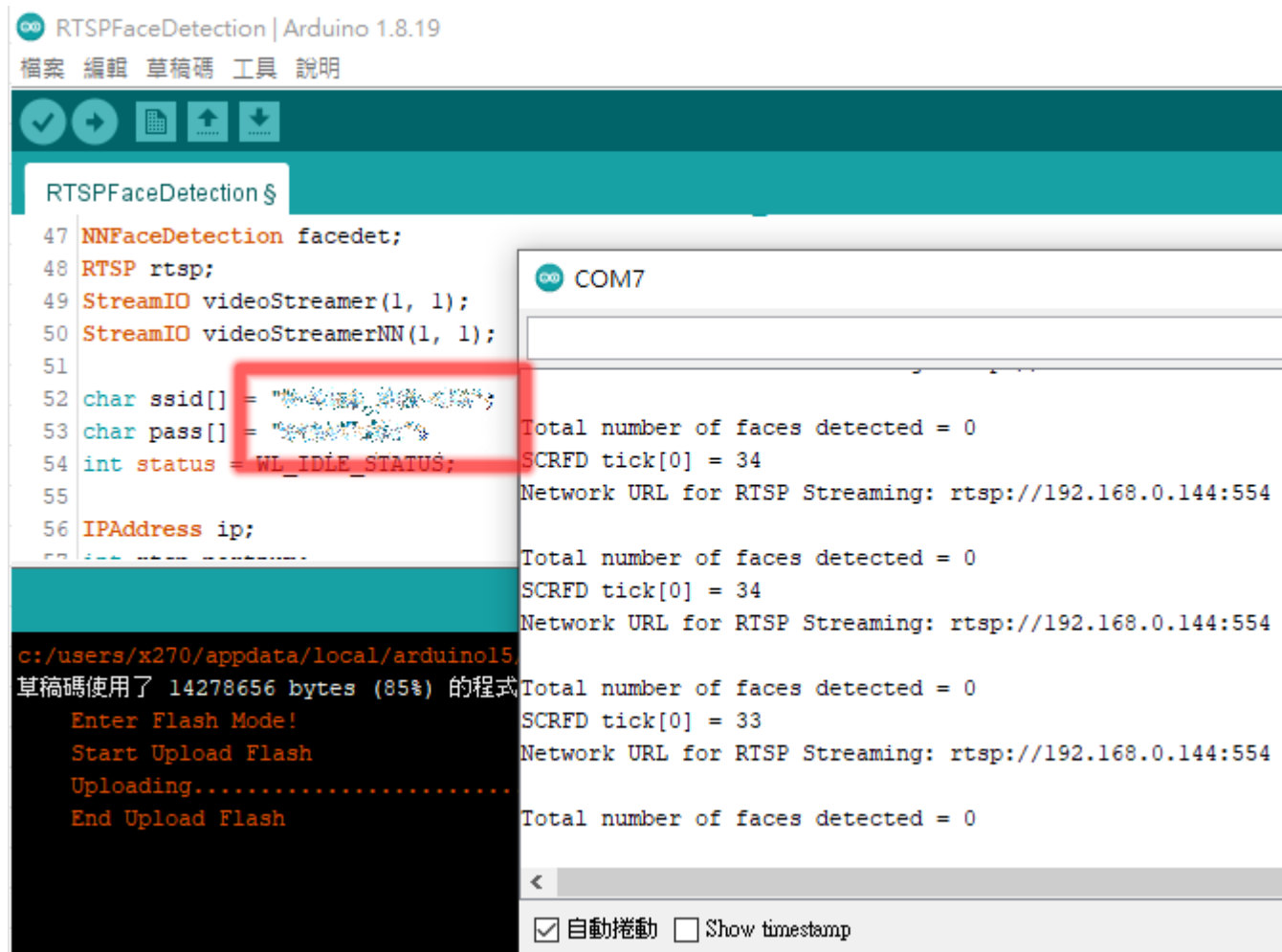


人臉偵測

RTSPFaceDetection

測試範例: RTSPFaceDetection

- 修改ssid, pass , 再燒錄



The screenshot displays the Arduino IDE interface for the 'RTSPFaceDetection' sketch. The code editor shows the following lines:

```
47 NNFaceDetection facedet;  
48 RTSP rtsp;  
49 StreamIO videoStreamer(1, 1);  
50 StreamIO videoStreamerNN(1, 1);  
51  
52 char ssid[] = "XXXXXXXXXX";  
53 char pass[] = "XXXXXXXXXX";  
54 int status = WL_IDLE_STATUS;  
55  
56 IPAddress ip;
```

The serial monitor, connected to COM7, shows the output of the program. It indicates that the sketch used 14278656 bytes (85%) of the flash memory and displays the following status messages:

```
Enter Flash Mode!  
Start Upload Flash  
Uploading.....  
End Upload Flash
```

The serial output also shows the results of the face detection process:

```
Total number of faces detected = 0  
SCRFD tick[0] = 34  
Network URL for RTSP Streaming: rtsp://192.168.0.144:554
```

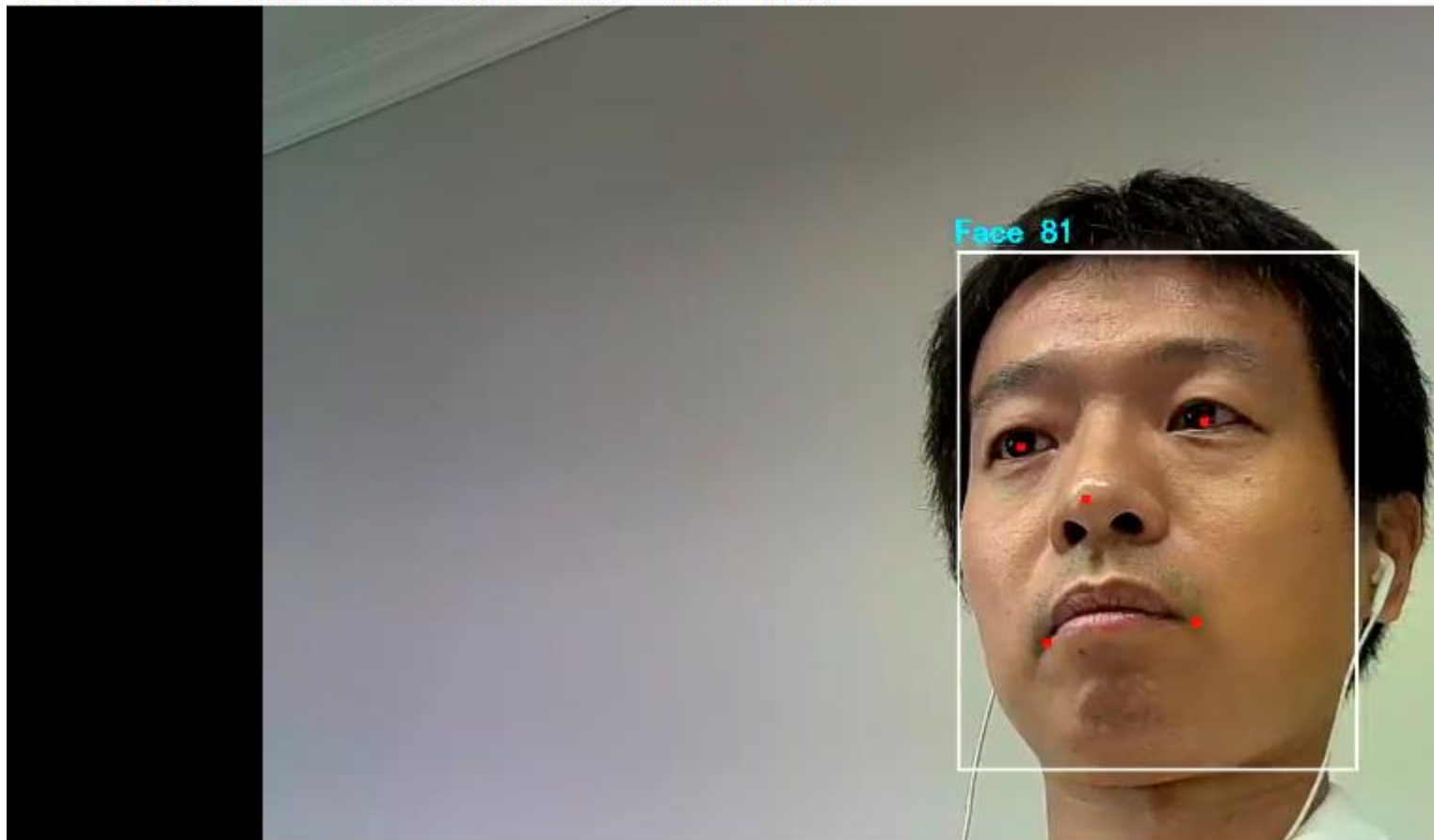
The bottom of the serial monitor shows a scroll bar and checkboxes for '自動捲動' (checked) and 'Show timestamp' (unchecked).

測試範例: RTSPFaceDetection

□ 使用VLC觀看(同demo1)

🔔 rtsp://192.168.0.144:554 - VLC 媒體播放器

媒體(M) 播放(P) 音訊(A) 視訊(V) 字幕(T) 工具(S) 檢視(V) 說明(H)



物件辨識

ObjectDetectionLoop

測試範例: ObjectDetectionLoop

- 先改ssid, pass
- 再燒錄
- 在uart可以看到IP

ObjectDetectionLoop | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

```
ObjectDetectionLoop $ ObjectClassList.h
49 StreamIO videoStreamer(1, 1);
50 StreamIO videoStreamerNN(1, 1);
51
52 char ssid[] = "meow" // your network SSID (name)
53 char pass[] = "12345678" // your network password
54 int status = WL_IDLE_STATUS;
55
56 IPAddress ip;
57 int rtsp_portnum;
58
59 void setup() {
60     Serial.begin(115200);
61
```

上傳完畢。

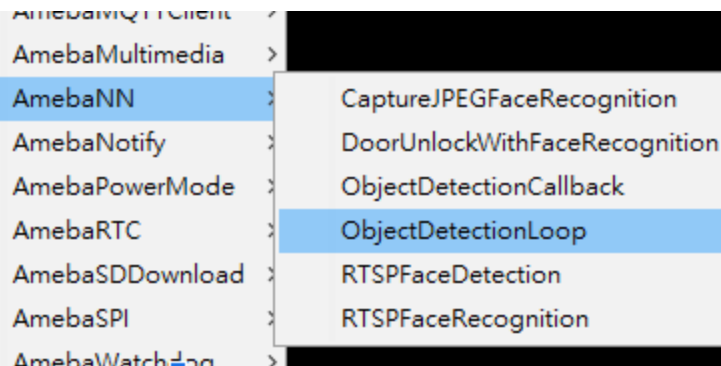
! /users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameba
草稿碼使用了 14278656 bytes (85%) 的程式儲存空間。上限為 16777216 bytes。

Enter Flash Mode!

Start Upload Flash

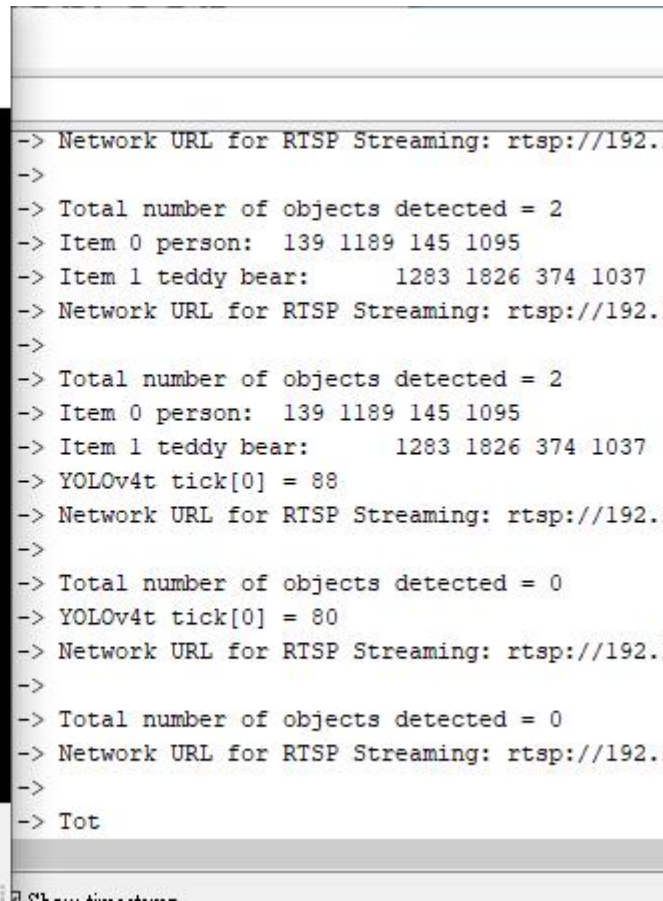
Uploading.....upload success

End Upload Flash



測試範例: ObjectDetectionLoop

- ❑ 訊息中會有目前辨識到的物件名稱
- ❑ ~~這個範例目前還很容易跑到掛掉，待原廠處理~~



測試範例: ObjectDetectionLoop

- 訊息中會有目前辨識到的物件名稱



控制Servo

PWM_ServoControl

測試範例: PWM_ServoControl

□ 燒錄過程同前

SingleVideoWithAudio | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

新增 Ctrl+N
開啟... Ctrl+O
開啟最近 >
草稿碼簿 >
範例 >
關閉 Ctrl+W
儲存 Ctrl+S
另存新檔... Ctrl+Shift+S
頁面設定 Ctrl+Shift+P
列印 Ctrl+P
偏好設定 Ctrl+Comma
離開 Ctrl+Q

09.USB >
10.StarterKit_BasicKit >
11.ArduinoISP >

任何板子皆可用的範例

Adafruit Circuit Playground >
Bridge >
Ethernet >
Firmata >
LiquidCrystal >
SD >
Stepper >
Temboo >
已廢棄 >

HUB-8735的範例

AmebaAnalog >
AmebaBLE >
AmebaFileSystem >

PWM_BuzzerPlayMelody

PWM_ServoControl

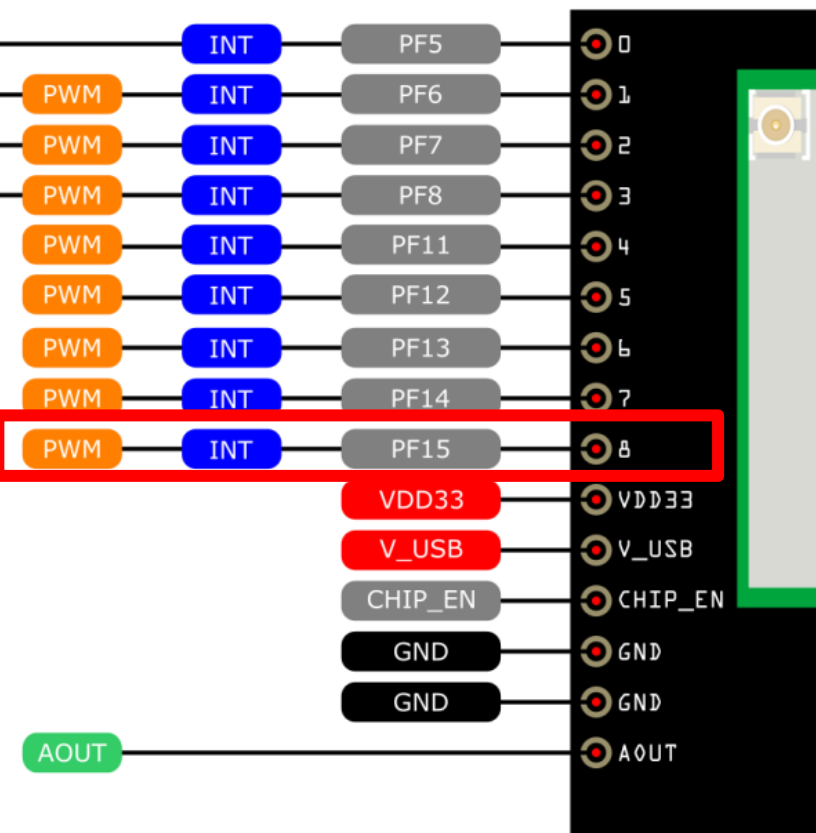
```
14 #include <RISCV.h>
15
16 #define CHANNEL 0
17
18 // Default preset con
19 // Channel 0 : 1920 x
20 // Channel 1 : 1280 x
21 // Channel 2 : 1280 x
22
23 // Default audio pres
24 // 0 : 8kHz Mono Ana
25 // 1 : 16kHz Mono Ana
```

arduino-video-rtsp/
quality, please refer to our

annel:

測試範例: PWM_ServoControl

- 使用pin8 (PF15) 具有PWM功能



PWM_ServoControl | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

PWM_ServoControl

```
13 https://www.amebaiot.com/en/amebapro2-amb82-mini-arduino-pwm-
14 */
15
16 #include <AmebaServo.h>
17
18 // create servo object to control a servo
19 // servo objects can be created correspond to PWM pins
20 AmebaServo myservo;
21
22 // variable to store the servo position
23 int pos = 0;
24
25 void setup() {
26     myservo.attach(8);
27 }
```

上傳完畢。

草稿碼使用了 4284416 bytes (25%) 的程式儲存空間。上限為 16777216 bytes。

Enter Flash Mode!

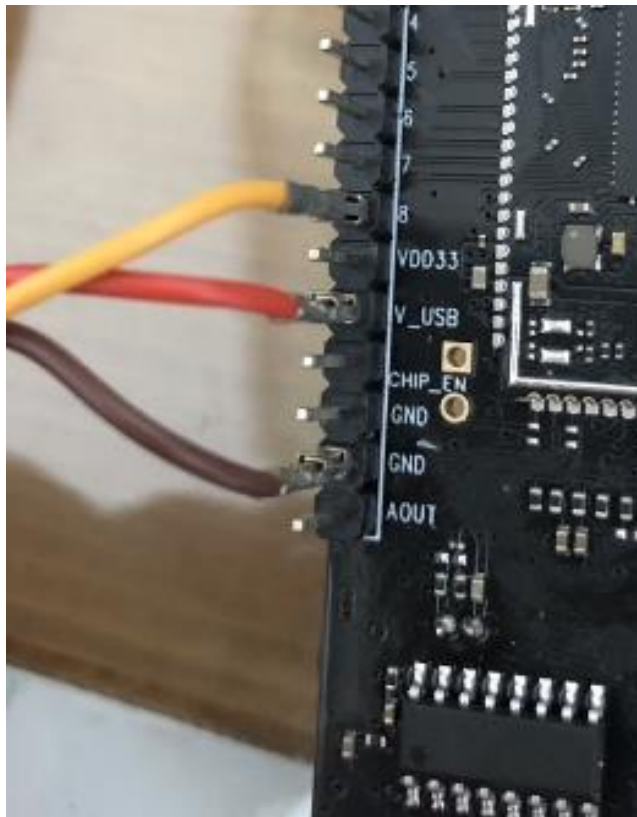
Start Upload Flash

Uploading.....upload success

End Upload Flash

測試範例: PWM_ServoControl

- 請把SG90的三個pin腳取出
- 分別接到5V, Gnd, pin8的腳位



測試範例: PWM_ServoControl

- 請把SG90的三個pin腳取出
- 分別接到5V, Gnd, pin8的腳位
- 燒錄程式後，會由0度轉到180度，再轉回0度
- <https://www.amebaiot.com/zh/amebapro2-amb82-mini-arduino-pwm-servo/>
- Ameba的Servo API仿照Arduino的API, 但是Arduino已經有原生的Servo API, 所以header檔我們改成 “AmebaServo.h”, Class的名稱也改成 AmebaServo

測試範例: PWM_ServoControl

- 我們可以進一步把程式簡單改為如右
- (燒錄前先把servo取下，比較不會因用電問題失敗)

servo_sg90 | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明



servo_sg90

```
24
25 void setup() {
26     myservo.attach(8); //pin8=F7, pin7=F8
27 }
28
29 void loop() {
30     myservo.write(0);
31     delay(2000);
32
33     myservo.write(180);
34     delay(2000);
35
36     myservo.write(90);
37     delay(5000);
38 }
```

上傳完畢。

草稿碼使用了 4284416 bytes (25%) 的程式儲存空間。上

Enter Flash Mode!

Start Upload Flash

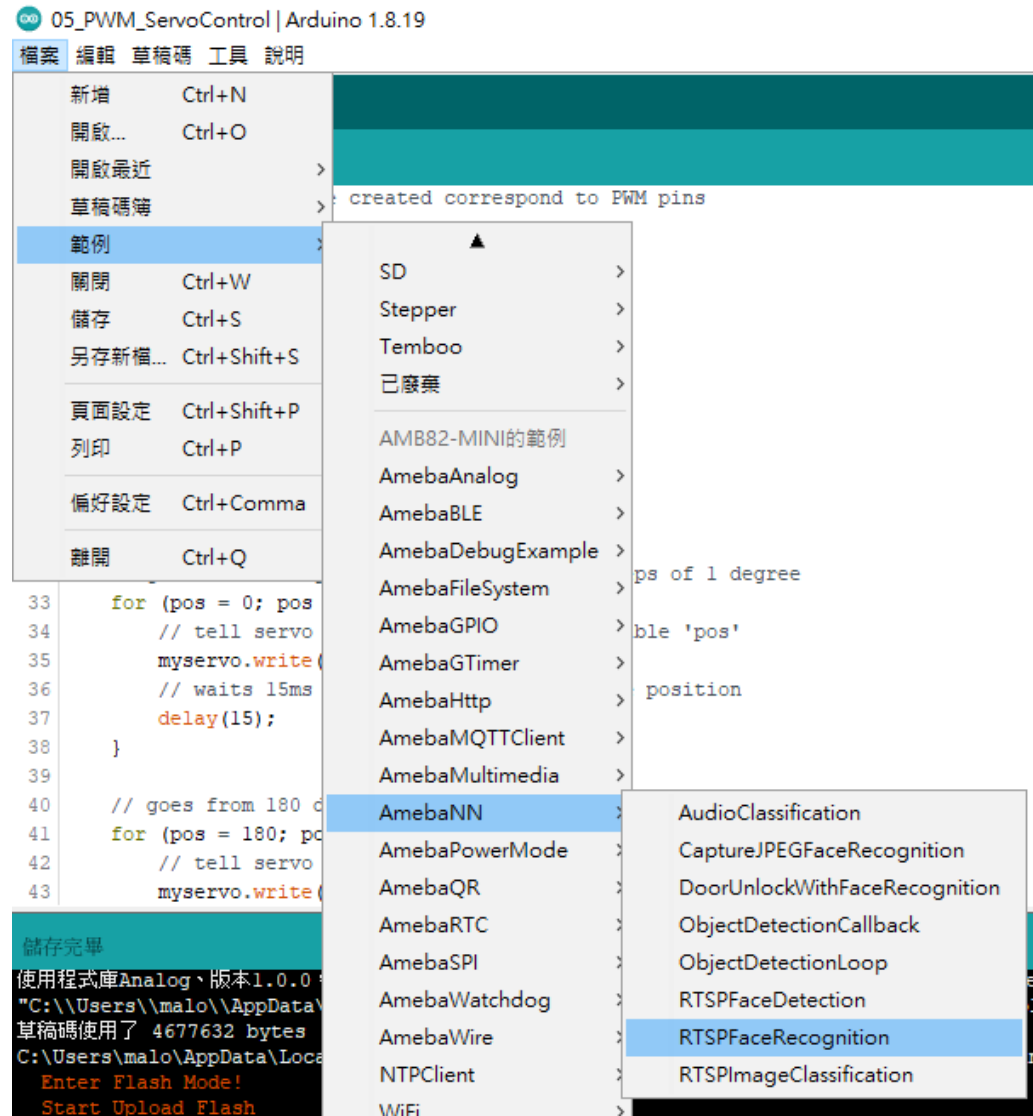
Uploading.....upload success

End Upload Flash

人臉辨識

測試範例: RTSPFaceRecognition

□ 開啟範例程式



測試範例: RTSPFaceRecognition

- 修改ssid, pass

RTSPFaceRecognition | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明



RTSPFaceRecognition \$

```
55 VideoSetting configNN(NNWIDTH, NNHEIGHT, 10, VIDEO_RGB, 0);
56 NNFaceDetectionRecognition facerecog;
57 RTSP rtsp;
58 StreamIO videoStreamer(1, 1);
59 StreamIO videoStreamerFDFR(1, 1);
60 StreamIO videoStreamerRGBFD(1, 1);
61
62 char ssid[] = "RTSPFaceRecognition"; // your network SSID (name)
63 char pass[] = "1234567890"; // your network password
64 int status = WL_IDLE_STATUS;
65
66 IPAddress ip;
```

上傳完畢。

c:/users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameba_

草稿碼使用了 14278656 bytes (85%) 的程式儲存空間。上限為 16777216 bytes。

Enter Flash Mode!

Start Upload Flash

Uploading.....upload success

End Upload Flash

測試範例: RTSPFaceRecognition

□ 一開始辨識為unknown

COM28

```
22:47:35.366 -> SCRFD tick[0] = 36
22:47:35.411 -> center 217,180 rotate -7.305760
22:47:35.411 -> MBFACENET tick[0] = 23
22:47:35.459 -> -----> no registered face data
22:47:35.459 -> Network URL for RTSP Streaming: rtsp://192.168.1.144:554
22:47:35.459 ->
22:47:35.459 -> Total number of faces detected = 1
22:47:35.459 -> Face 0 name unknown: 623 969 337 826
22:47:35.459 -> SCRFD tick[0] = 36
22:47:35.504 -> center 218,180 rotate -7.305760
22:47:35.504 -> MBFACENET tick[0] = 23
22:47:35.551 -> -----> no registered face data
22:47:35.551 -> Network URL for RTSP Streaming: rtsp://192.168.1.144:554
22:47:35.551 ->
22:47:35.551 -> Total number of faces detected = 1
22:47:35.551 -> Face 0 name unknown: 626 969 337 826
22:47:35.597 -> SCRFD tick[0] = 35
22:47:35.597 -> center 217,180 rotate -7.305760
22:47:35.643 -> MBFACENET tick[0] = 24
22:47:35.643 -> >>>>>>>> MBFACENET FPS = 9.93, 288 28996
```



測試範例: RTSPFaceRecognition

- 下指令「REG=Malo」時，只能有目標在影像中
- 之後看到Malo就會標記出來

The image displays a terminal window on the left and a video player on the right, illustrating the RTSPFaceRecognition process.

Terminal Window (COM28):

- The command `REG=Malo` is entered and highlighted with a red box.
- Subsequent log entries show face detection results for 'Malo'.
- The first detection at 22:49:02.154 shows coordinates: 249 693 155 739.
- The second detection at 22:49:02.292 shows coordinates: 256 693 155 735.
- The third detection at 22:49:02.386 shows coordinates: 256 693 151 739.
- The final detection at 22:49:02.433 shows coordinates: 116,140 rotate -12.804266.

Video Player:

- The video shows a man in a blue shirt.
- A green bounding box labeled 'Face: Malo' is drawn around the man's face, indicating successful recognition.
- The video player interface includes controls for media (M), play (P), audio (A), video (V), subtitles (D), and tools (I).

人臉辨識門禁系統

DoorUnlockWithFaceRecognition

測試範例:

DoorUnlockWithFaceRecognition

DoorUnlockWithFaceRecognition | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

新增 Ctrl+N

開啟... Ctrl+O

開啟最近

草稿碼簿

範例

關閉 Ctrl+W

儲存 Ctrl+S

另存新檔... Ctrl+Shift+S

頁面設定 Ctrl+Shift+P

列印 Ctrl+P

偏好設定 Ctrl+Comma

離開 Ctrl+Q

```
14 #include "AmebaGPIO.h"
15 #include "AmebaFatFS.h"
16
17 #define CHANNELVID 0
18 #define CHANNELJPEG 1
19 #define CHANNELNN 3
20
21 // Customised resolution
22 #define NNWIDTH 576
23 #define NNHEIGHT 320
24
25 // OSD layers
26 #define RECTTEXTLAYER
```

Firmata

LiquidCrystal

SD

Stepper

Temboo

已廢棄

HUB-8735的範例

AmebaAnalog

AmebaBLE

AmebaFileSystem

AmebaGPIO

AmebaGTimer

AmebaHttp

AmebaMQTTClient

AmebaMultimedia

AmebaNN

AmebaNotify

AmebaPowerMode

Ensure that there is only one
Stop trying to register a face
Forget all previously register
Save registered faces to flash
Load registered faces from fla

meaning
napshots
NN only available on channel 3

CaptureJPEGFaceRecognition

DoorUnlockWithFaceRecognition

ObjectDetectionCallback

測試範例:

DoorUnlockWithFaceRecognition

- ❑ 修改ssid, pass
- ❑ 確認接到pin8
- ❑ 完整版加入以下

```
29 // Pin Definition
30 #define RED_LED 2 //p2(A1)
31 #define GREEN_LED 3 //p3(A0)
32 #define BUTTON_PIN 5 //p5(A2)
33 #define SERVO_PIN 8 //F7
34
```

DoorUnlockWithFaceRecognition | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明



DoorUnlockWithFaceRecognition \$

```
31 #define GREEN_LED 4
32 #define BUTTON_PIN 5
33 #define SERVO_PIN 8
34
35 VideoSetting configVID(VIDEO_FHD, 30, VIDEO_H264, 0);
36 VideoSetting configJPEG(VIDEO_FHD, CAM_FPS, VIDEO_JPEG, 1);
37 VideoSetting configNN(NNWIDTH, NNHEIGHT, 10, VIDEO_RGB, 0);
38 NNFaceDetectionRecognition facerecog;
39 RTSP rtsp;
40 StreamIO videoStreamer(1, 1);
41 StreamIO videoStreamerFDFR(1, 1);
42 StreamIO videoStreamerRGBFD(1, 1);
43 AmebaServo myservo;
44
45 char ssid[] = "your-network-ssid" // your network SSID (name)
46 char pass[] = "your-network-pass" // your network password
47 int status = WL_IDLE_STATUS;
48
```

上傳完畢。

c:/users/x270/appdata/local/arduino15/packages/ideashatch/tools/ameba-
草稿碼使用了 14278656 bytes (85%) 的程式儲存空間。上限為 16777216 bytes。

Enter Flash Mode!

Start Upload Flash

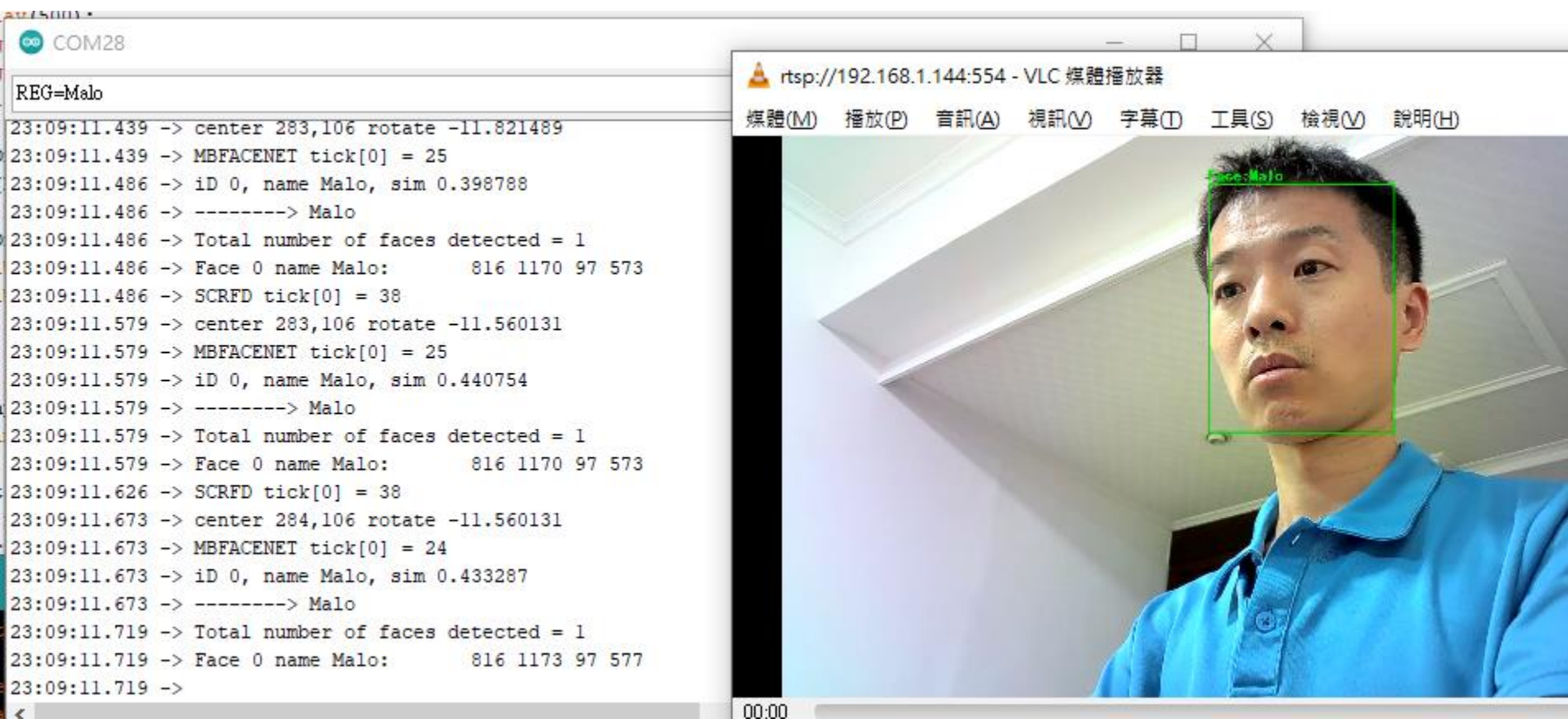
Uploading.....upload success

End Upload Flash

測試範例:

DoorUnlockWithFaceRecognition

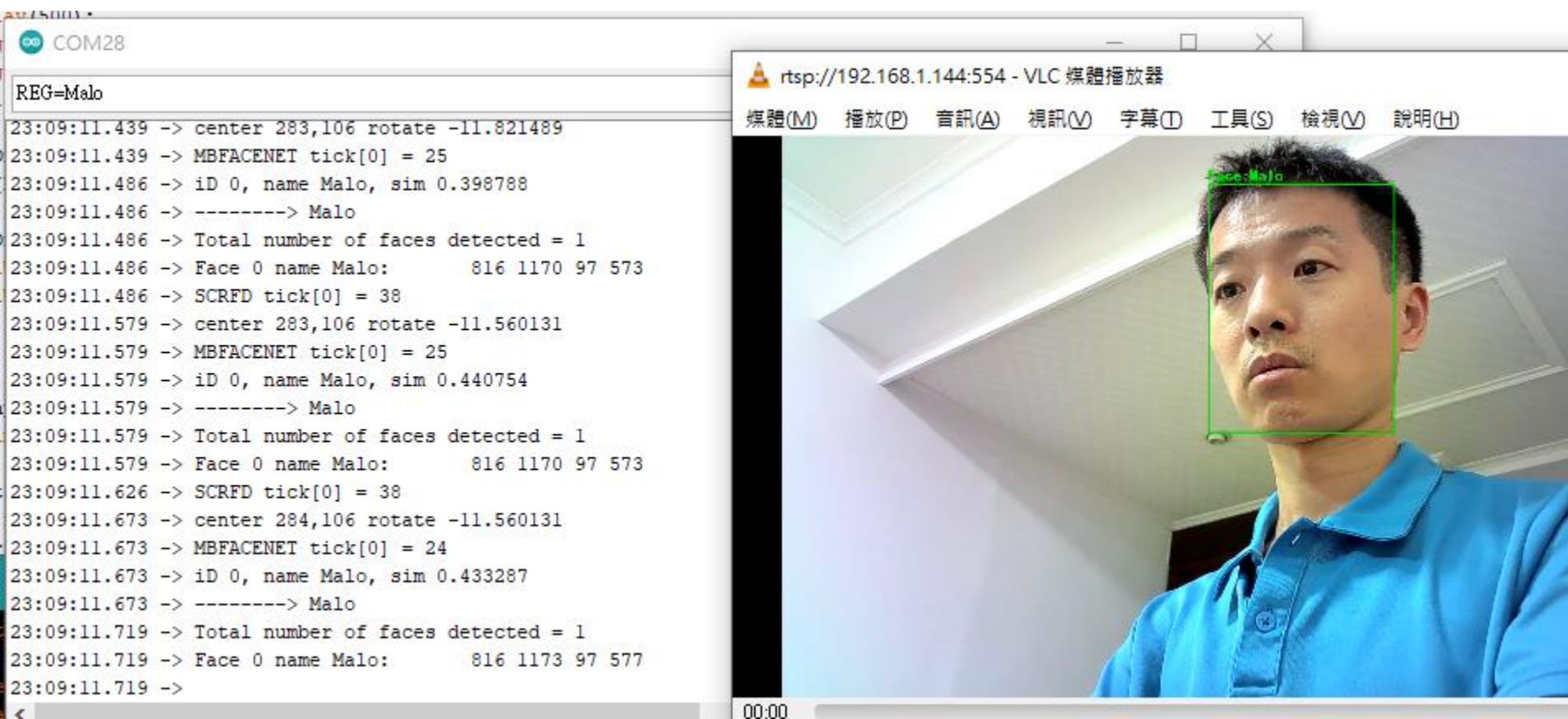
- 同臉部辨識一樣使用REG去註冊一張臉



測試範例:

DoorUnlockWithFaceRecognition

- 同臉部辨識一樣使用REG去註冊一張臉



測試範例: RTSPFaceDetection_Track

- 人臉追蹤雲台應用
- 人臉離開影像正中央時，攝影機會自動跟隨轉動