

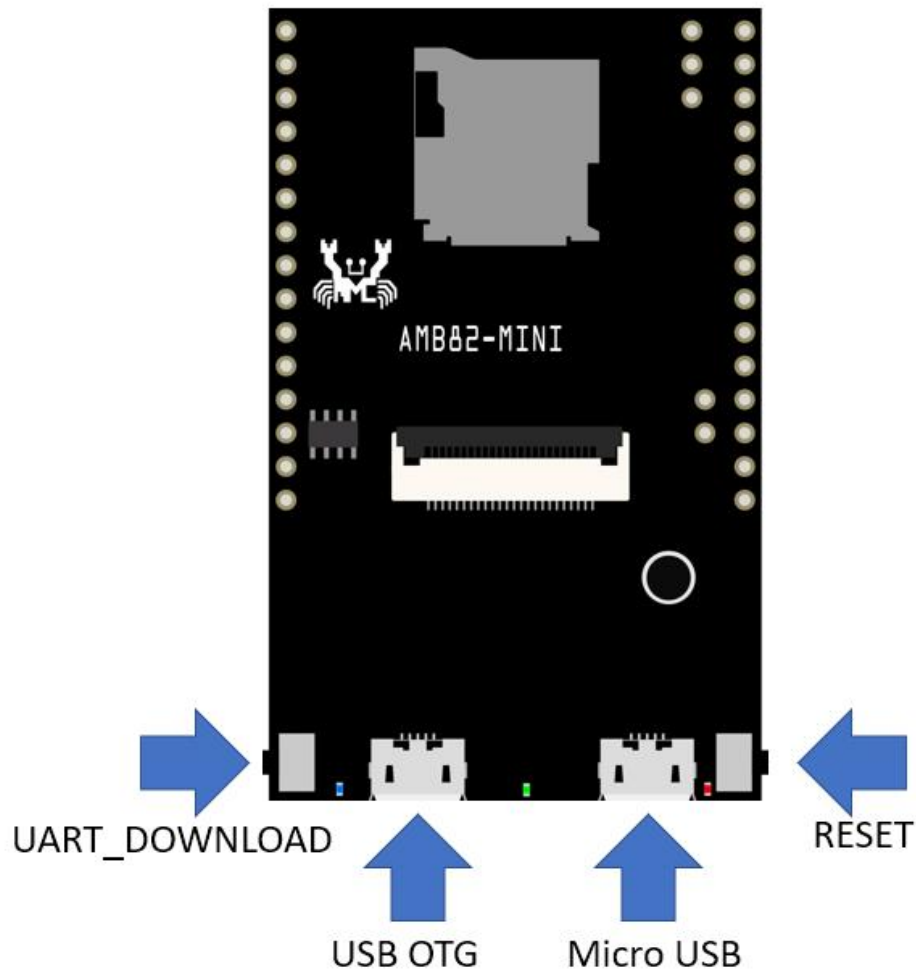
AMB82 mini

twgo.io/amb82

AMB82 mini

官方介绍：

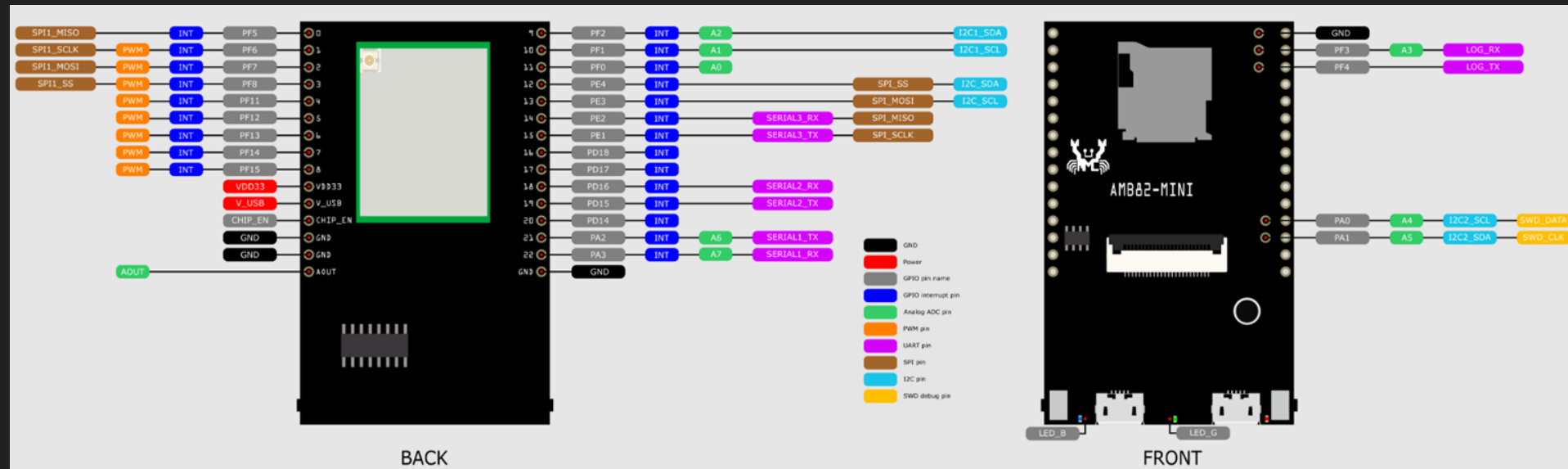
<https://www.amebaiot.com/zh/amebapro2-amb82-mini-arduino-getting-started/>



特點

- 500萬畫素
- 內建WIFI (2.4G) 、 BLE
- 支援I2C、UART、SPI、PWM、ADC
- 內建音頻編解碼器、視頻編解碼器(H264, H265)和 NN (內置 NPU)
- USB OTG：可連接讀取USB裝置，例如鍵盤滑鼠Webcam
- 購買網址：<https://twgo.io/nmksp>

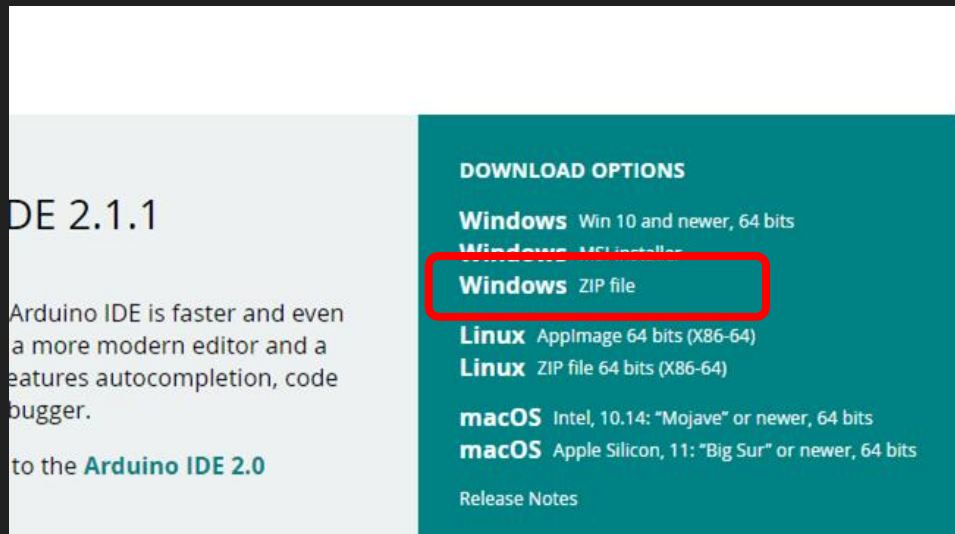
腳位圖



環境設定

安裝Arduino 2.0

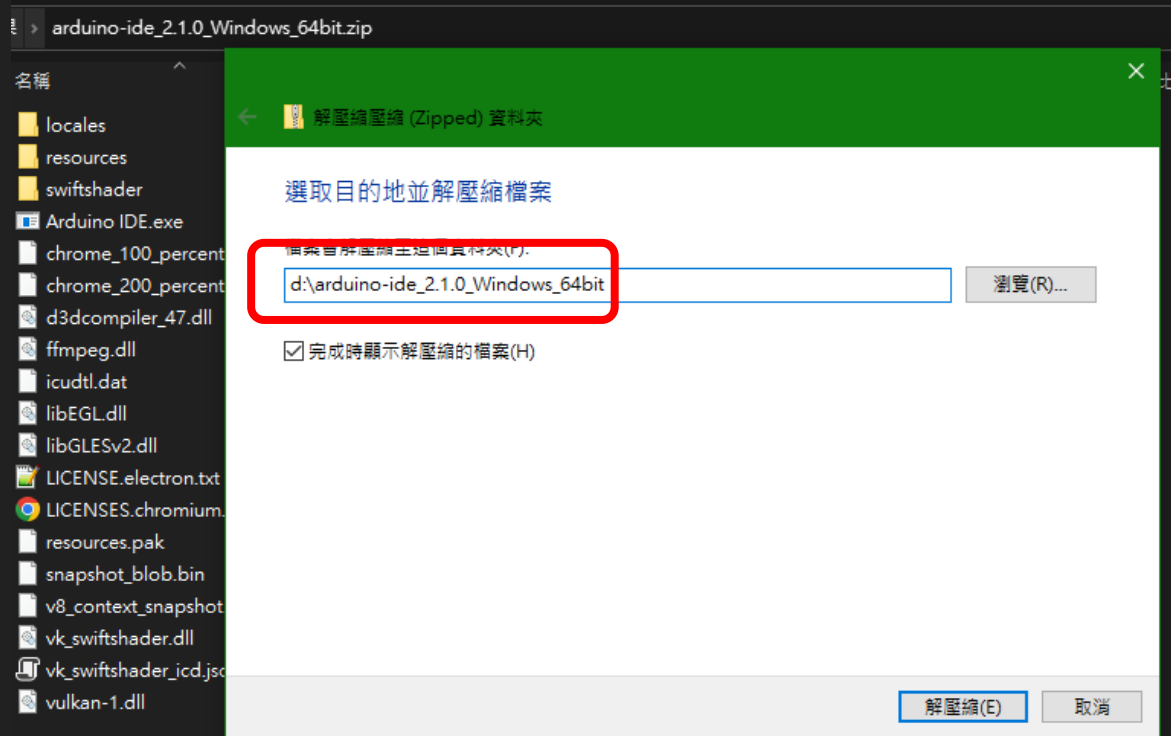
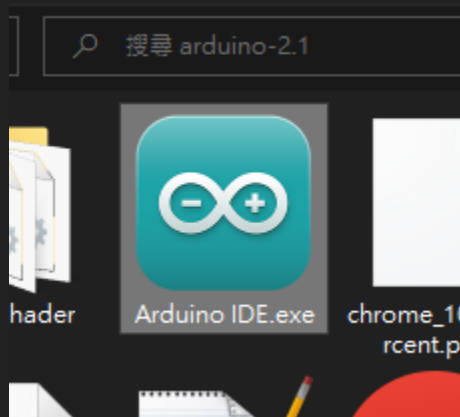
- 下載Arduino 2.0網址：
<https://www.arduino.cc/en/software>
- Google雲端：<https://twgo.io/yczdi>
- 建議選擇windows Zip
- 關於2.0與1.8的比較可以參考我的教學：
<https://youtube.com/live/FBzjly3ModE>



解壓縮

解壓縮到C或D的根目錄

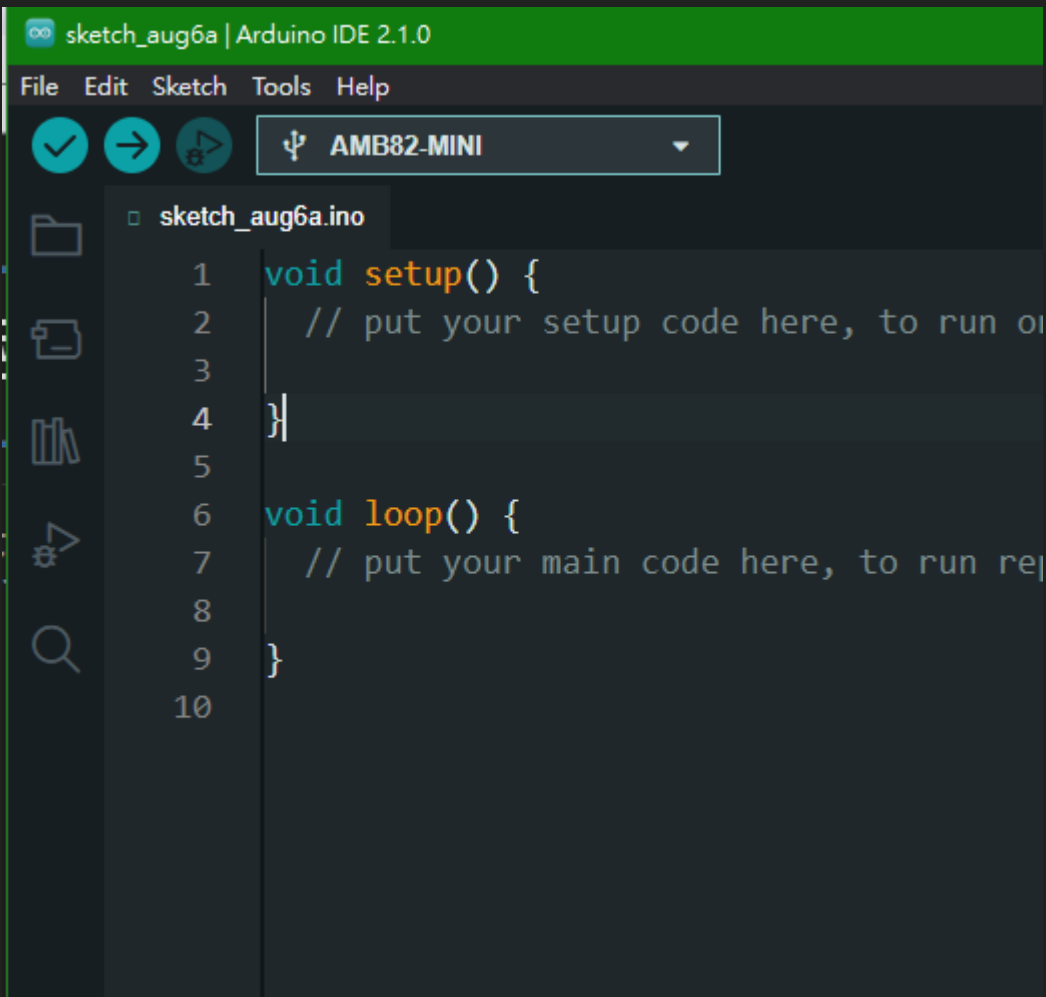
點選主程式即可開啟



執行主程式

setup()：本部份只會執行一次

loop()：重複執行不停止，無窮迴圈



The screenshot shows the Arduino IDE 2.1.0 interface. The title bar reads 'sketch_aug6a | Arduino IDE 2.1.0'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar, there are three circular icons: a checkmark, a right arrow, and a right arrow with a bug icon. To the right of these icons is a dropdown menu showing 'AMB82-MINI'. The main editor area displays the code for 'sketch_aug6a.ino'. The code is as follows:

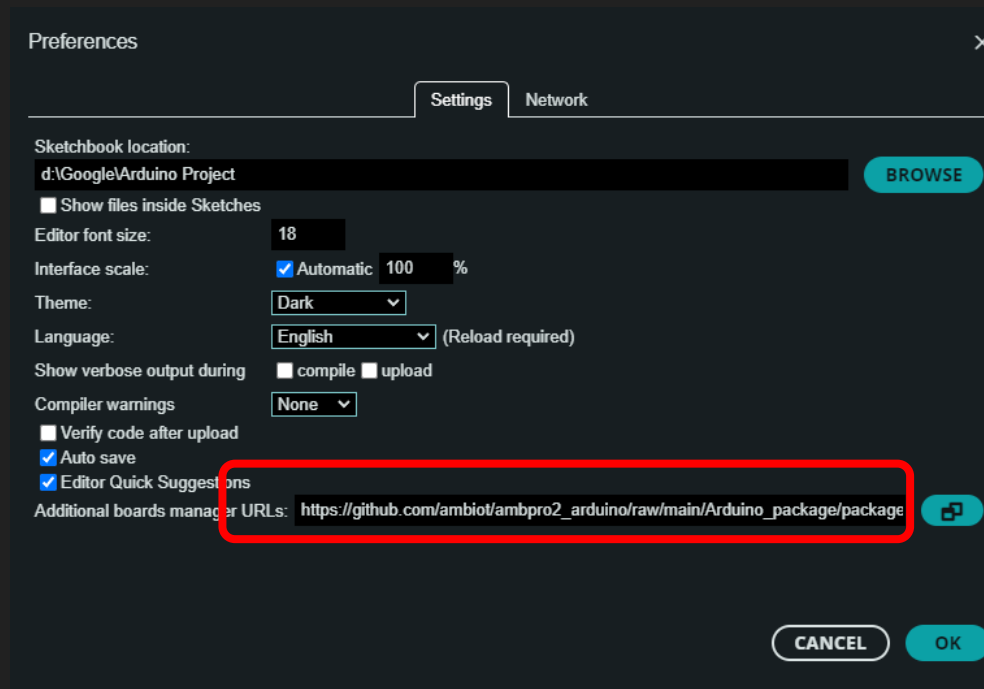
```
1 void setup() {  
2   // put your setup code here, to run once  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly  
8  
9 }  
10
```


安裝開發板

功能表/file/preferences/

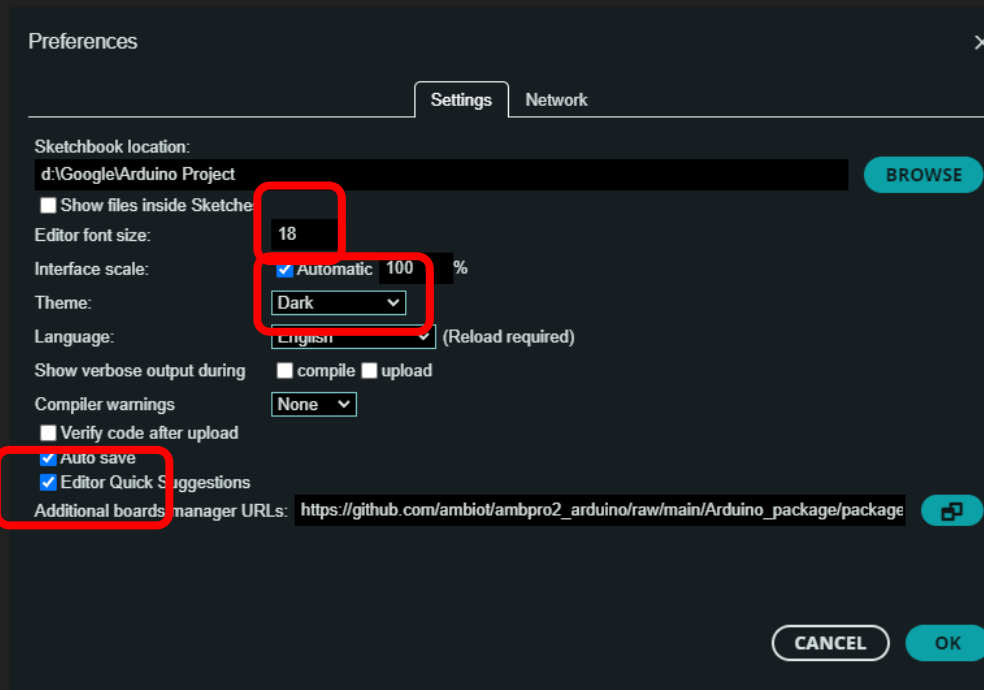
貼上以下內容在空白處：

https://github.com/ambiot/ambpro2_arduino/raw/main/Arduino_package/package_realtek_ameba_pro2_index.json



順便修改選項

1. 選擇黑色背景
2. 選擇合適的大字型
3. 勾選Editor Quick Suggestions

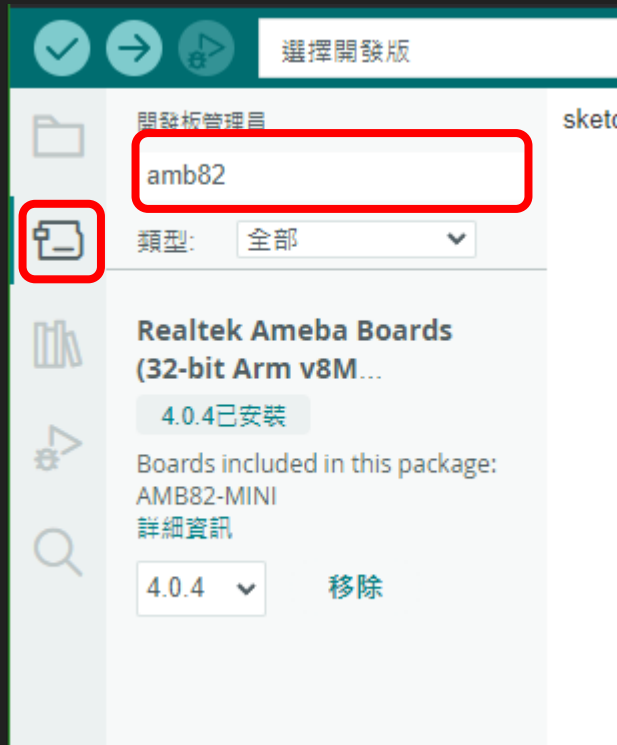


安裝開發版核心

使用左手開發版圖示

輸入關鍵字 AMB82

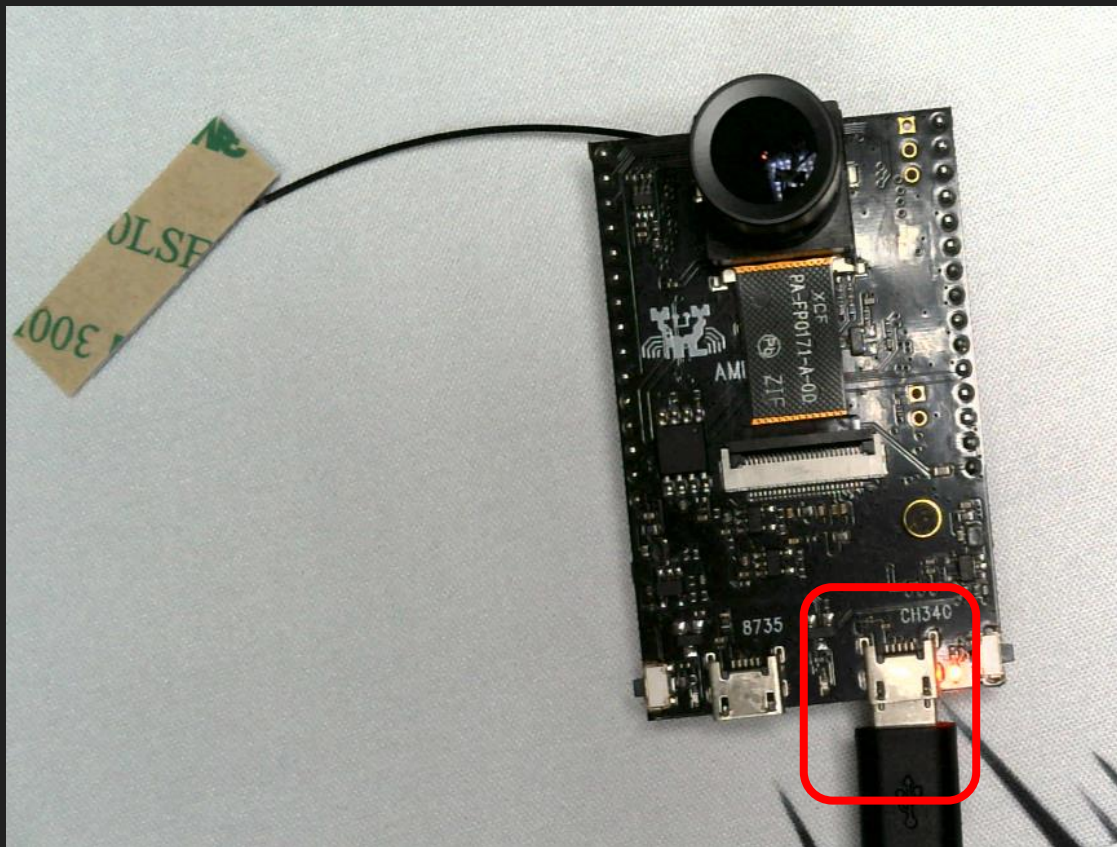
完成安裝



插入USB線

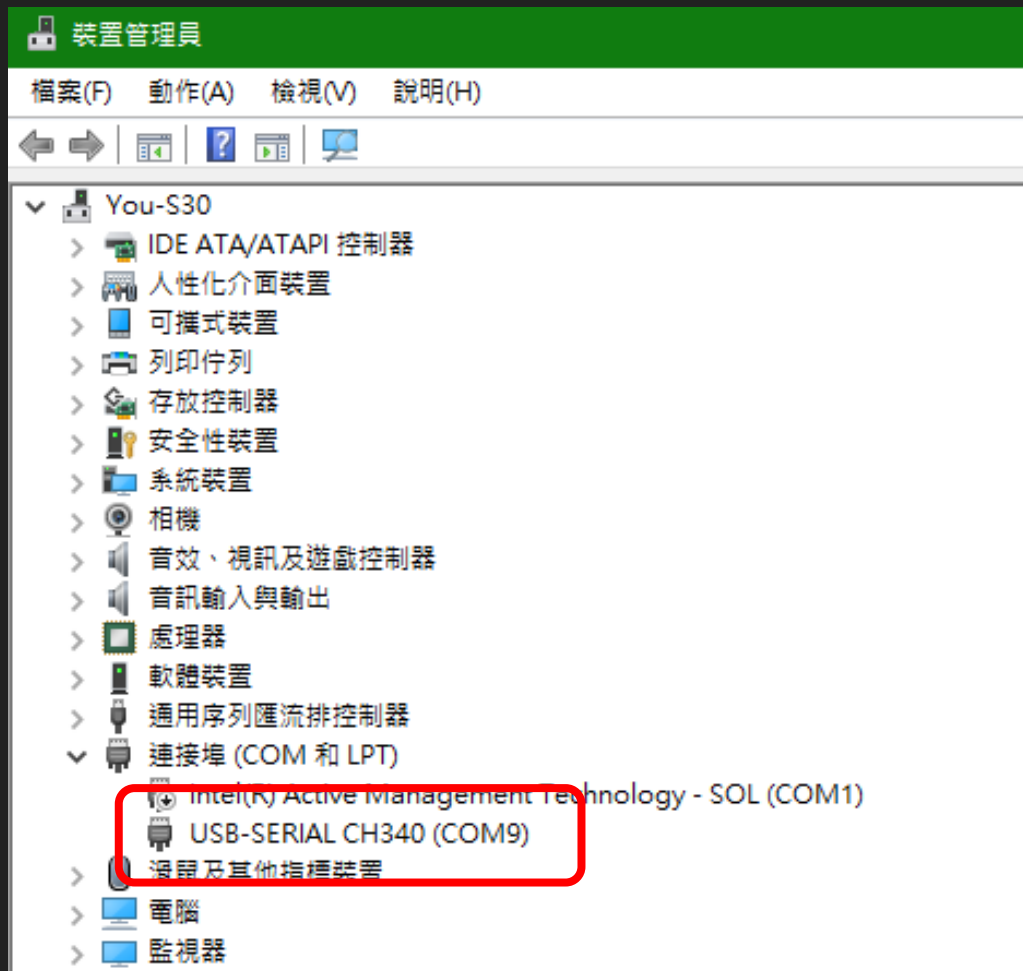
CH340驅動下載：

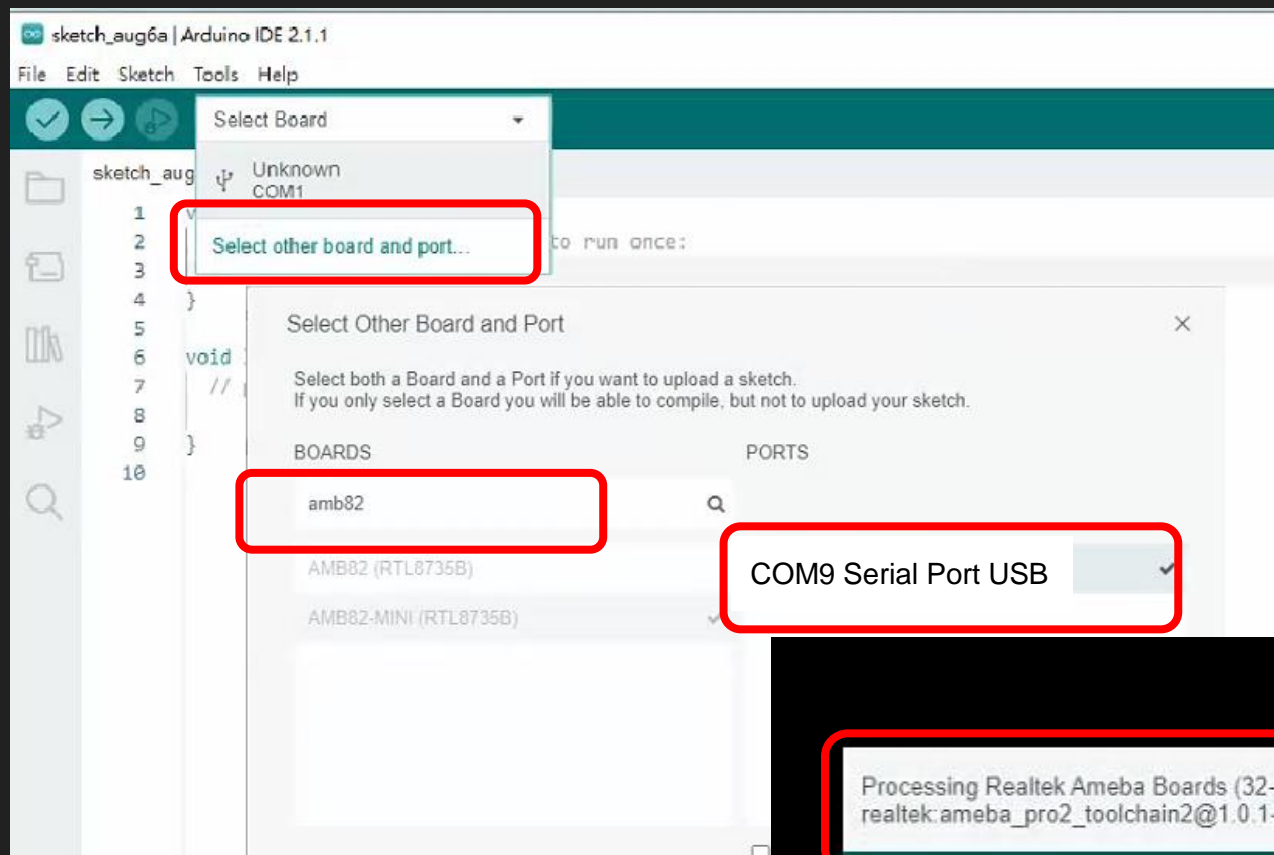
twgo.io/ch340



查看裝置管理員

確認Ch340取得的連接埠編號

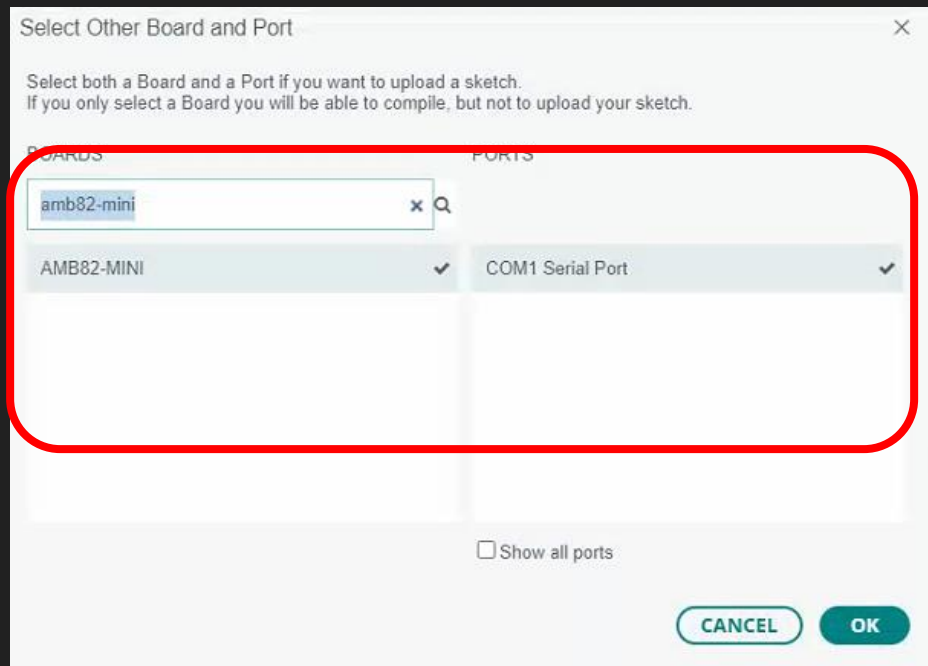
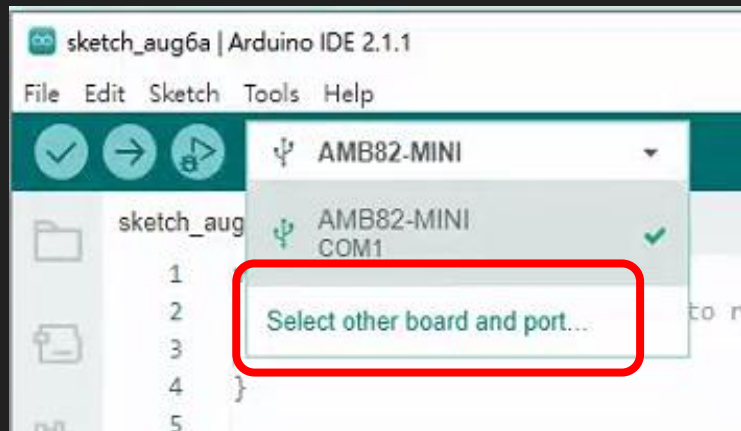




Processing Realtek Ameba Boards (32-bit Arm v8M @500MHz):4.0.4:
realtek:ameba_pro2_toolchain2@1.0.1-p2

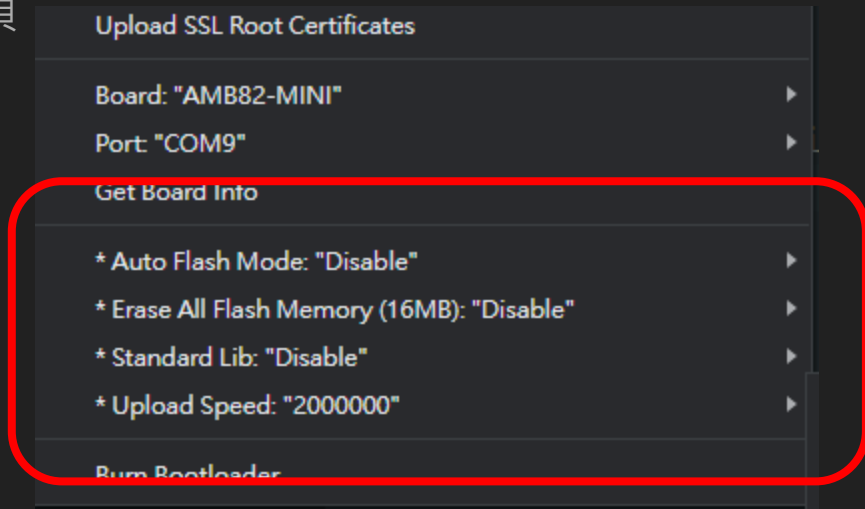
Ln 3, Col 1 AMB82-MINI (RTL8735B) on COM1

確認是否能選擇



調整燒錄選項

功能表/Tools查看選項



基礎練習

Blink

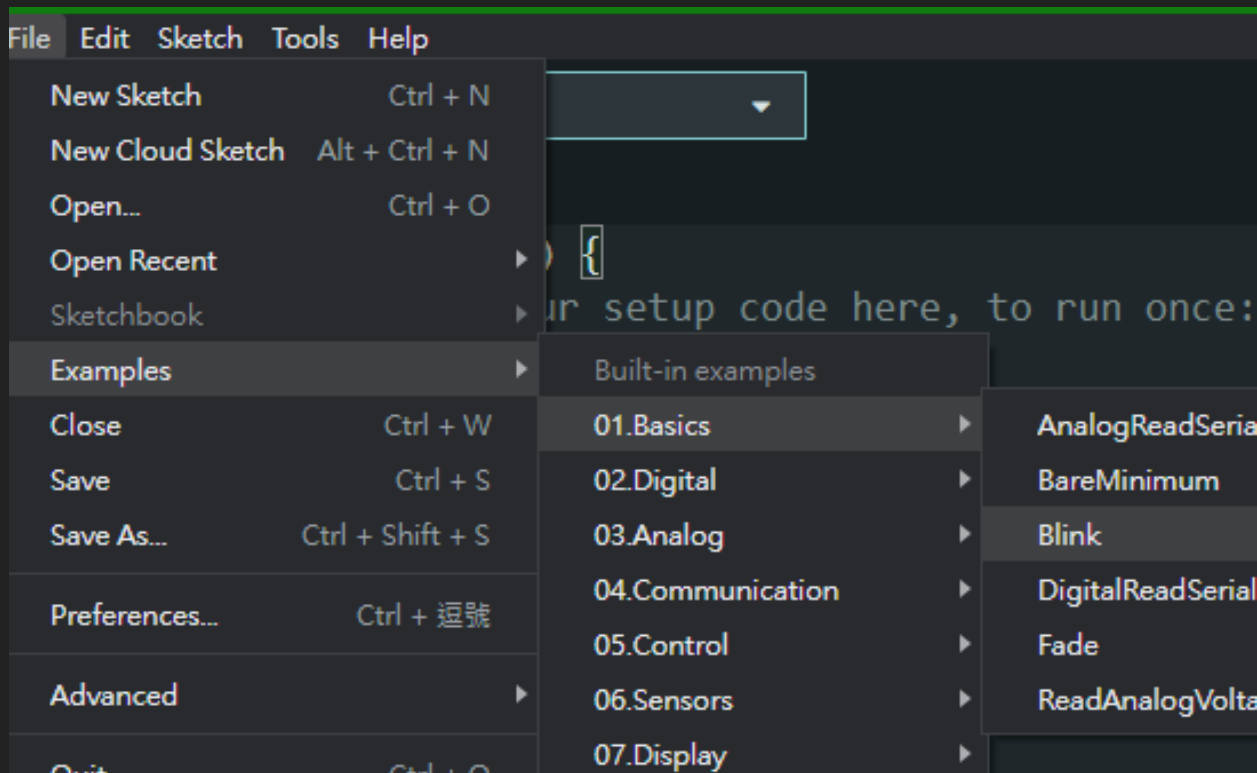
開啟範例

開啟範例程式Blink

AMB82有兩顆LED，分別是

LED_B=LED_BUILTIN=23

LED_G=24



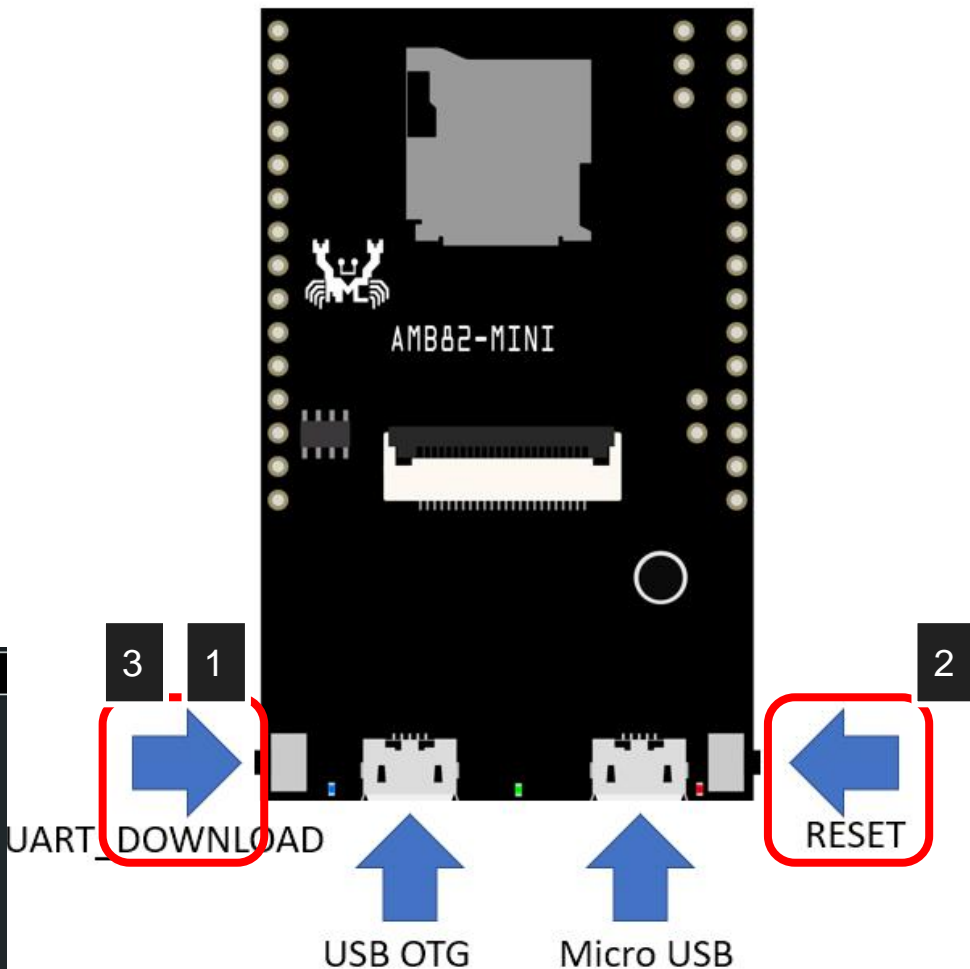
轉換燒錄模式

1. 先按住左側「下載按鈕」不放
2. 按右側「重開機按鈕」一下
馬上放開「重開機按鈕」
3. 最後再放開「下載按鈕」

完成模式轉換，藍燈會慢慢亮起

Message (Enter to send message to 'AMB82-MINI' on 'COM9')

```
21:48:58.302 ->  
21:48:58.302 -> == Rtl8735b IoT Platform ==  
21:48:58.302 -> Chip VID: 0, Ver: 0  
21:48:58.302 -> ROM Version: v3.0  
21:48:58.302 -> Test Mode: boot_cfg1=0x0  
21:48:58.302 ->  
21:48:58.302 -> [test mode PG]  
21:48:58.302 -> test_mode_img_download  
21:48:58.302 -> Download Image over UART1[tx=4,rx=3] baud=115200
```



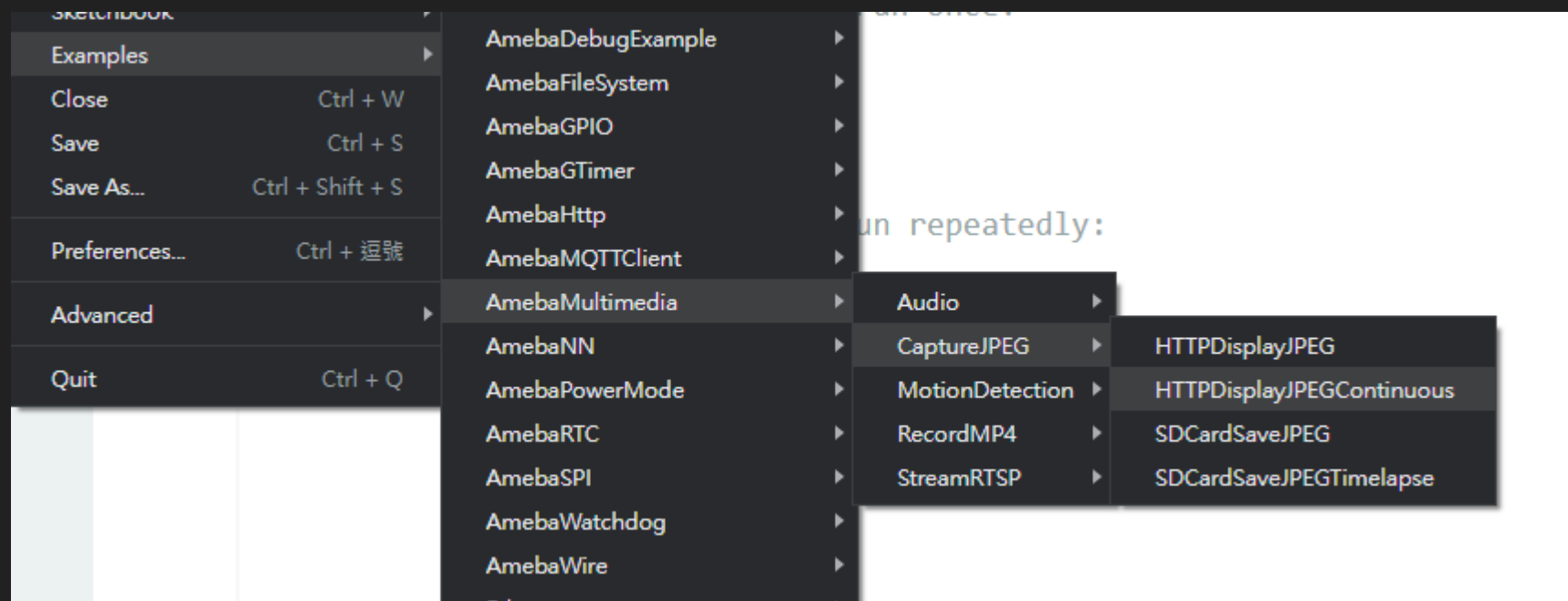
上傳過程

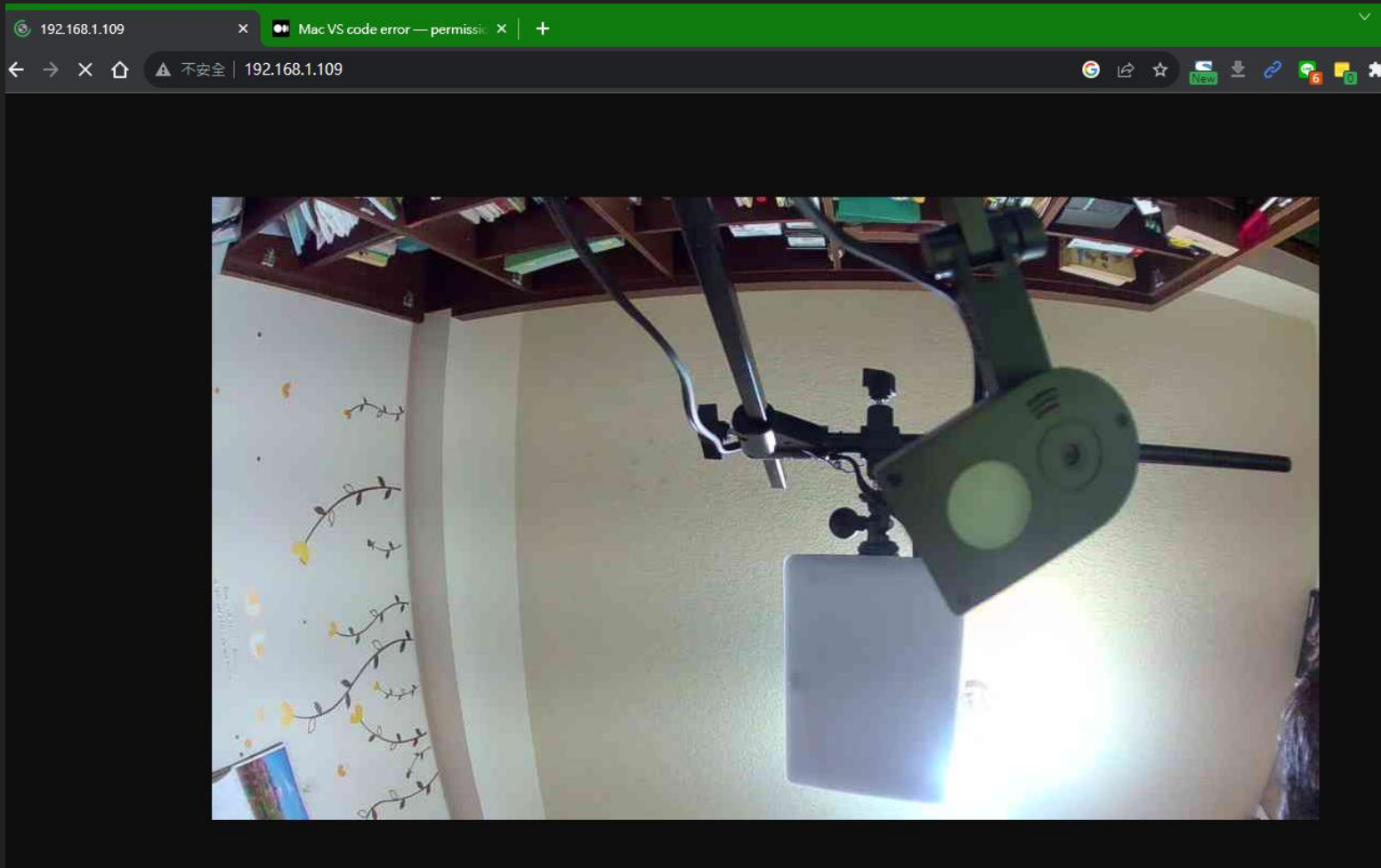
燒錄完畢，還要按右側**RST**按鈕 重開機，才會開始執行

Output

```
Sketch uses 4284416 bytes (25%) of program storage space. Maximum is 16777216 by
Enter Auto Flash Mode!
Start Upload Flash
| Uploading.....upload success
End Upload Flash
```

JPEG Stream



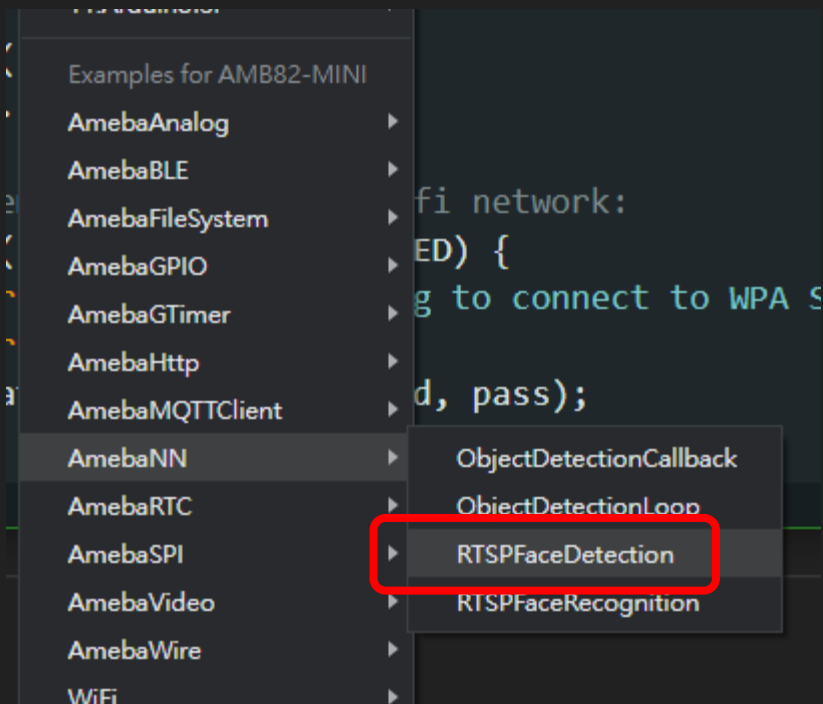


使用RTSP協定 (H264壓縮+聲音)

RTSP協定

選擇人臉偵測範例

開啟範例：AmebaNN/RTSPFaceDetection



修改範例

修改Wifi設定
(30、31行)

```
20 #define NNWIDTH 576
21 #define NNHEIGHT 320
22
23 VideoSetting config(VIDEO_FHD, 30, VIDEO_H264, 0);
24 VideoSetting configNN(NNWIDTH, NNHEIGHT, 10, VIDEO_RGB, 0);
25 NNFaceDetection facedet;
26 RTSP rtsp;
27 StreamIO videoStreamer(1, 1);
28 StreamIO videoStreamerNN(1, 1);
29
30 char ssid[] = "You2F"; // your network SSID (name)
31 char pass[] = "0933932774"; // your network password
32 int status = WL_IDLE_STATUS;
33
34 void setup() {
```

查看網路影像

下載VLC 可觀看RTSP播放器：<https://www.videolan.org/vlc/>

媒體/開啟網路串流

填上：

rtsp://你的IP位置:554



查看結果



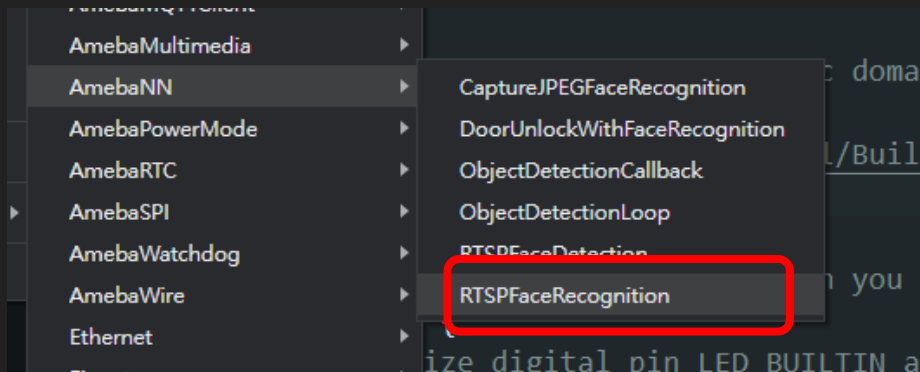
使用內建的人臉辨識

選擇人臉辨識範例

開啟範例：

AmebaNN/RTSPFaceRecognition

臉部辨識的紀錄僅存於記憶體，重開機後會消失。



註冊臉部

還沒註冊時，所有臉部都是未知 (UNKNOWN)



註冊臉部

在序列視窗空白處輸入 **REG=英文名字**
(註冊時僅能有一個臉部)

註冊後等約5秒即可出現辨識結果



傳LINE影像

傳LINE通知

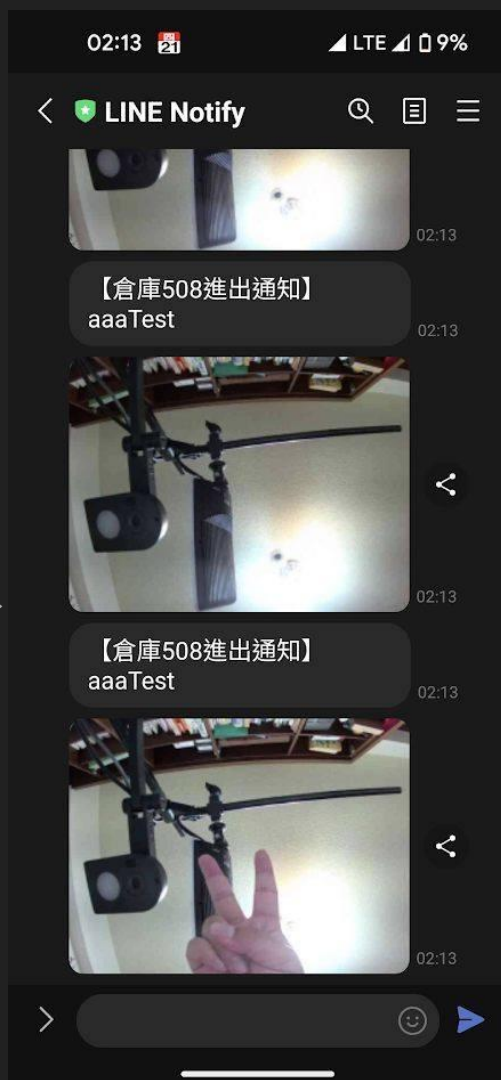
1. 申請LINE Token密碼

- https://notify-bot.line.me/zh_TW/

2. 程式碼：<https://twgo.io/aplzo>

3. 結合前一節人臉辨識功能，看到不認識的人

程式碼：<https://twgo.io/byiiw>



傳MQTT影像

傳MQTT

MQTT透過中介方式轉傳資訊，讓資料可以於廣域網路通訊，類似eMAIL信箱的概念

伺服器：MQTTGO.io

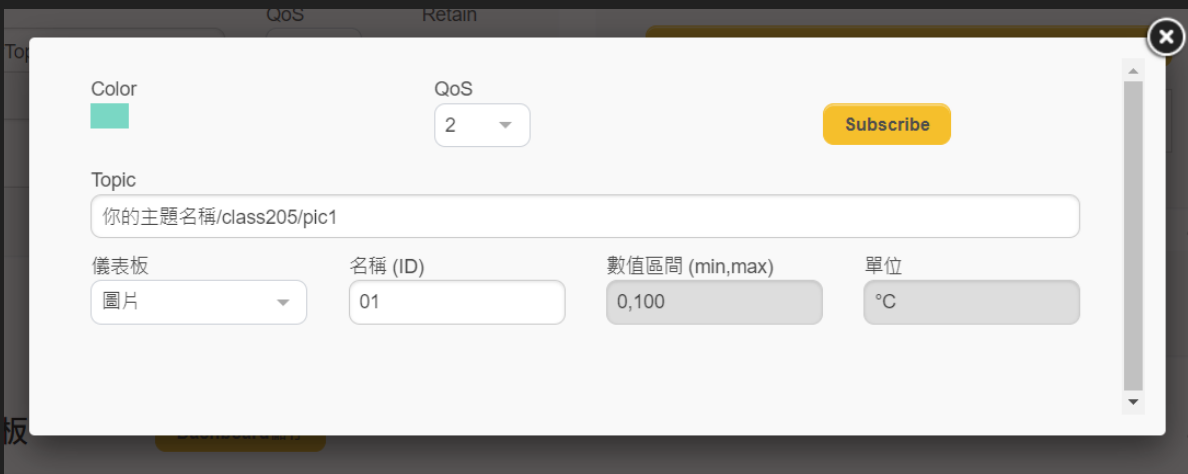
信箱：主題可自取名，原則以斜線區分階層【 / 】

例如：艾瑞克/AI教室/即時影像 或 alice/class205/image

2. 程式碼：<https://twgo.io/vwder>

查看MQTT照片(手機也可以)

1. 使用瀏覽器輸入網站：mqttgo.io
2. 先點連線
3. 點選訂閱，輸入Arduino所輸入MQTT影像主題的名稱後，點選圖片，並修改適合的名稱(ID)，ID每一個圖表都需不相同



A screenshot of the MQTTgo.io subscription form. The form is white with a grey border and a close button (X) in the top right corner. It contains the following fields and controls:

- Color:** A color selection button showing a teal square.
- QoS:** A dropdown menu currently set to '2'.
- Subscribe:** A yellow button to confirm the subscription.
- Topic:** A text input field containing '你的主題名稱/class205/pic1'.
- 儀表板 (Dashboard):** A dropdown menu currently set to '圖片' (Image).
- 名稱 (ID):** A text input field containing '01'.
- 數值區間 (min,max):** A text input field containing '0,100'.
- 單位:** A text input field containing '°C'.



使用內建的YOLO模型

支援的模型列表

YOLOv3 model: DEFAULT_YOLOV3TINY, CUSTOMIZED_YOLOV3TINY

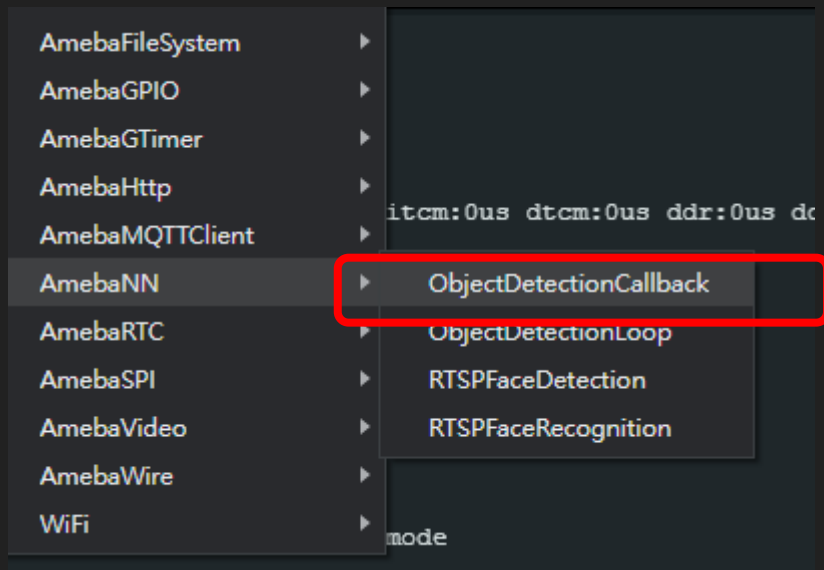
YOLOv4 model: DEFAULT_YOLOV4TINY, CUSTOMIZED_YOLOV4TINY

YOLOv7 model: DEFAULT_YOLOV7TINY, CUSTOMIZED_YOLOV7TINY

Valid Face Detection model: DEFAULT_SCRFD, CUSTOMIZED_SCRFD

開啟範例檔

file/example/AmebaNN/ObjectDetectionCallback (發現物件時產生回呼)



使用VCL播放器查看結果



將結果列表顯示

<https://docs.google.com/document/d/1bfalHNVN0tvxb7te3uTutnDYS-TzqyLkQKZObRqE7Rw/edit>

Message (Enter to send message to 'AMB82-MINI' on 'COM3')

```
13:53:21.280 -> font of char 0000007b not exist
13:53:21.280 -> font of char 0000002c not exist
13:53:21.280 -> font of char 0000007d not exist
13:53:21.280 -> YOLOv4t tick[0] = 71
13:53:21.451 -> Total number of objects detected = 3
13:53:21.451 -> Item 0 bus:      592 918 602 924
13:53:21.451 -> Item 1 car:      612 899 563 923
13:53:21.451 -> Item 2 car:      1247 1241 124 288
13:53:21.451 -> {bus:1 , car:2 , }
13:53:21.497 -> font of char 0000007b not exist
13:53:21.497 -> font of char 0000002c not exist
13:53:21.497 -> font of char 0000002c not exist
13:53:21.497 -> font of char 0000007d not exist
```

計算影像中的人數，顯示在畫面的左上角

程式碼：<https://twgo.io/davws>

使用錢幣辨識模型

錢幣辨識模型

網址：<https://github.com/wildman8606/HUB-8735-Coin-Detection-tutorial/tree/master>

使用自己建好的YOLO模型

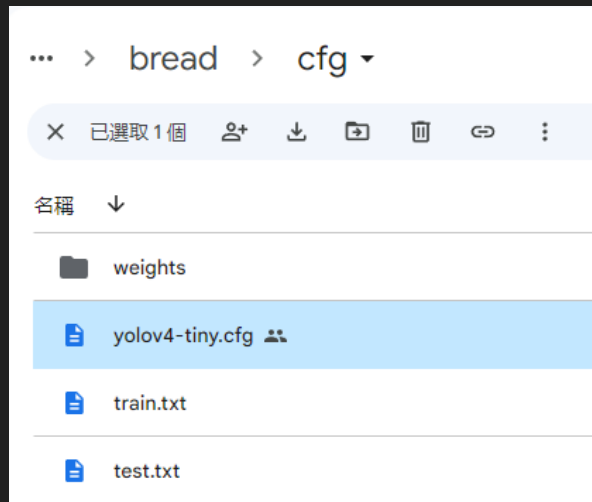
從訓練結果下載YOLO權重檔及設定檔

訓練結果應該在colab notebooks/darknet/bread/cfg，包含權重檔與設定檔

權重檔：weights/yolov4-tiny.weights

設定檔：yolov4-tiny.cfg

老師分享：<https://twgo.io/nkidl>



透過線上轉換檔案格式(YOLO->NB)

註冊並登入realtech AMB網站

線上轉換網址：<https://www.amebaiot.com/zh/amebapro2-ai-convert-model/>

可將YOLO模型(.weight, cfg)轉成AMB專屬的.nb檔案

已轉換的模型：<https://twgo.io/zvcgn>

E-mail (required, After submission, we will send out download link to you through email.)

Confirm E-mail (required)

Model (required, YOLO-TINY or SCRFD or MobileFaceNet)

YOLO-TINY

Quantize Type (required, UINT8 or INT16)

UINT8

Upload zip file including a cfg file and a weights file(required, please upload the file contained the ".cfg" and ".weights" files, all named in English, limit:35MB)

選擇檔案 未選擇任何檔案

Upload one jpg file (required, limit:1MB)

選擇檔案 未選擇任何檔案

file contained the ".cfg" and ".weights" files, all named in English, limit:35MB)

選擇檔案 未選擇任何檔案

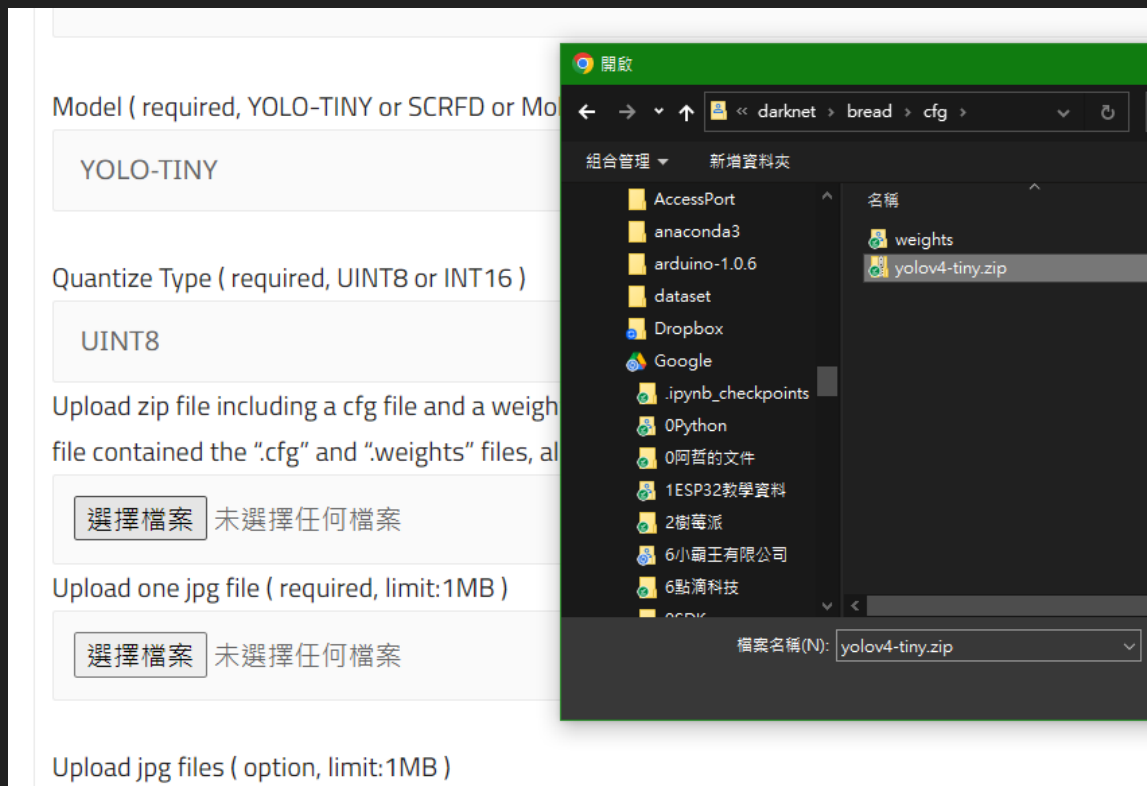
Upload one jpg file (required, limit:1MB)

選擇檔案 未選擇任何檔案

Upload jpg files (option, limit:1MB)

選擇檔案 未選擇任何檔案

上傳訓練好的模型檔



模型精細度

目前提供兩種精細度

UINT8：不帶符號8 bit整數，模型約4-5M，速度約5-6 fps

INT16：帶符號16位元整數，模型約9-10M，速度約2-3 fps



Quantize Type (required, UINT8 or INT16)

INT16
UINT8
INT16

選擇檔案 未選擇任何檔案

Upload one jpg file (required, limit:1MB)

Amebapro2 AI 模型轉換

完成後將發送電子郵件通知。

單擊通知電子郵件中的連結來下載您的文件。

收取MAIL內的檔案



取代原本的YOLO模型檔案

確認使用YOLOv4-tiny

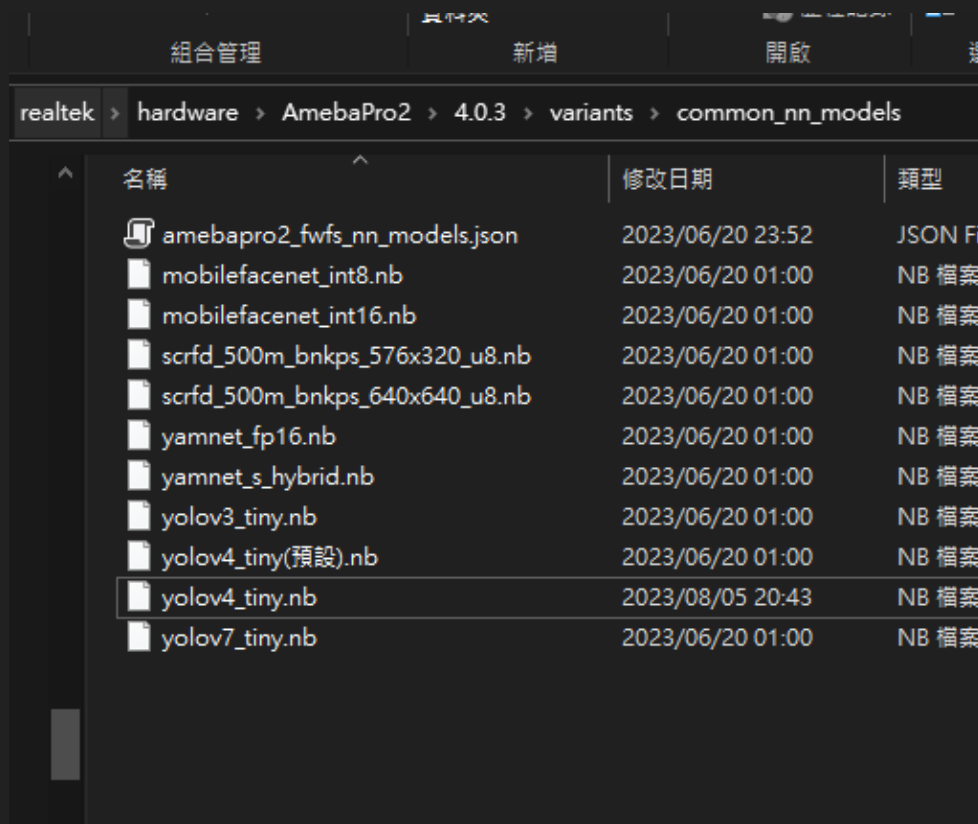
開啟檔案總管，尋找預設的檔案位置

C:\Users\你的Win帳號

\AppData\Local\Arduino15\packages\
realtek\hardware\AmebaPro2\版本
\variants\common_nn_models

先備份yolov4_tiny.nb，例如更名為
yolov4_tiny(預設).nb

再將下載的network_binary.nb，更名為
yolov4_tiny.nb，貼到資料夾內。



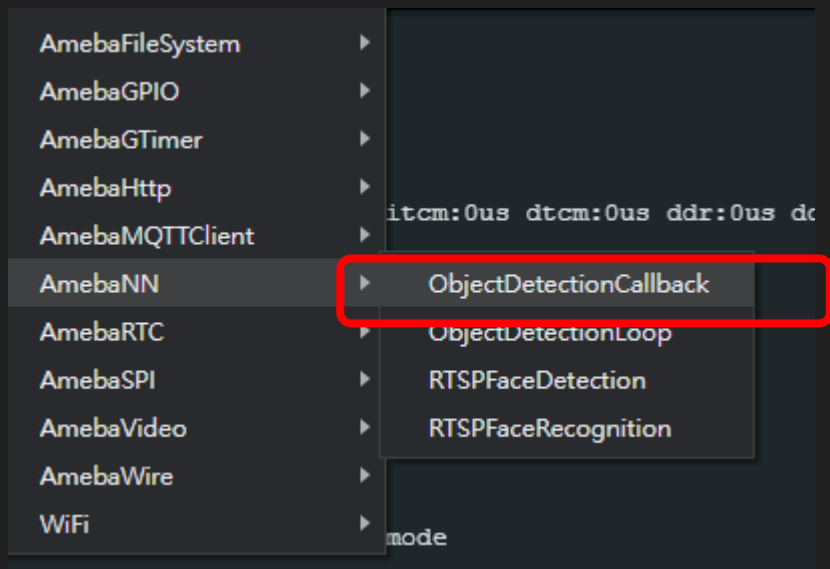
可暫時用老師的**NB**檔測試

網址：

https://drive.google.com/file/d/14zCSDeaJhZAlOOOnafVAutlWXdk8Fncv/view?usp=drive_link

開啟Arduino 2選擇範例

修改48,49行WiFi的設定



```
45 StreamIO videoStreamer(1, 1);
46 StreamIO videoStreamerNN(1, 1);
47
48 char ssid[] = "a"; // your network SSID (name)
49 char pass[] = "12345678"; // your network password
50 int status = WL_IDLE_STATUS;
51
52 IPAddress ip;
53 int portnum;
```

修改物件列表

將YOLO訓練時的obj.names開啟後，依需要填入

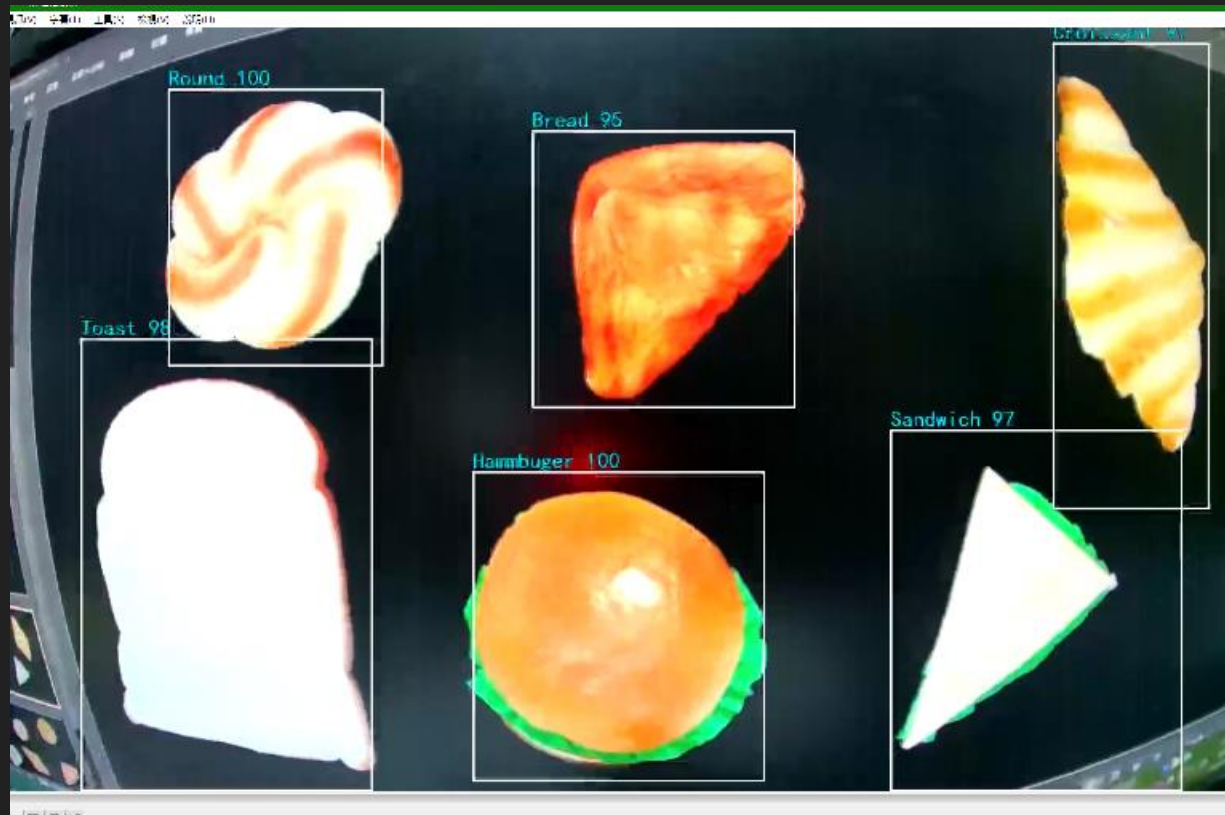
1. 修改數量為6
2. 修改名稱，順序必須依照當時的obj.names

```
1 imgFolder = "images"
2 xmlFolder = "labels"
3 saveYoloPath = "yolo"
4 classes = 6
5 classList = { "B":0, "C":1, "H":2, "T":3, "S":4, "R":5} #類別及編號
6
```

```
ObjectDetectionLoop.ino  ObjectClassList.h

1 struct ObjectDetectionItem {
2     uint8_t index;
3     const char* objectName;
4     uint8_t filter;
5 };
6
7 // List of objects the pre-trained model
8 // Index number is fixed and hard-coded
9 // Set the filter value to 0 to ignore a
10 ObjectDetectionItem itemList[6] =
11 {
12     { 0, "Bread", 1 },
13     { 1, "Croissant", 1 },
14     { 2, "Hamburger", 1 },
15     { 3, "Round", 1 },
16     { 4, "Sandwich", 1 },
17     { 5, "Toast", 1 }
18 };
```

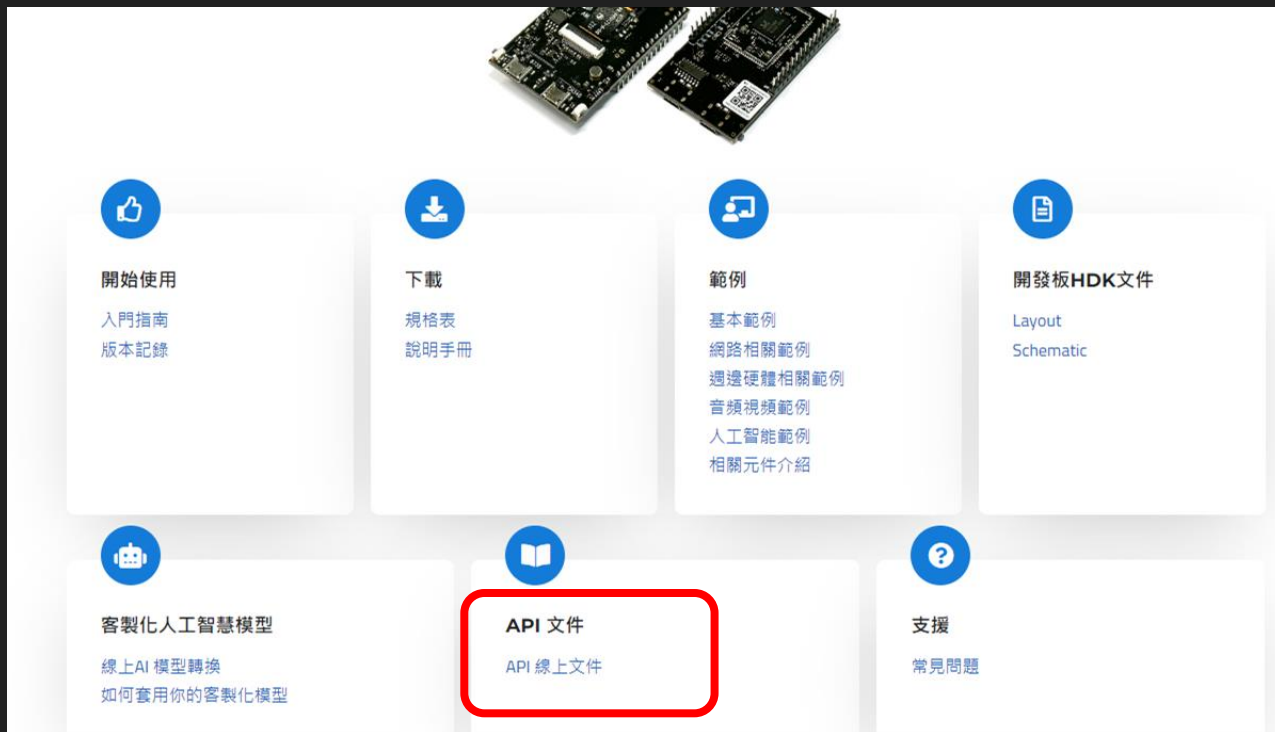
查看結果



程式結構分析

AMB82-MINI API網址

<https://www.amebaiot.com/zh/ameba-arduino-summary/>



AI辨識的位置

```
129
130  if (ObjDet.getResultCount() > 0) {
131      for (uint32_t i = 0; i < ObjDet.getResultCount(); i++) {
132          int obj_type = results[i].type();
133          if (itemList[obj_type].filter) { // check if item should be ignored
134
135              ObjectDetectionResult item = results[i];
136              // Result coordinates are floats ranging from 0.00 to 1.00
137              // Multiply with RTSP resolution to get coordinates in pixels
138              int xmin = (int)(item.xMin() * im_w);
139              int xmax = (int)(item.xMax() * im_w);
140              int ymin = (int)(item.yMin() * im_h);
141              int ymax = (int)(item.yMax() * im_h);
142
143              // Draw boundary box
144              printf("Item %d %s:\t%d %d %d %d\n\r", i, itemList[obj_type].objectName, xmin, xmax, ymin, ymax);
145              OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_WHITE);
146
147              // Print identification text
148              char text_str[20];
149              snprintf(text_str, sizeof(text_str), "%s %d", itemList[obj_type].objectName, item.score());
150              OSD.drawText(CHANNEL, xmin, ymin - OSD.getTextHeight(CHANNEL), text_str, OSD_COLOR_CYAN);
151          }
152      }
153  }
154  OSD.update(CHANNEL);
```

作法

修改ObjectClassList.h

1. 修改結構，新增一個int Price
用來紀錄產品價格
2. 在產品的後方，多一個屬性
為該產品的價格

ObjectDetectionLoop.ino ObjectClassList.h

```
1  struct ObjectDetectionItem {
2      uint8_t index;
3      const char* objectName;
4      uint8_t filter;
5      int Price;
6  };
7
8  // List of objects the pre-trained model is
9  // Index number is fixed and hard-coded from
10 // Set the filter value to 0 to ignore any r
11 ObjectDetectionItem itemList[6] = {
12     { 0, "Bread", 1,15},
13     { 1, "Croissant", 1,45 },
14     { 2, "Hamburger", 1,52 },
15     { 3, "Toast", 1,3 },
16     { 4, "Sandwich", 1,28 },
17     { 5, "Round", 1,26 }
18 };
```

1

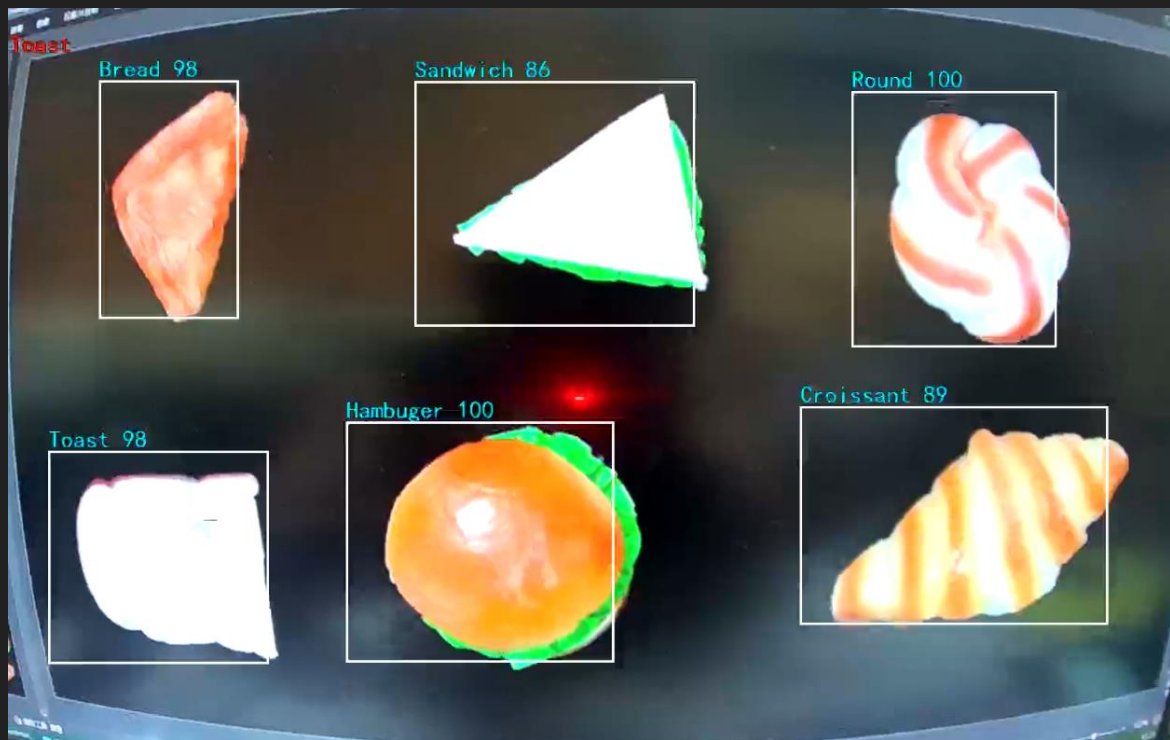
2

OSD drawtext

只能設定一次，因此畫面中只能有一個字形大小16x16或32x32

必須出現在OSD begin之前

最多一次20個字



完成價格計算及修改字形大小

<https://twgo.io/uexyw>

