



AI影像辨識小工程師 辨識模型訓練於自動車應用_拓可思

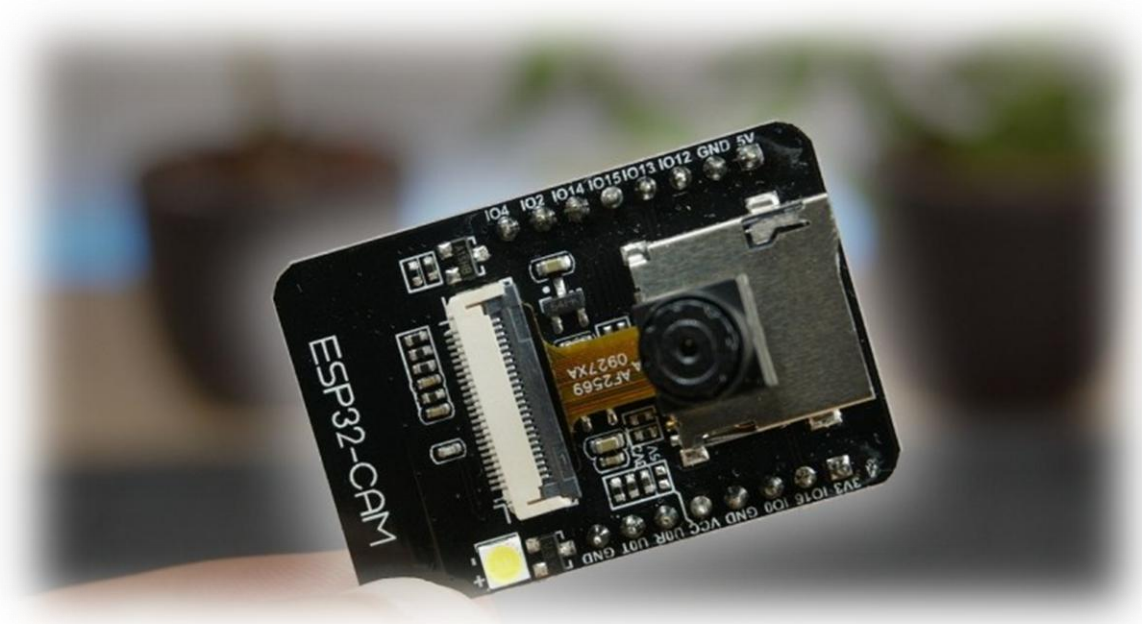
拓可思科創研發

A photograph of three students, two girls and one boy, sitting at a desk in a classroom. They are looking at a large computer monitor on the left, which displays a blue screen with white text and a bar chart. The girl in the foreground is pointing at the screen with her right index finger. The boy is sitting next to her, also looking at the screen. The girl in the background is looking at a tablet device. A large yellow circle is overlaid on the center of the image, containing the text 'AI 影像辨識小工程師' in white.

AI 影像辨識小工程師

學習目標及課程獲得成果

1. 以自動車內容為主題培養在學高中職生習得AI知識及應用體驗，激發學習興趣，提升新世代數位人才的競爭力。
2. 學習使用Arduino控制ESP32, 連接各種感測器裝置及運用，並使用 Google Teachable Machine訓練模型，透過AI辨識結果輸出至程式。
3. Arduino實作練習：將訓練模型觀念應用至BeeCar小車, 做出影像辨識自動會車系統，並延伸思考未來發展的功能面。



認識介面

內建裝置	
WIFI BLE 藍芽 相機 OV2640 TF 卡槽 閃光燈	
特殊腳位	
GPIO0->燒錄用 GPIO1->U0TXD GPIO3->U0RXD GPIO4->閃光燈共用	

工具1
ESP32開發板
Node MCU
機械語言

通訊協定之間的轉換

TTL/UART

USB



工具2
USB to TTL轉換
CH340 (HW-597)模組

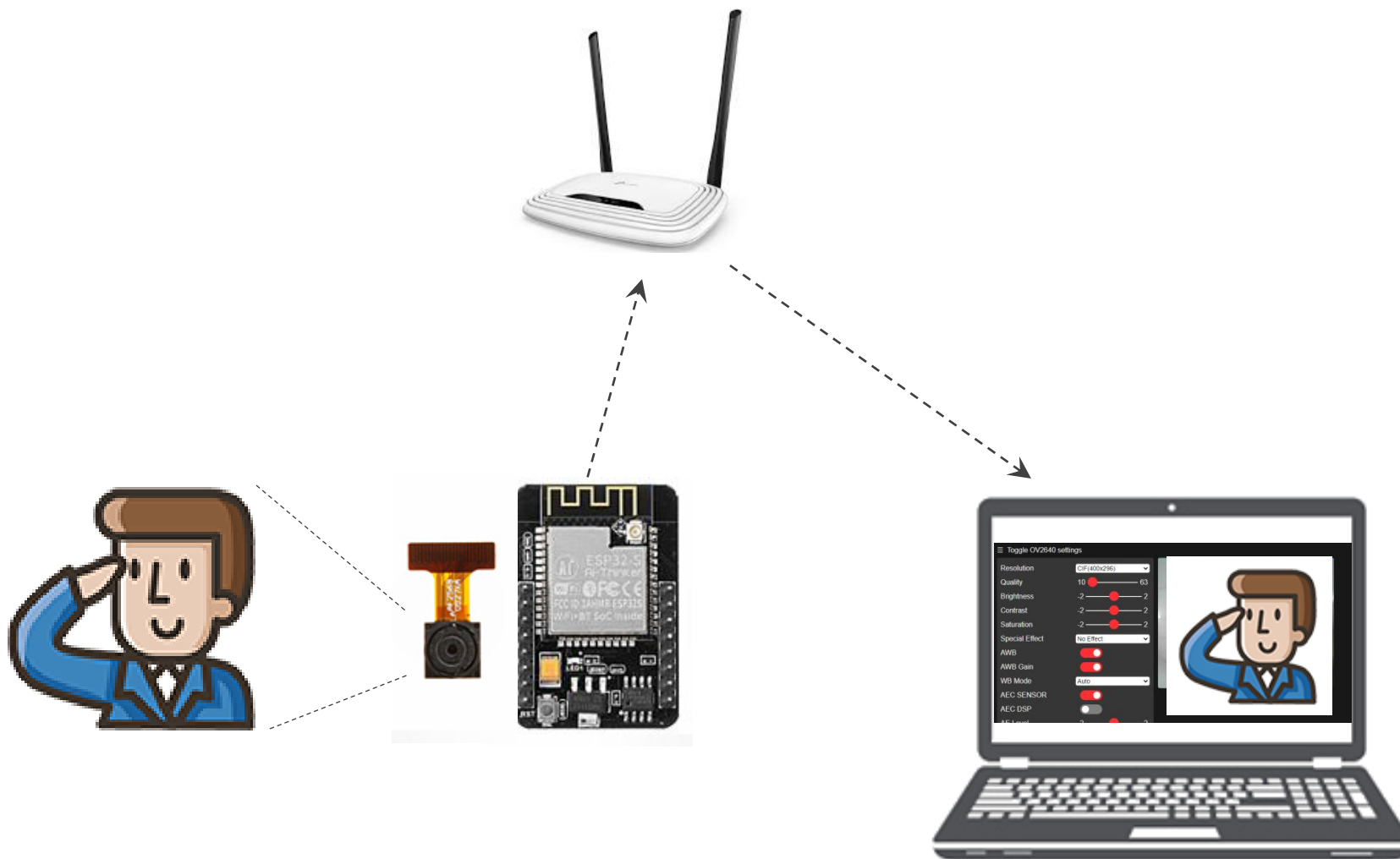


電腦系統系統
MCU: Intel
OS: Windows
Tool: C++
IDE: Arduino IDE

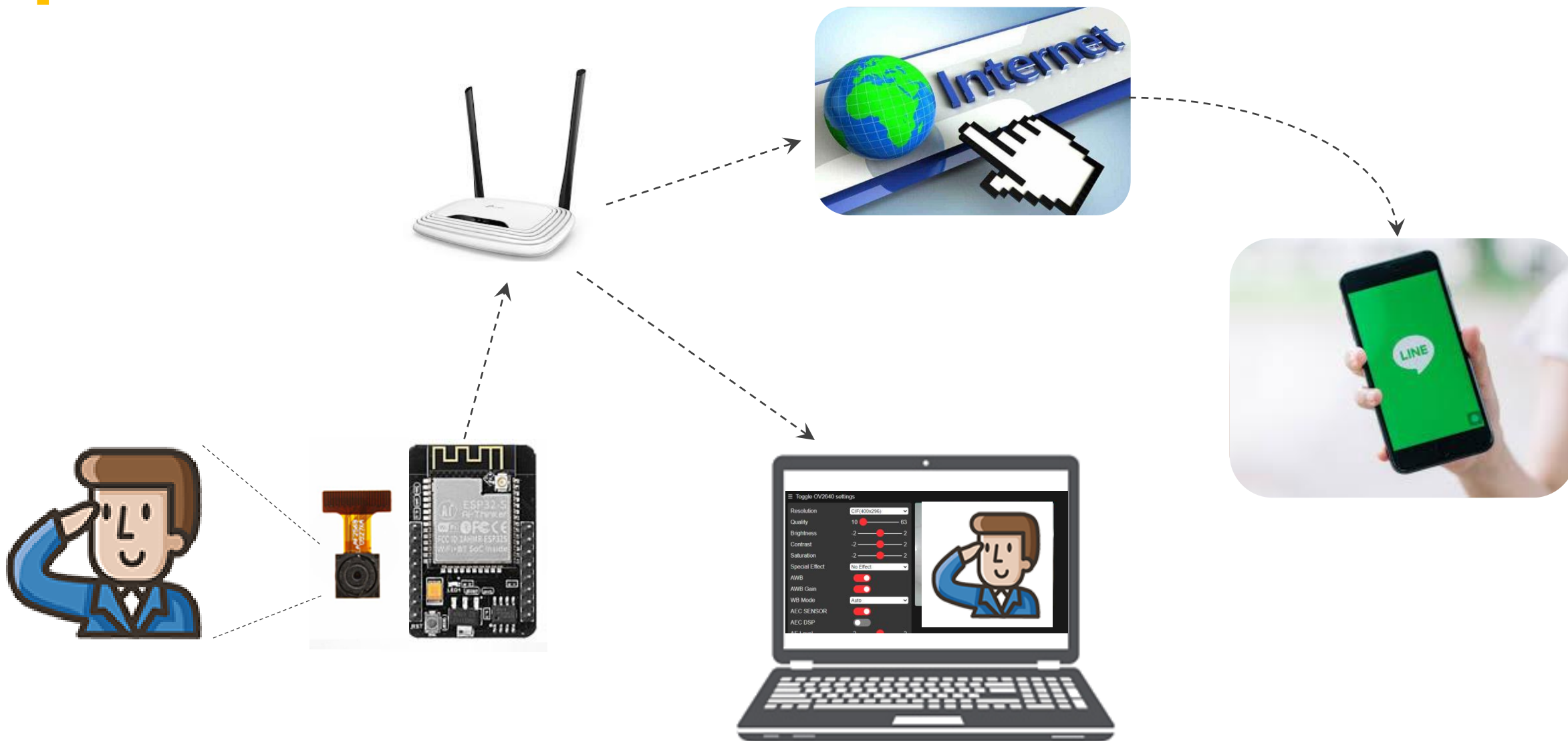


ARDUINO/C語言

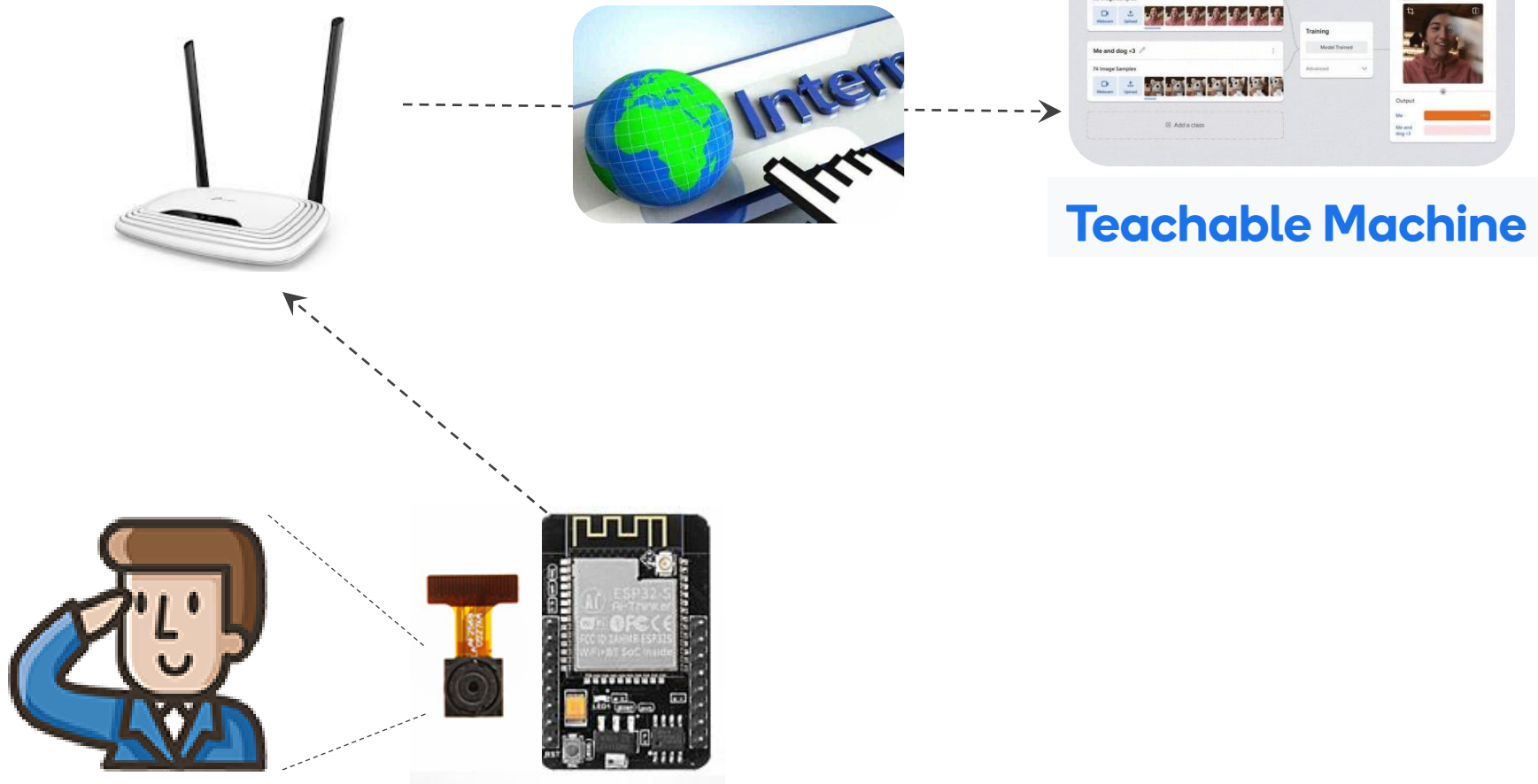
專題實作 Step 1



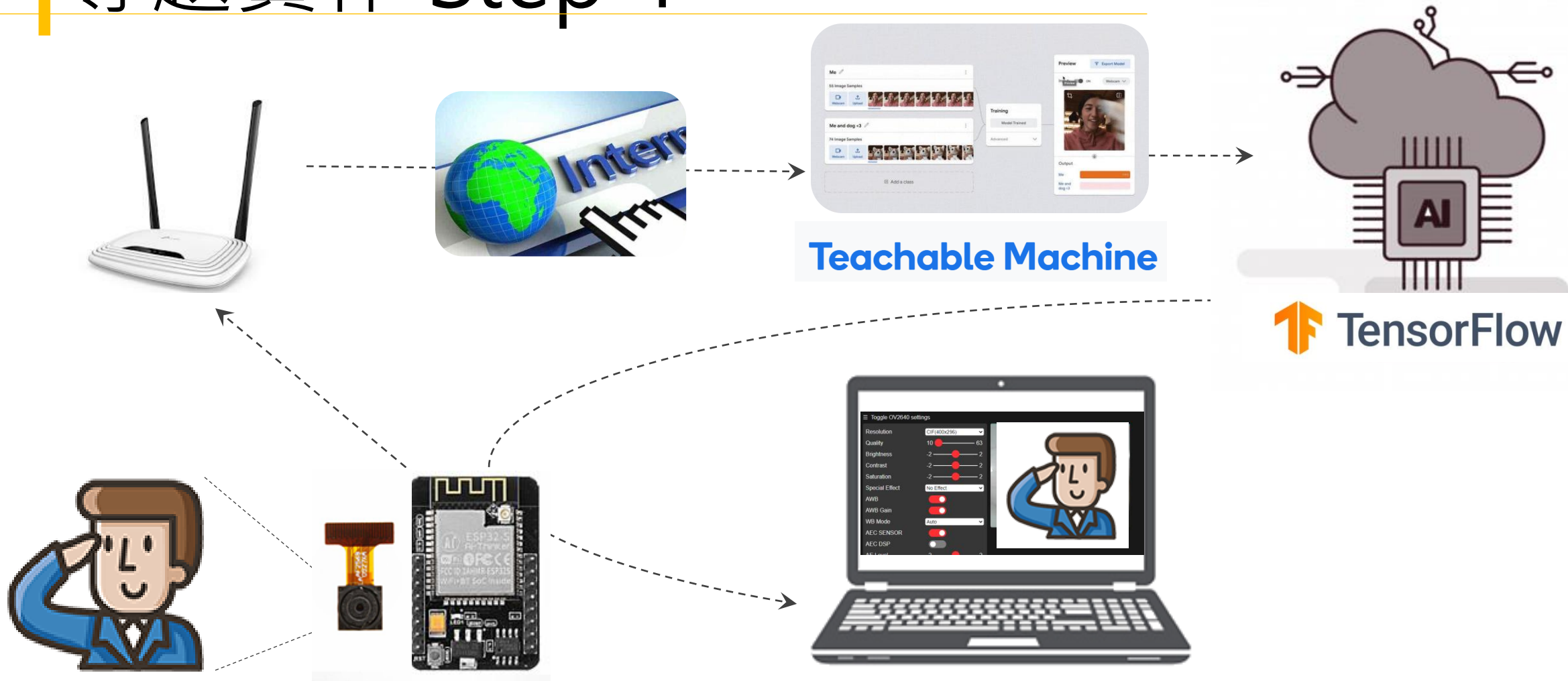
專題實作 Step 2



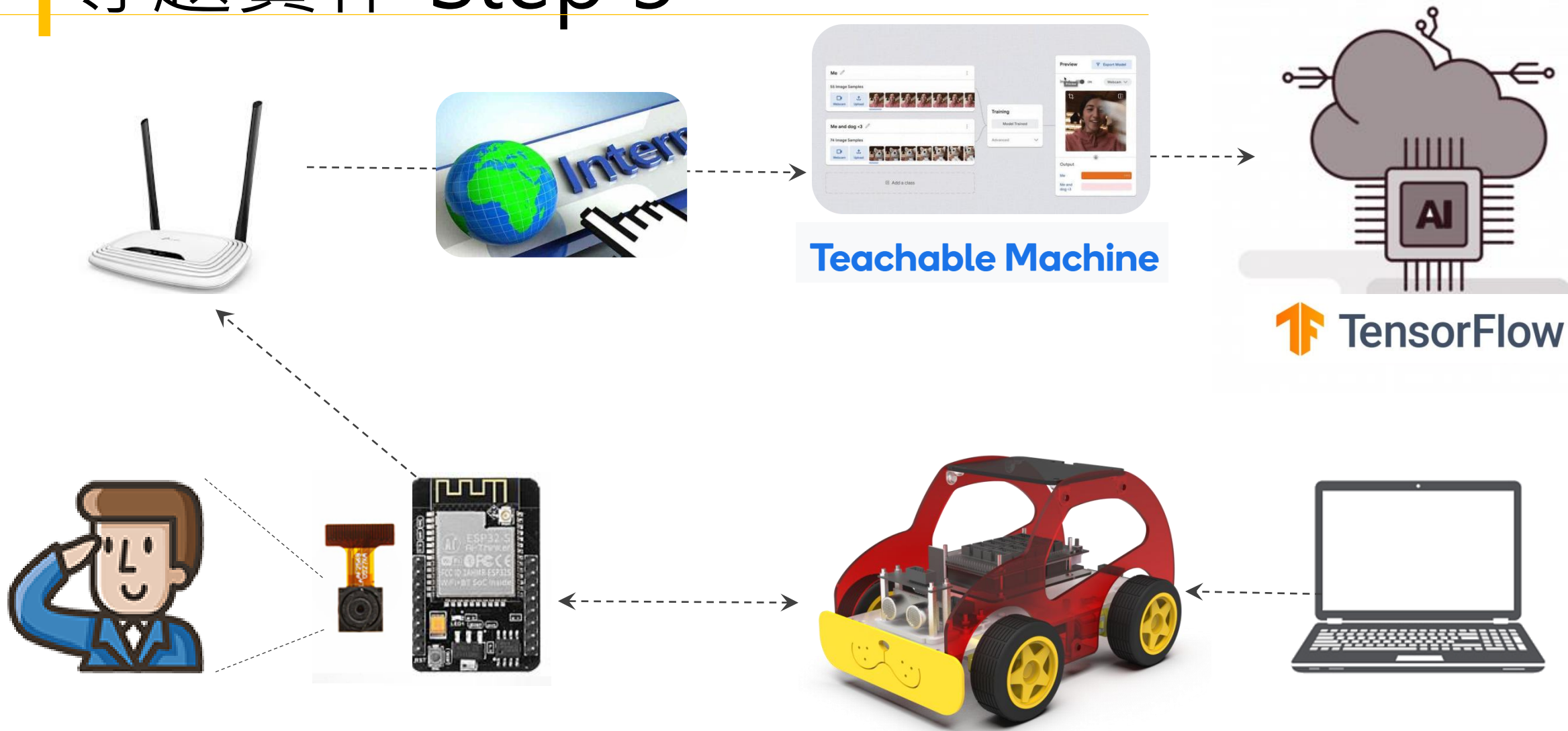
專題實作 Step 3



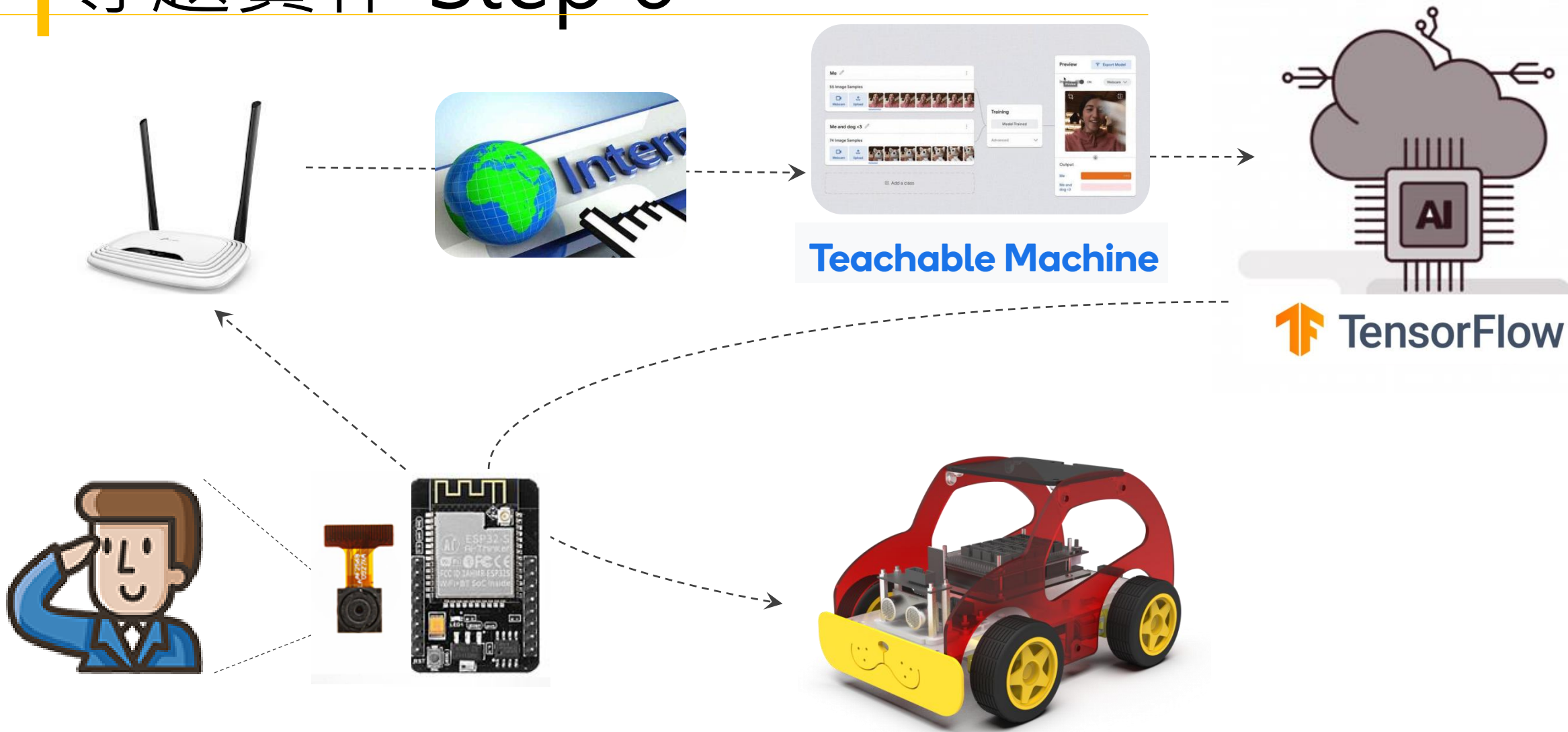
專題實作 Step 4



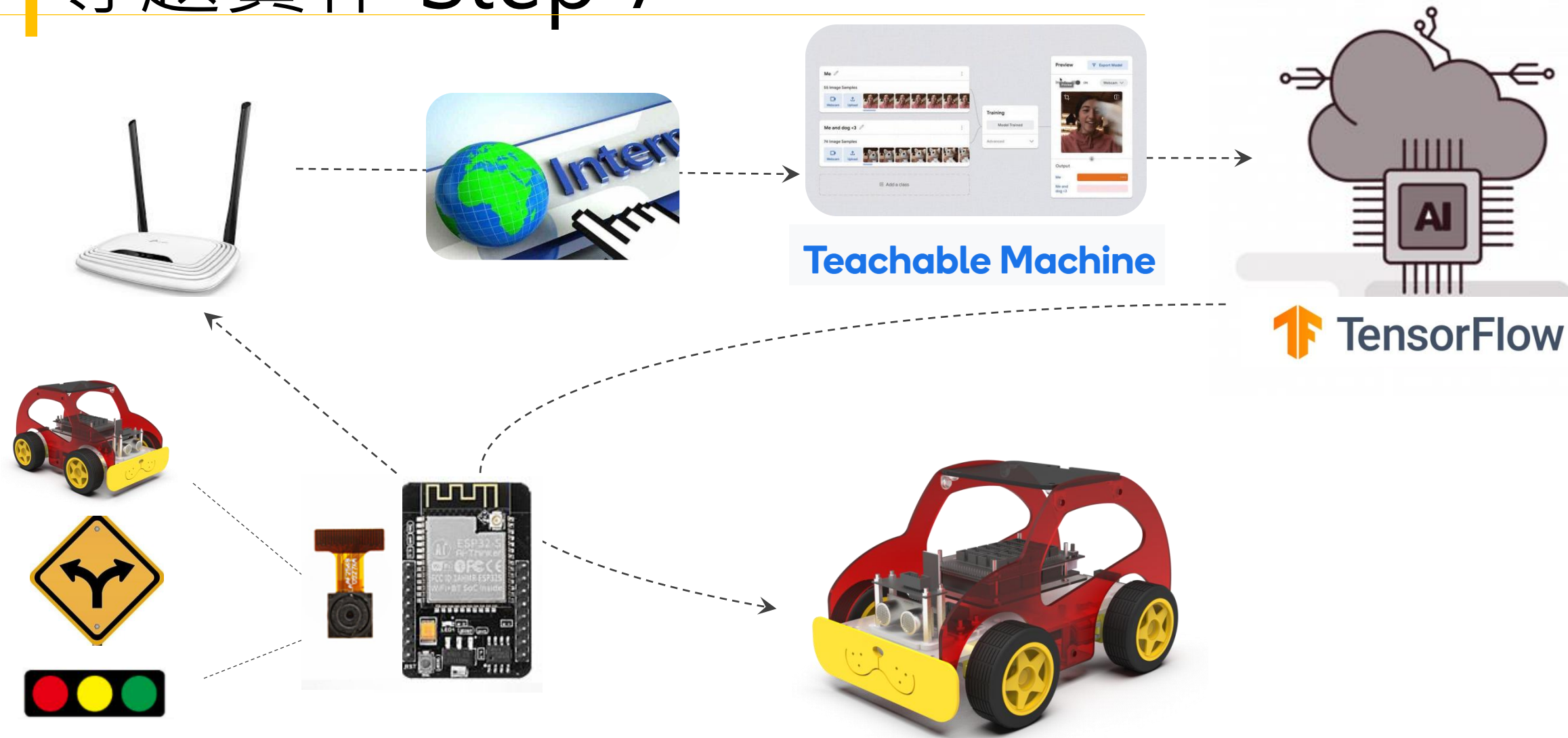
專題實作 Step 5



專題實作 Step 6



專題實作 Step 7

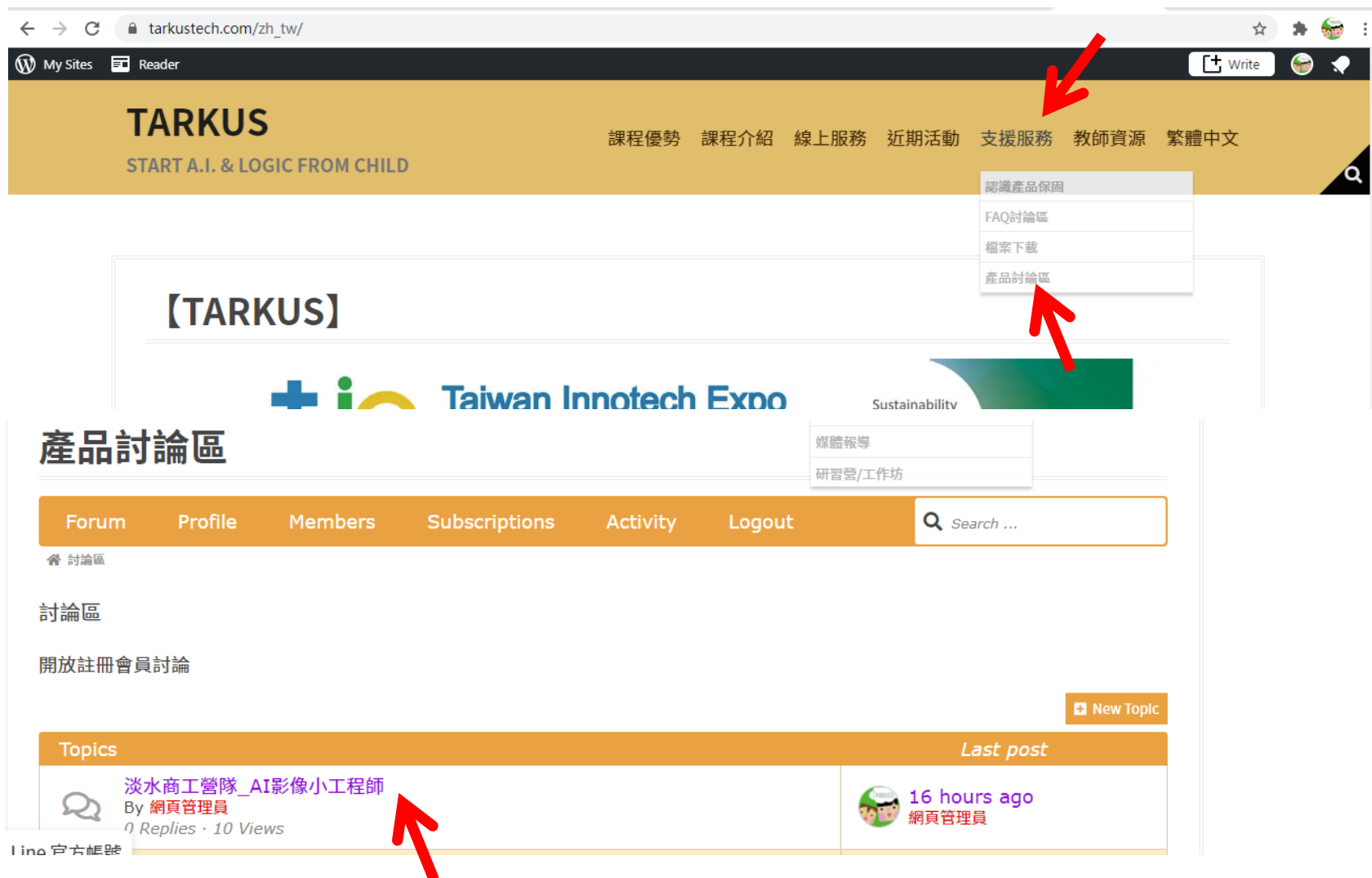


A photograph of three students, two girls and one boy, sitting at a desk and looking at a large computer monitor. The girl on the left is pointing at the screen with her right index finger. The boy in the middle is wearing glasses and looking at the screen. The girl on the right is also wearing glasses and looking at the screen. The monitor displays a blue interface with text and a bar chart. A yellow circle is overlaid on the image, containing the text "AI影像辨識小工程師" and "課程投影片".

AI影像辨識小工程師 課程投影片

資料下載

https://tarkustech.com/zh_tw/



The screenshot shows the TARKUS website interface. The top navigation bar includes links for 課程優勢, 課程介紹, 線上服務, 近期活動, 支援服務, 教師資源, and 繁體中文. A dropdown menu is open under 支援服務, showing options like 認識產品保固, FAQ討論區, 檔案下載, and 產品討論區. The main content area features a banner for Taiwan Innotech Expo and a section titled 產品討論區. Below this is a forum navigation bar with links for Forum, Profile, Members, Subscriptions, Activity, and Logout. The forum post list shows a topic titled 淡水商工營隊_AI影像小工程師 by 網頁管理員, with 10 replies and 10 views. A red arrow points to this post.

TARKUS
START A.I. & LOGIC FROM CHILD

課程優勢 課程介紹 線上服務 近期活動 支援服務 教師資源 繁體中文

認識產品保固
FAQ討論區
檔案下載
產品討論區

【TARKUS】

Taiwan Innotech Expo

產品討論區

Forum Profile Members Subscriptions Activity Logout Search ...

討論區

開放註冊會員討論

New Topic

Topics	Last post
 淡水商工營隊_AI影像小工程師 By 網頁管理員 10 Replies · 10 Views	 16 hours ago 網頁管理員

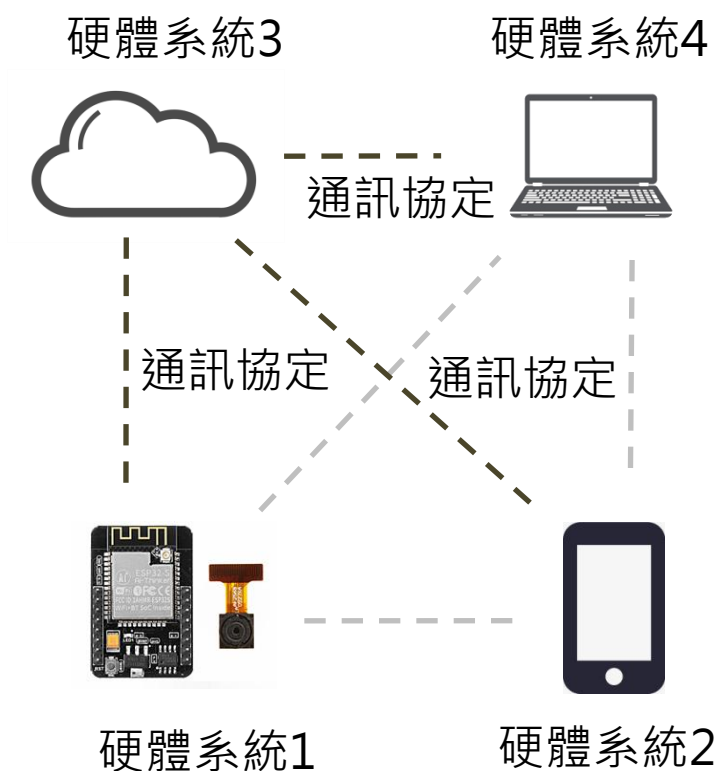
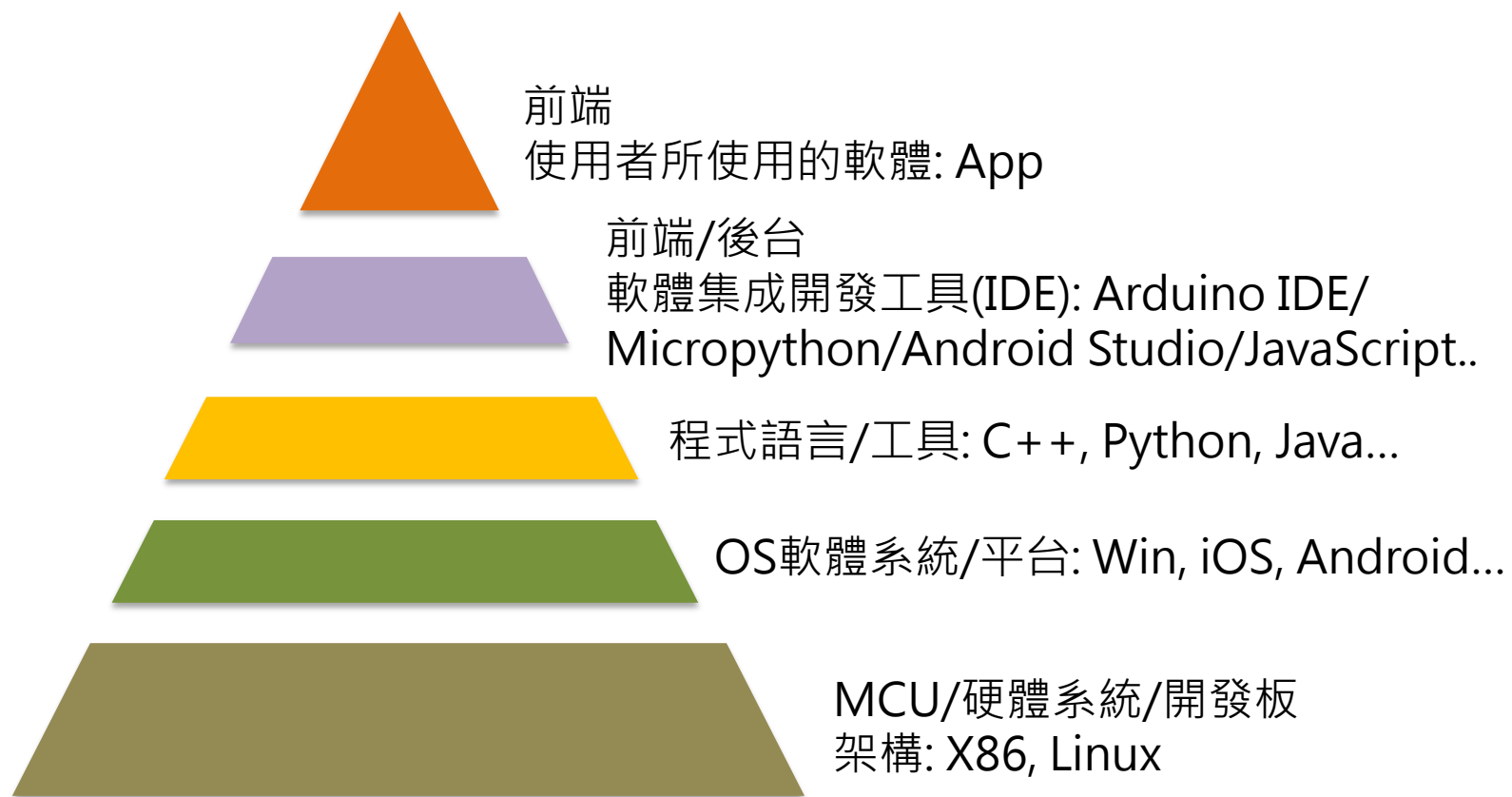
1 line 官方帳號

學習地圖

1. 學習使用Arduino控制ESP32, 連接各種感測器裝置及運用
2. 學習使用 Google Teachable Machine訓練模型，並將結果輸出至ESP32
3. Arduino實作練習：將訓練模型觀念應用至BeeCar小車, 做出影像辨識自動會車系統
4. 專題思考--未來的應用



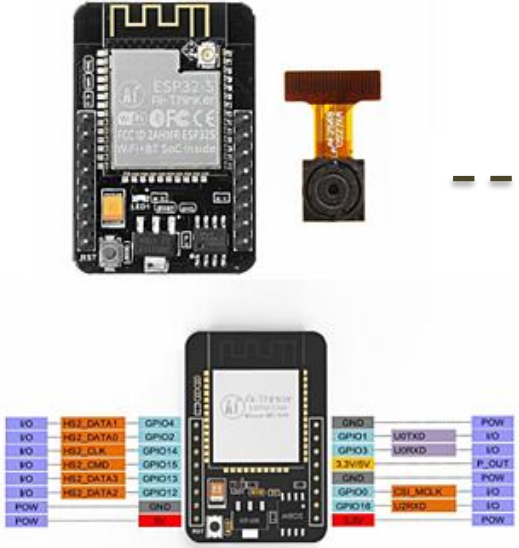
認識工具



A photograph of three students in a classroom or computer lab. A female student in the foreground is pointing at a large computer monitor displaying a blue-themed interface with text and a small chart. Two male students, both wearing glasses, are looking at the screen. Another laptop is visible on the desk, showing a colorful grid pattern. A large yellow circle is overlaid on the right side of the image, containing the text 'ESP32 影像實作'.

ESP32 影像實作

認識介面

內建裝置	
WIFI	
BLE 藍芽	
相機 OV2640	
TF 卡槽	
閃光燈	
特殊腳位	
GPIO0->燒錄用	
GPIO1->U0TXD	
GPIO3->U0RXD	
GPIO4->閃光燈共用	

工具1
ESP32開發板
Node MCU
機械語言 ←

通訊協定之間的轉換

TTL/UART

USB



電腦系統系統
MCU: Intel
OS: Windows
Tool: C++
IDE: Arduino IDE



工具2
USB to TTL轉換
CH340 (HW-597)模組

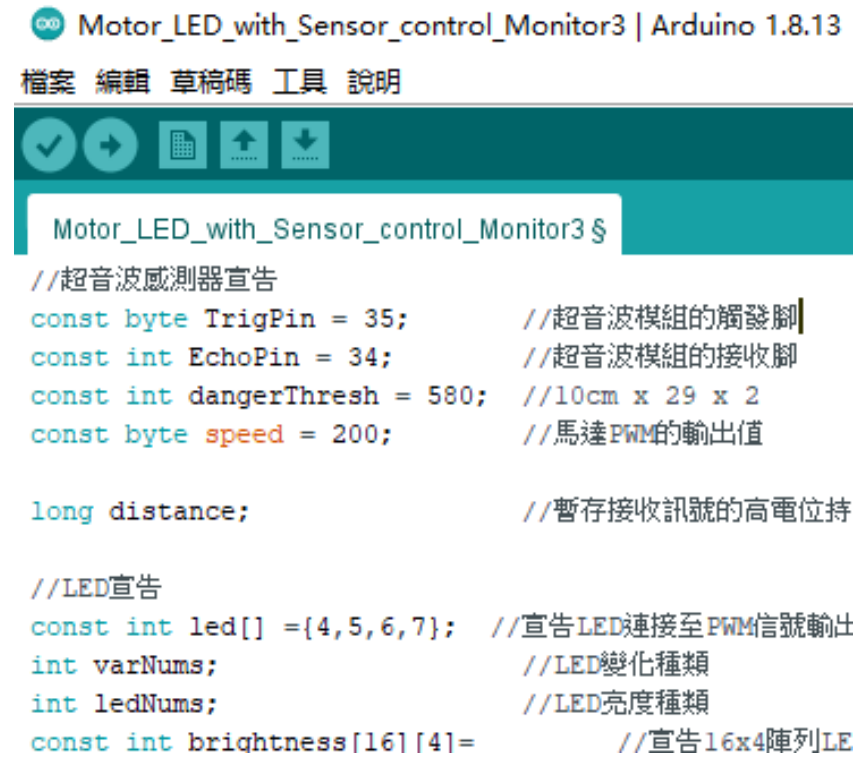


ARDUINO/C語言

Arduino IDE 學習工具

Arduino IDE相容於C和C++的語言

1. 因為源於Processing語言，這套程式也可以說是Java語言的基礎
2. 使用簡單的語法框架之下，讓零基礎的程式入門者可以輕鬆的進程式編寫
3. 同時透過Arduino相容的硬體套件，可以透過實際的反饋讓程式語言以"所見即所得"(What You See Is What You Get，縮寫WYSIWYG)的方式讓學程式這件事情更有具體的概念



Motor_LED_with_Sensor_control_Monitor3 | Arduino 1.8.13

檔案 編輯 草稿碼 工具 說明

Motor_LED_with_Sensor_control_Monitor3 \$

```
//超音波感測器宣告
const byte TrigPin = 35;           //超音波模組的觸發腳
const int EchoPin = 34;            //超音波模組的接收腳
const int dangerThresh = 580;      //10cm x 29 x 2
const byte speed = 200;            //馬達PWM的輸出值

long distance;                     //暫存接收訊號的高電位持

//LED宣告
const int led[] = {4,5,6,7};      //宣告LED連接至PWM信號輸出
int varNums;                       //LED變化種類
int ledNums;                       //LED亮度種類
const int brightness[16][4] =     //宣告16x4陣列LE
```

Arduino 下載方式

Google search results for "arduino ide".

約有 14,900,000 項結果 (搜尋時間: 0.47 秒) 在新分頁中開啟結果

www.arduino.cc › software ▾ 翻譯這個網頁

Software | Arduino

Arduino IDE 1.8.13. 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 ...

Arduino IDE · Donate · Release Notes

其他人也搜尋了以下項目

- arduino ide介紹 <https://arduino.cc>
- arduino打不開 [Arduino x64](#)
- Arduino code [Arduino online upload](#)

Arduino IDE 軟體

譯自英文 - Arduino集成開發
數編寫的跨平台應用程序。

Arduino.cc website showing the "Support the Arduino IDE" section.

Since its first release in March 2015, the Arduino IDE has been downloaded **47,503,327** times — impressive! Help its development with a donation.

Donation options: \$3, \$5, \$10, \$25, \$50, Other

JUST DOWNLOAD (circled) CONTRIBUTE & DOWNLOAD

Help

Arduino.cc website showing the "Downloads" section.

Arduino IDE 1.8.13

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.

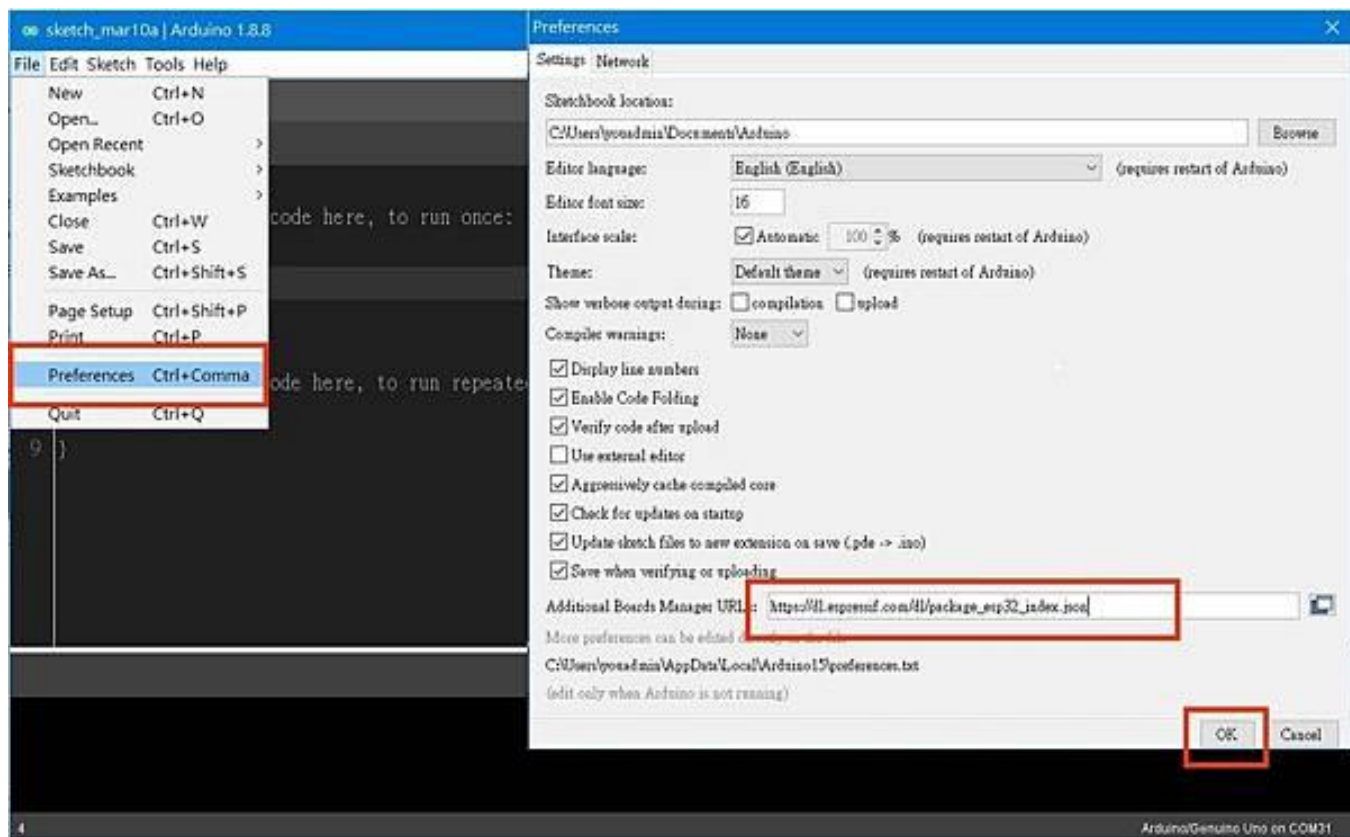
DOWNLOAD OPTIONS

- 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

Windows Device Manager showing the "Ports (COM & LPT)" section.

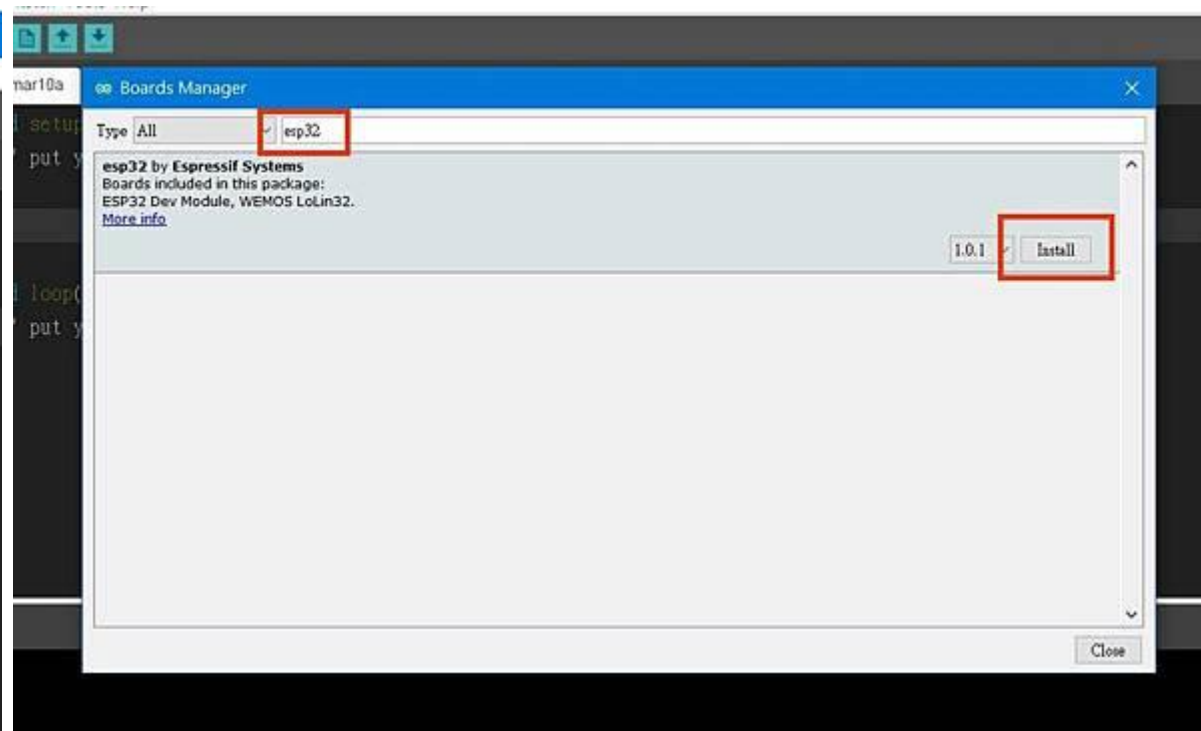
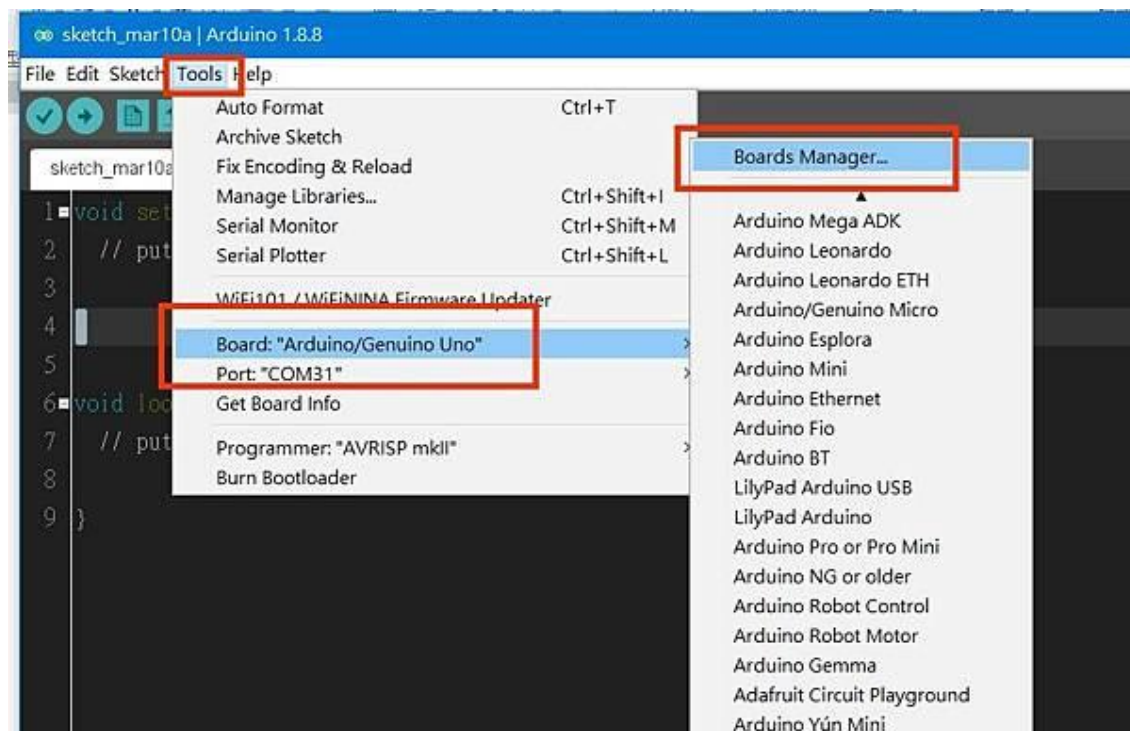
USB-SERIAL CH340 (COM5) (circled)

認識介面

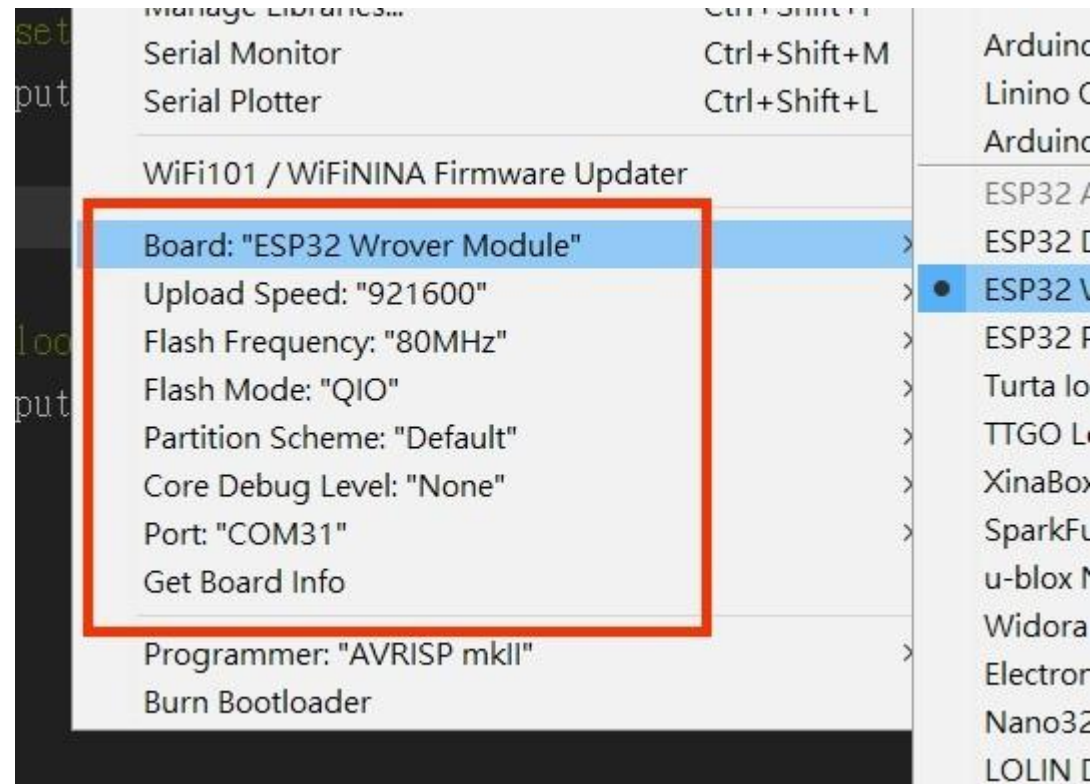
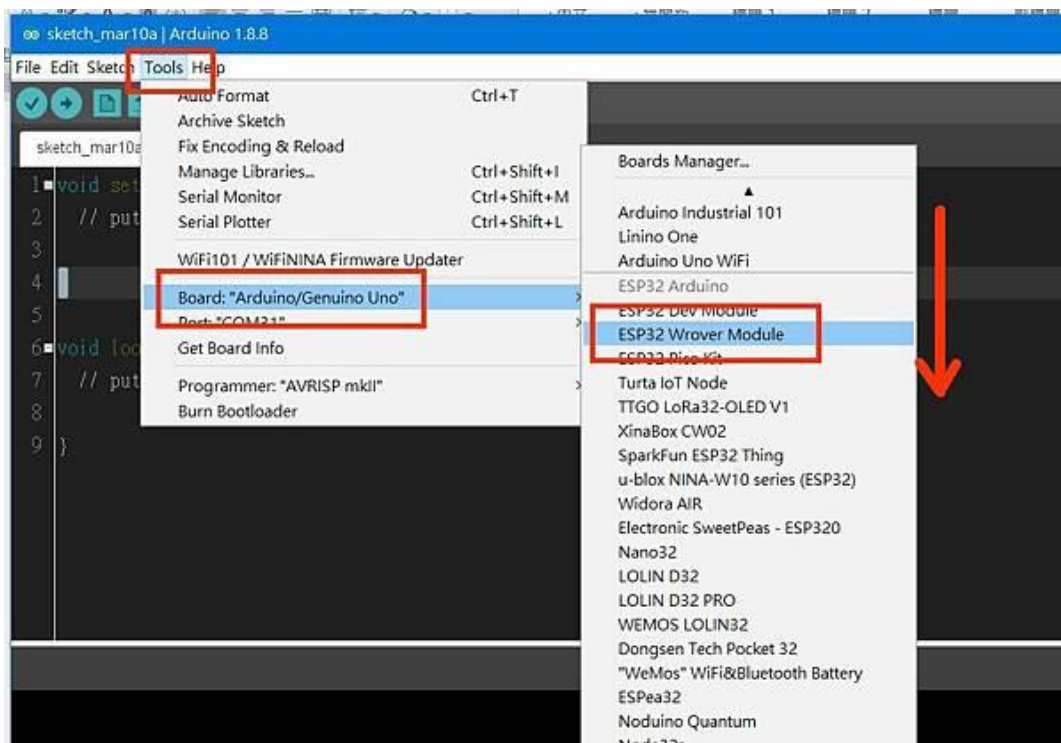


https://dl.espressif.com/dl/package_esp32_index.json

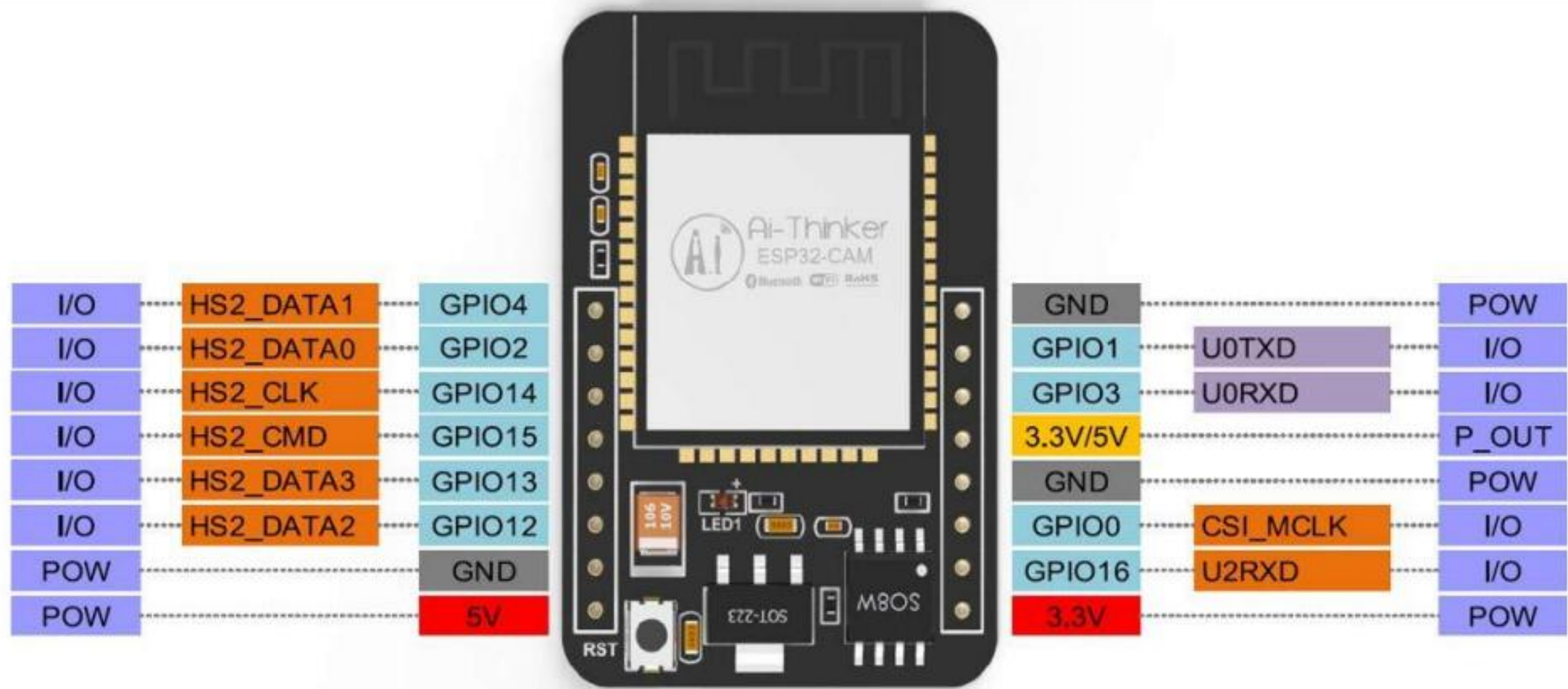
認識介面



認識介面



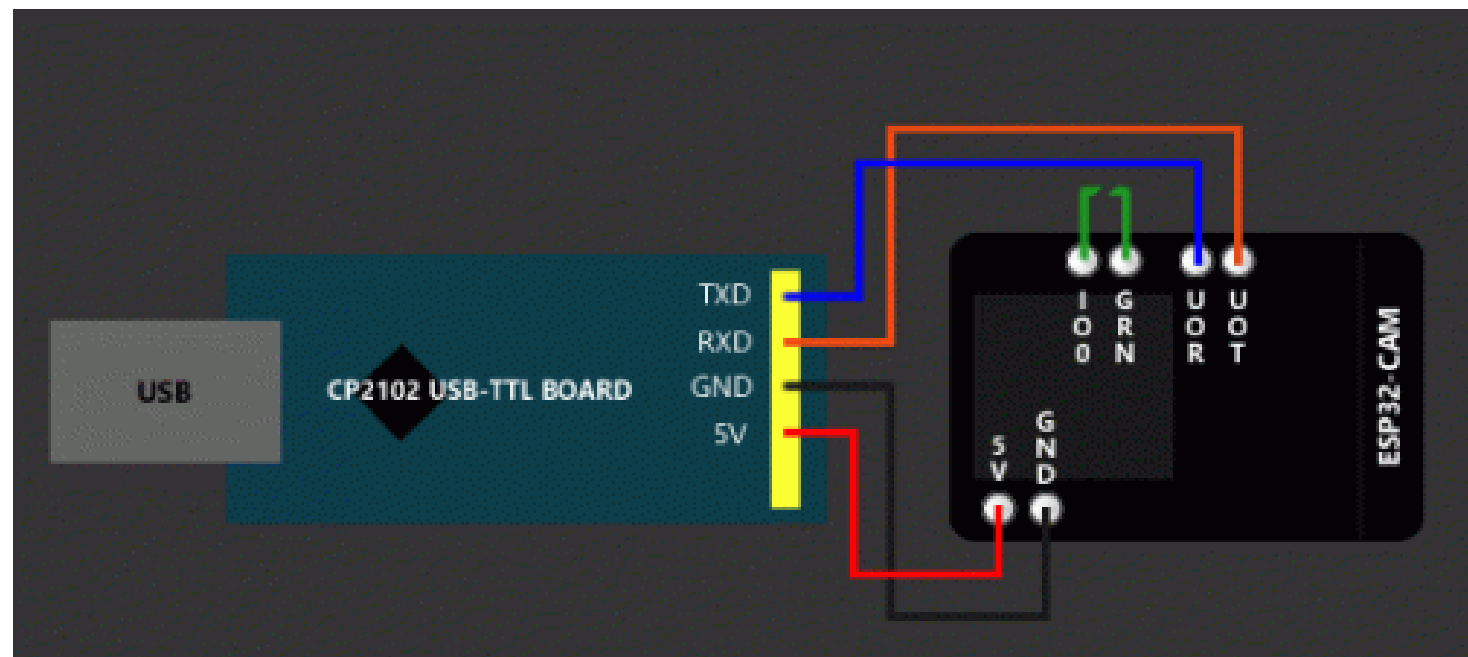
認識介面



工具1
ESP32開發板

認識介面

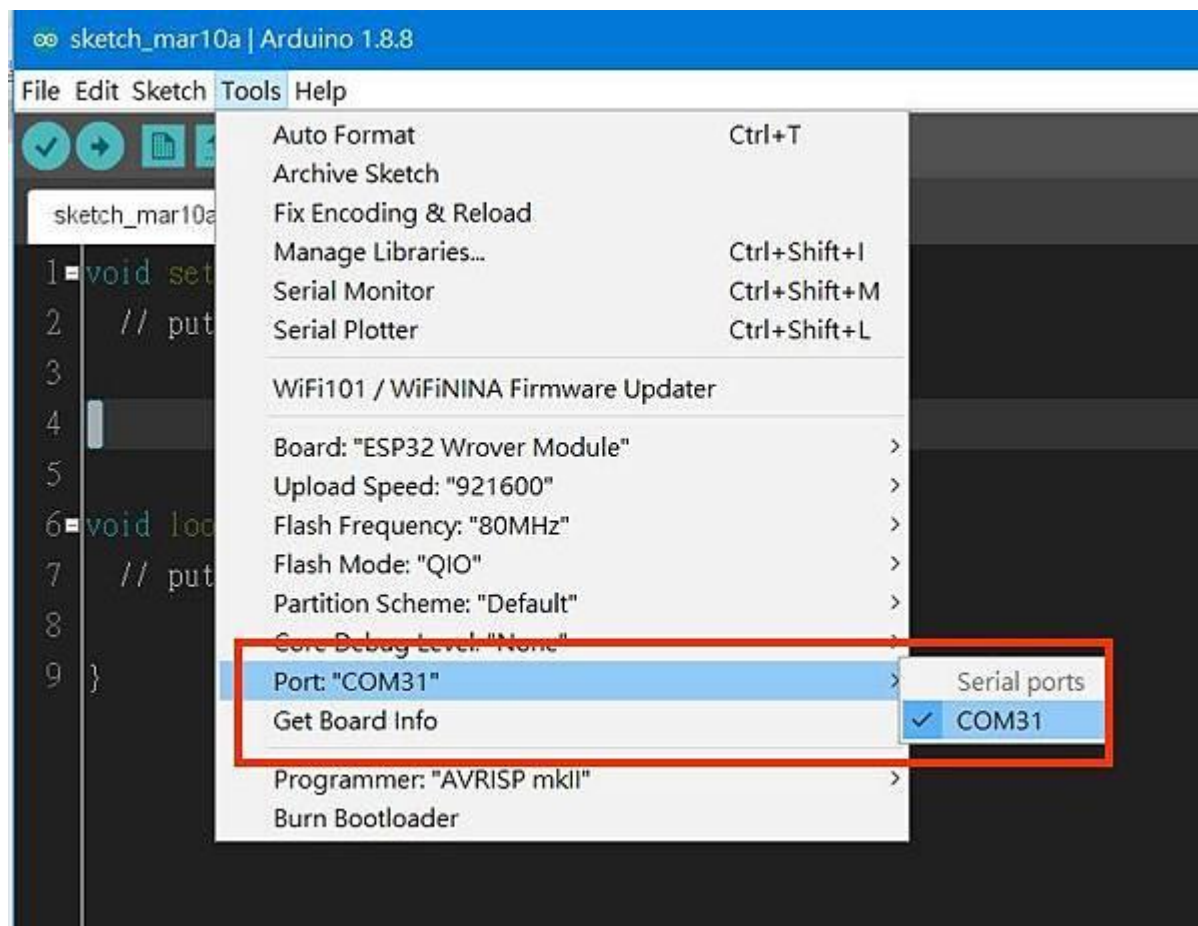
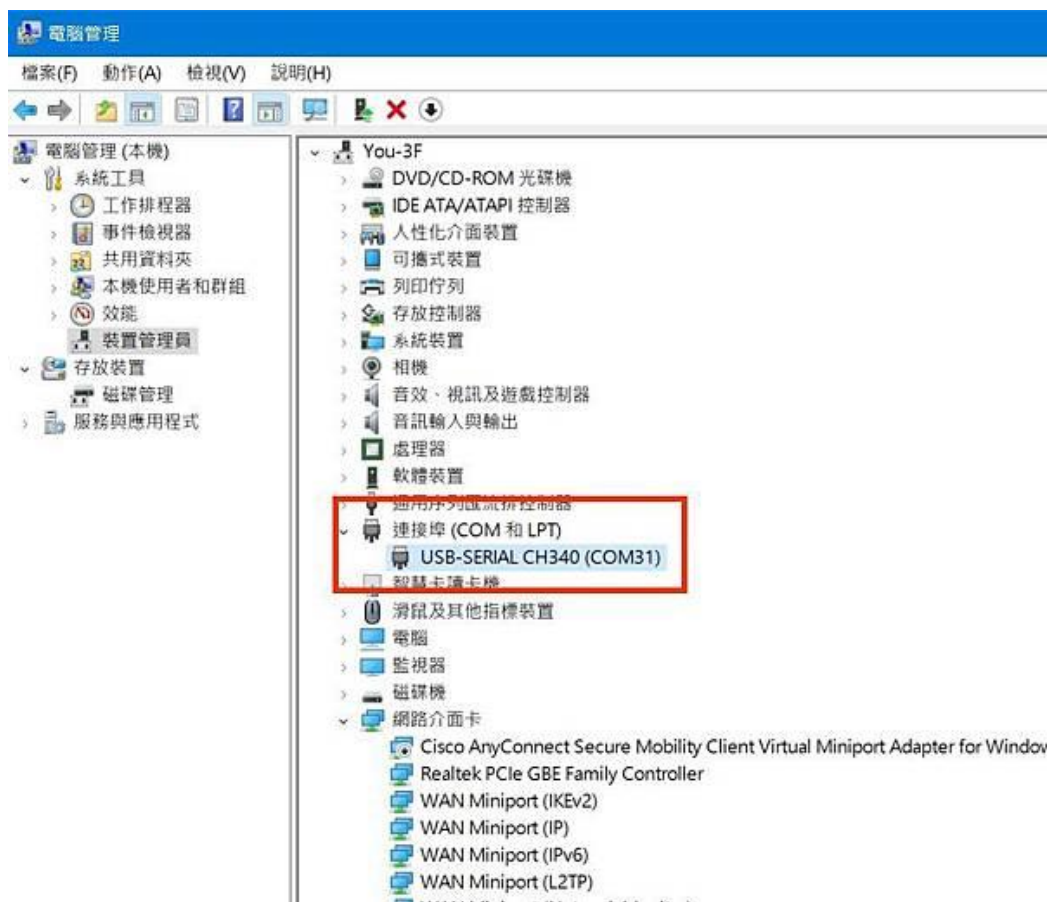
ESP32-CAM		USBTTL
U0RXD	—	TXD
U0TXD	—	RXD
5V	—	5V
GND	—	GND



工具2
USB to TTL轉換
CH340 (HW-597)模組

工具1
ESP32開發板

認識介面



認識介面



```
// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_ESP_EYE
// #define CAMERA_MODEL_M5STACK_PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE
// #define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
```



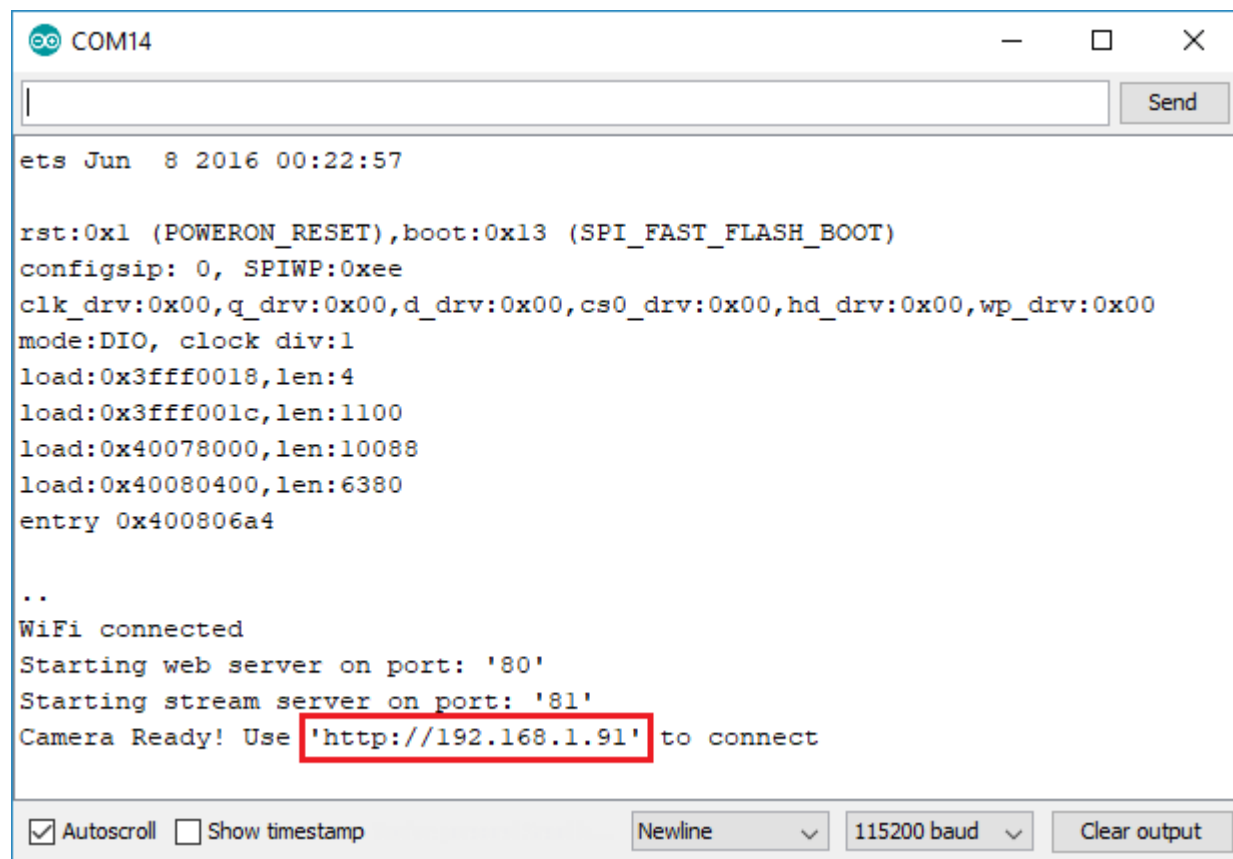
認識介面



```
esptool.py v2.3.1
Connecting.....
Chip is ESP32D0WD06 (revision 1)
<
23
```

1. 看到Connecting.....
 2. 將GPIO0與GND短路
 3. 按下Reset鍵
 4. 開始燒錄
- (步驟2~3可以在按下燒錄前先做)

認識介面



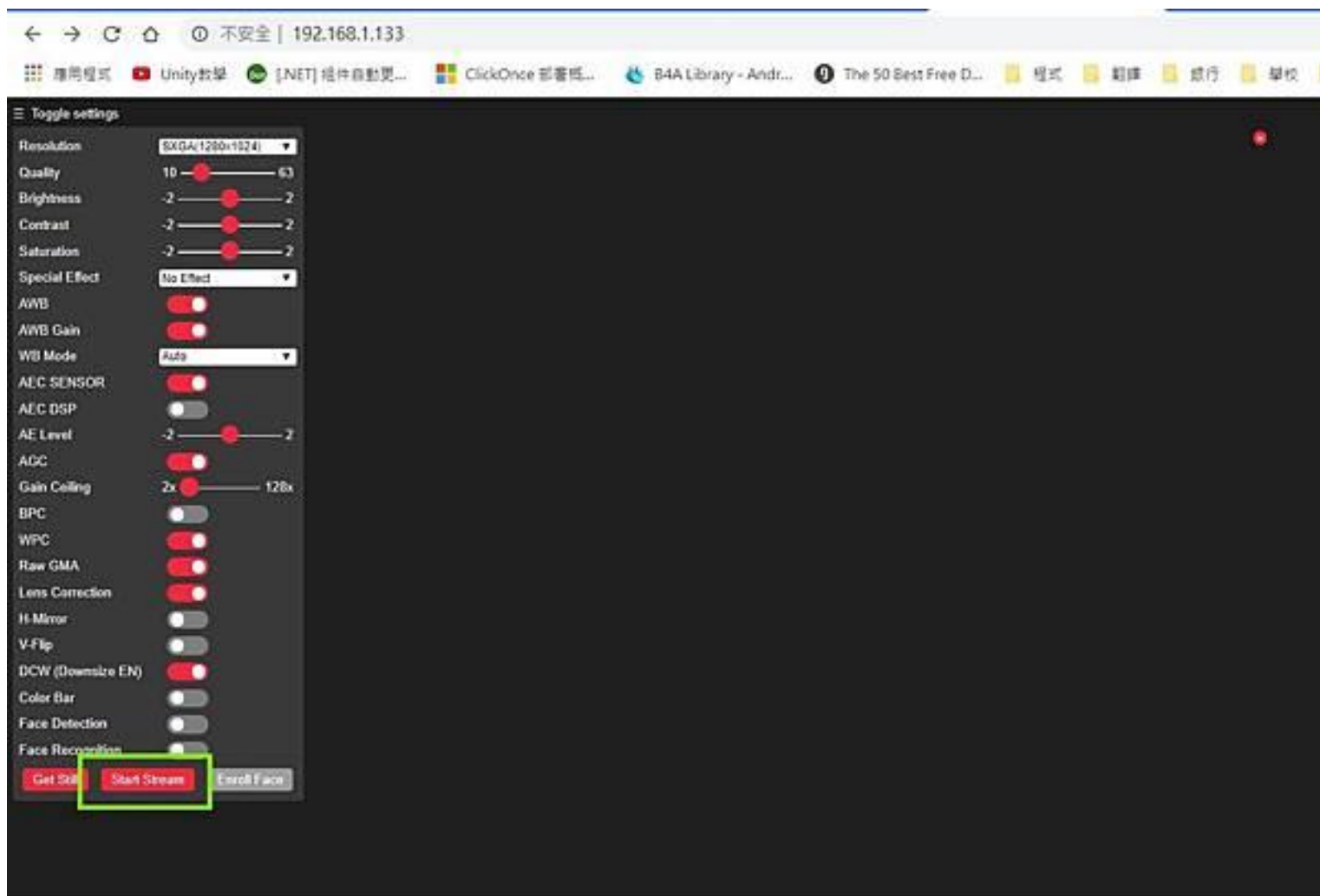
```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 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:1100
load:0x40078000,len:10088
load:0x40080400,len:6380
entry 0x400806a4

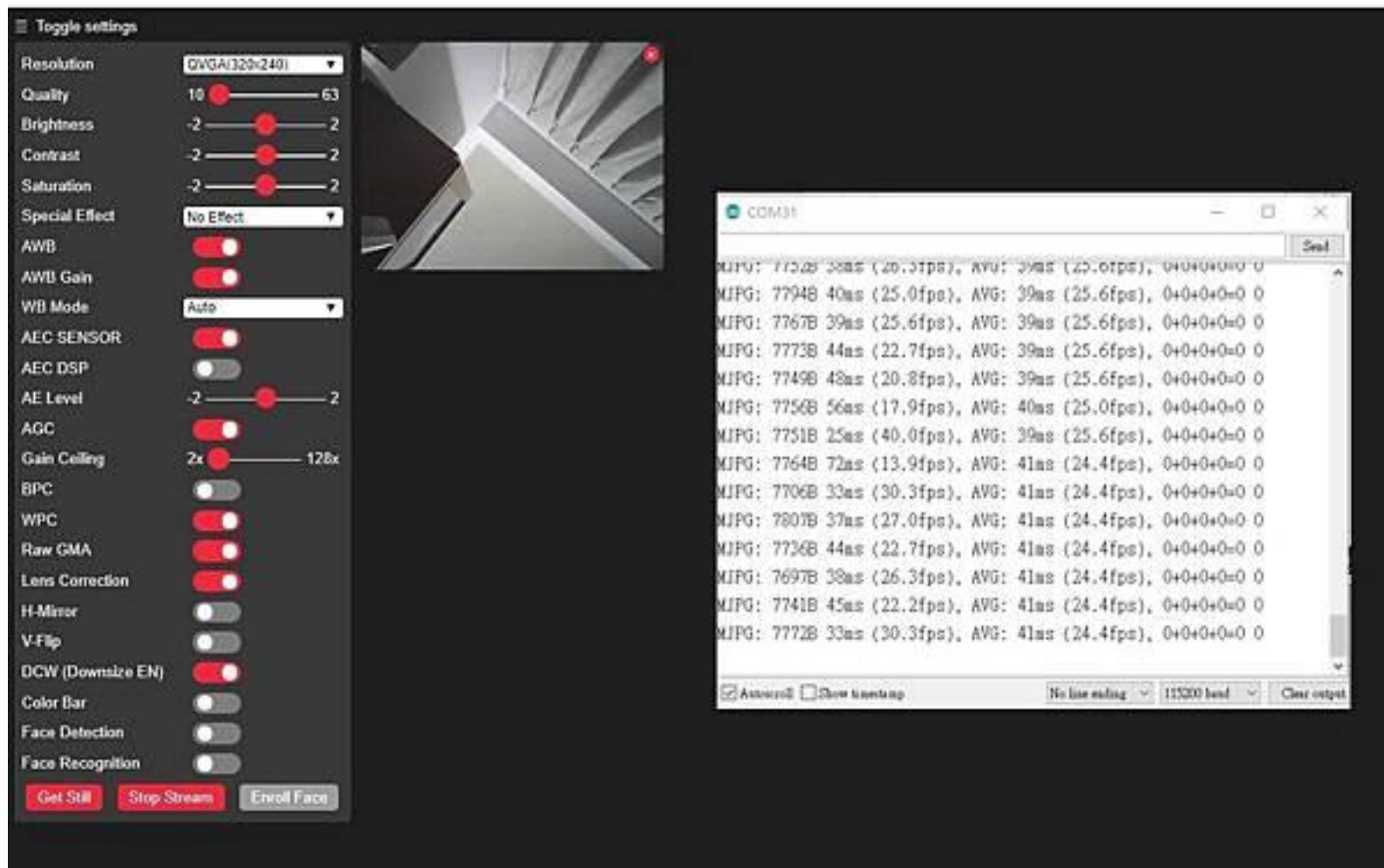
..
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.91' to connect
```

The screenshot shows a serial terminal window with a title bar 'COM14'. At the top, there is a text input field and a 'Send' button. The main area displays the boot sequence of an ESP8266, including reset information, configuration, and memory loading. It confirms WiFi connectivity and the starting of web and stream servers on ports 80 and 81. The final line states 'Camera Ready! Use 'http://192.168.1.91' to connect', with the IP address highlighted by a red rectangle. The bottom of the window features a control bar with checkboxes for 'Autoscroll' (checked) and 'Show timestamp' (unchecked), a 'Newline' dropdown menu, a baud rate dropdown set to '115200 baud', and a 'Clear output' button.

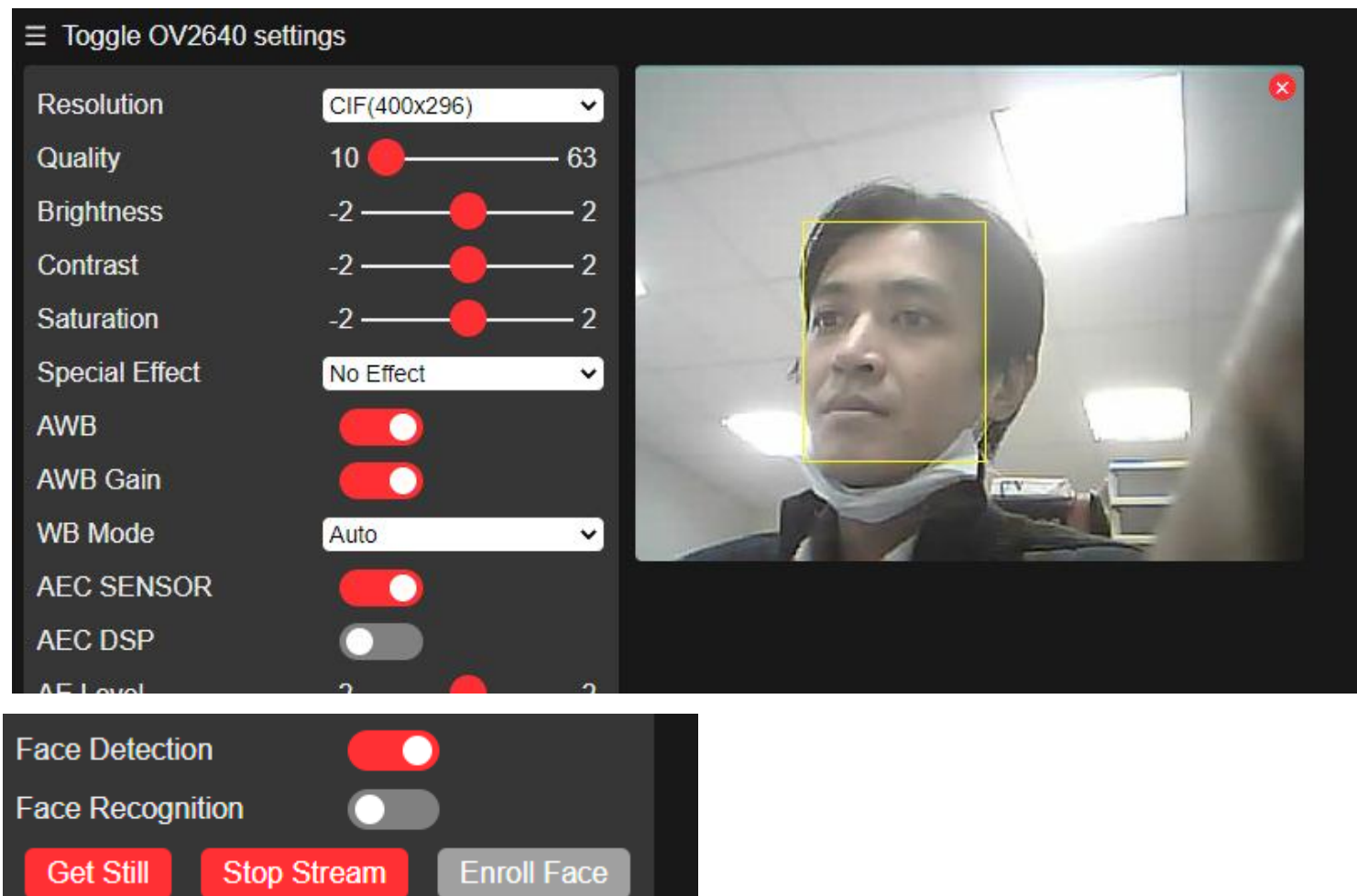
認識介面



認識介面



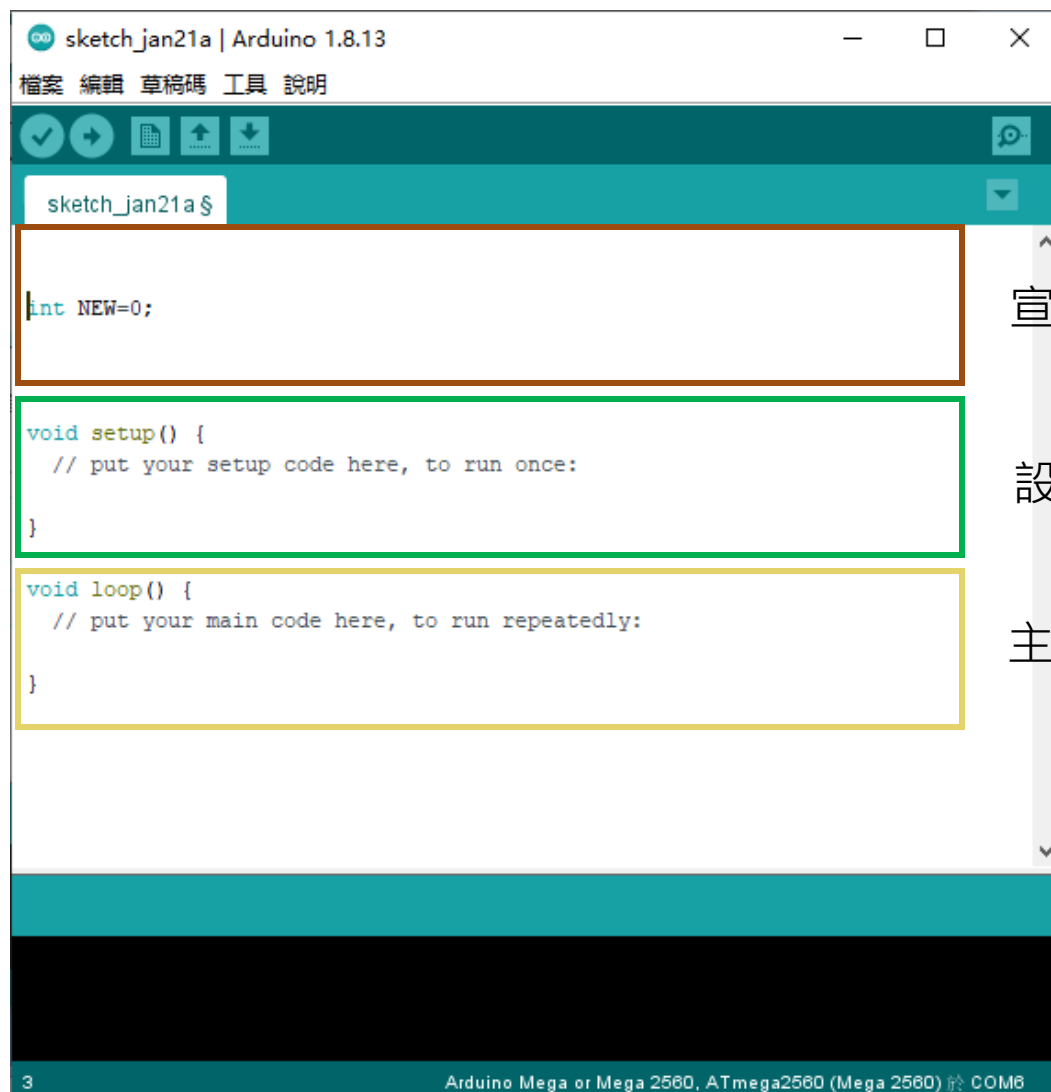
認識介面



A photograph of three students in a classroom or lab setting. A female student in the foreground is pointing at a large computer monitor displaying lines of code. Two male students, one wearing glasses, are looking on. A yellow circular graphic is overlaid on the right side of the image, containing the title text.

Arduino環境 基礎認識

認識介面



宣告: 變數/對應名稱/引入函式庫/設定函式

設定: 硬體資訊

主程式: 重複執行的動作

認識語言的架構

常用的資料型態：

資料型態	位元	範圍	C 格式化字串
char (字元)	8	-128~+127	%c
byte (位元組)	8	0~255	
int (整數)	16	-32,768~+32,767	%i , %d
long (長整數)	32	-2,147,483,648~+2,147,483,647	%li , %ld
float (單精實數)	32	-3.4028235E~+3.4028235E+38	%f , %e , %g
double (雙精實數)	32	-3.4028235E~+3.4028235E+38	%f , %e , %g

變數命名規則：

1. 第一個字元只可以是大小寫字母或底線 (A 至 Z、a 至 z 或 “_”)
2. 第二個或之後的字元只可以是大小寫字母、底線或數字 (A至Z、a至z、_或0至9)
3. 不可以是「保留字」(ReservedIdentifiers)，如：main、int、long、float、double等。

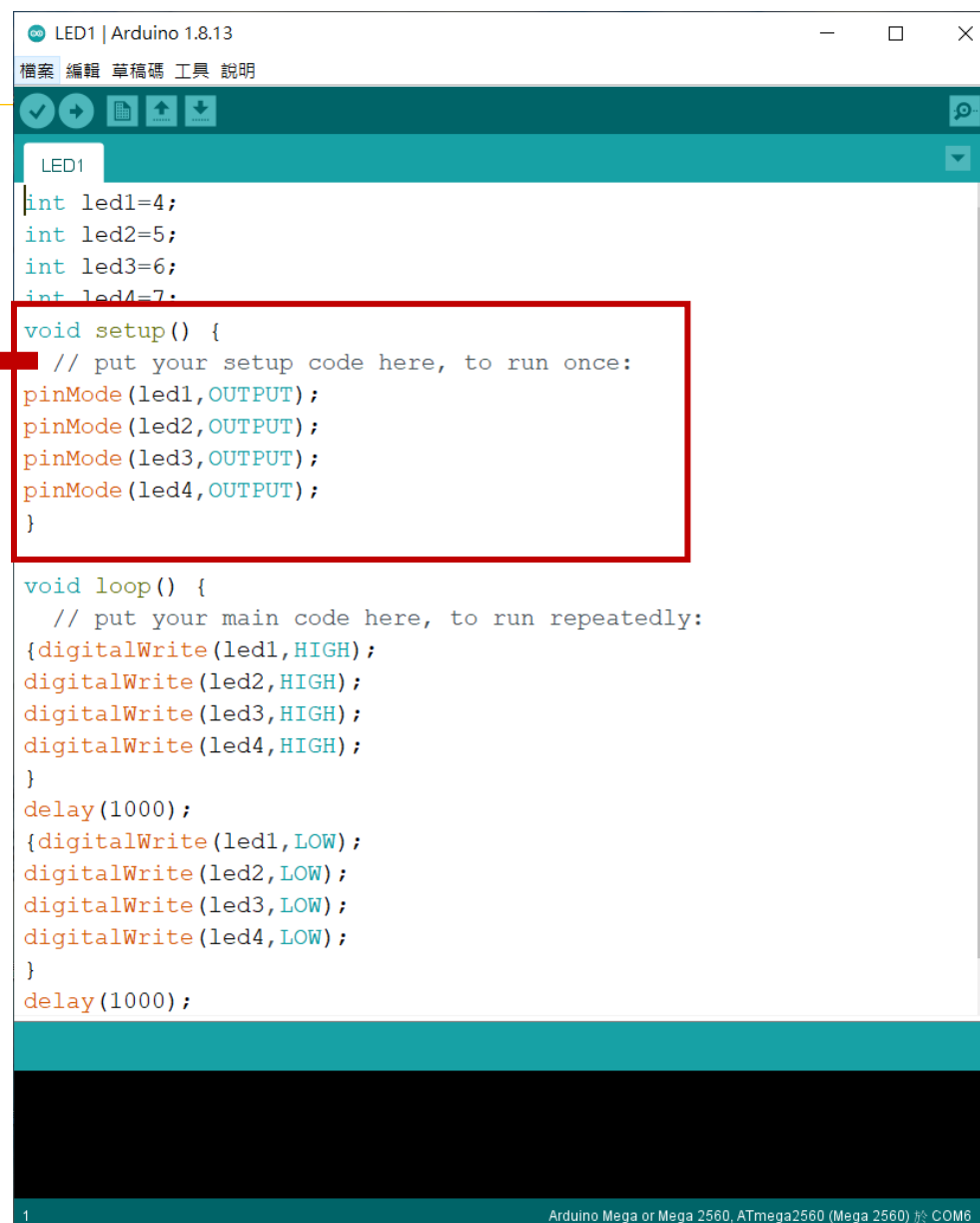
認識語言的架構

Arduino void setup :
(初始化設定)

宣告硬體模組PIN腳模式(對應全域宣告腳位)，輸出 OUTPUT、輸入 INPUT、序列埠監控視窗初始化 Serial.begin(9600); 等等。

```
main.c
1
2 #include <stdio.h>
3
4 int main()
5 {
6     printf("Hello World");
7
8     return 0;
9 }
10
```

※在C Compiler IDE 中沒有 void setup 這個區域。



```
LED1 | Arduino 1.8.13
檔案 編輯 草稿碼 工具 說明

LED1
int led1=4;
int led2=5;
int led3=6;
int led4=7;

void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    }
    delay(1000);
    {digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    }
    delay(1000);
}
```

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) 於 COM6

認識語言的架構

宣告全域變數：

整數變數 `i`，初始值為0。

整數常數 `k`，值為3。

字元變數 `ID`，初始值為 `A`。

實數變數 `val`，初始值為12.34。

宣告區域變數：

變數整數 `a`，只在void

`loop()`內才有效。

變數整數 `c`，只在for迴圈內才有效。

sketch_sep11a | Arduino 1.8.13

檔案 編輯 草稿碼 工具 說明

sketch_sep11a \$

```
int i=0;
const int k=3;
char ID=A;
float val=12.34;
```

```
void setup() {
    // put your setup code here, to run once:
}
```

```
void loop() {
    // put your main code here, to run repeatedly:
```

```
int a;
for(int c=0;c<5;c++)
{
    //...
}
```

認識語言的架構

算術運算子：

運算子	動作	範例	說明
+	加法	$a+b$	a與b相加
-	減法	$a-b$	a與b相減
*	乘法	$a*b$	a與b相乘
/	除法	a/b	a除以b的商數
%	餘數	$a\%b$	a除以b的餘數
++	遞增	$a++$	a值加1，即 $a=a+1$
--	遞減	$a--$	a值減1，即 $a=a-1$
=	等於	$a=b$	a等於b的值

關係運算子：

運算子	動作	範例	說明
==	等於	$a==b$	a等於b? 是為true，否為false
!=	不等於	$a!=b$	a不等於b? 是為true，否為false
<	小於	$a<b$	a小於b? 是為true，否為false
>	大於	$a>b$	a大於b? 是為true，否為false
<=	小於等於	$a<=b$	a小於等於b? 是為true，否為false
>=	大於等於	$a>=b$	a大於等於b? 是為true，否為false

認識語言的架構

邏輯運算子：

運算子	動作	範例	說明
&&	AND	a&&b	a與b執行邏輯AND運算 兩數皆為真時 結果才為真
	OR	a b	a與b執行邏輯OR運算 有任一數為真時 結果為真
!	NOT	!a	a與b執行邏輯NOT運算 原數為真，結果為假 原數為假，結果為真

在邏輯運算中，非0的數即為真(true)，為0的數即為假(false)

位元運算子：

運算子	動作	範例	說明
&	AND	a&b	a與b兩數每一同位元 執行AND運算
	OR	a b	a與b兩數每一同位元 執行OR運算
^	XOR	a^b	a與b兩數每一同位元 執行XOR運算
~	補數	~a	將a中的每一位元相反
<<	左移	a<<2	將a內含的值 左移2個位元
>>	右移	a>>2	將a內含的值 右移2個位元

認識語言的架構

複合運算子：

運算子	動作	範例	說明
<code>+=</code>	加法	<code>a+=b</code>	同 <code>a=a+b</code> 運算
<code>-=</code>	減法	<code>a-=b</code>	同 <code>a=a-b</code> 運算
<code>*=</code>	乘法	<code>a*=b</code>	同 <code>a=a*b</code> 運算
<code>/=</code>	除法	<code>a/=b</code>	同 <code>a=a/b</code> 運算
<code>%=</code>	餘數	<code>a%=b</code>	同 <code>a=a%b</code> 運算
<code>&=</code>	遞增	<code>a&=b</code>	同 <code>a=a&b</code> 運算
<code> =</code>	遞減	<code>a =b</code>	同 <code>a=a b</code> 運算
<code>^=</code>	等於	<code>a^=b</code>	同 <code>a=a^b</code> 運算

C 運算子：

遞增：`++a`

先自增在參與運算

遞減：`--a`

先自減在參與運算

A photograph of three students in a classroom setting. A female student in the foreground is pointing at a large computer monitor displaying a blue-themed interface with text and a bar chart. Two male students, both wearing glasses, are looking at the screen. A yellow circular graphic is overlaid on the right side of the image, containing the title text.

距離感測器 基礎認識

Arduino 語法

距離感測器宣告 -

```
const int TrigPin = 35;  
const int EchoPin = 34;  
int duration = 0;  
int S_DISTANCE;  
void distance()  
{
```

```
    digitalWrite(TrigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(TrigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(TrigPin, LOW);  
    duration = pulseIn(EchoPin, HIGH);  
    S_DISTANCE = duration * 0.034 / 2;  
};
```

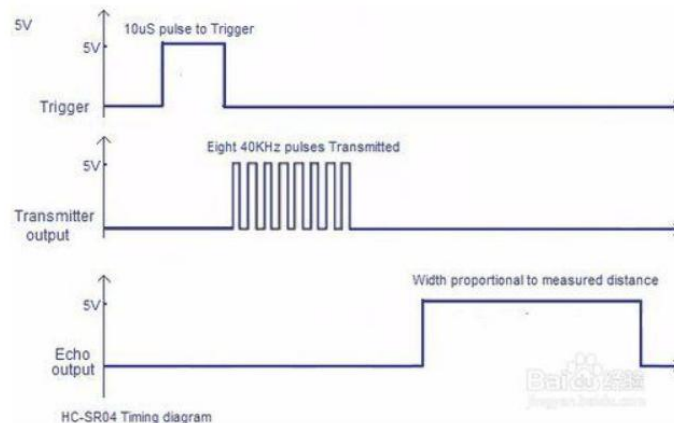
計算距離：時間乘上聲音在空氣中傳送的速度再除以 2。



用低高低電位發出一個 10 微秒的超聲波脈衝。

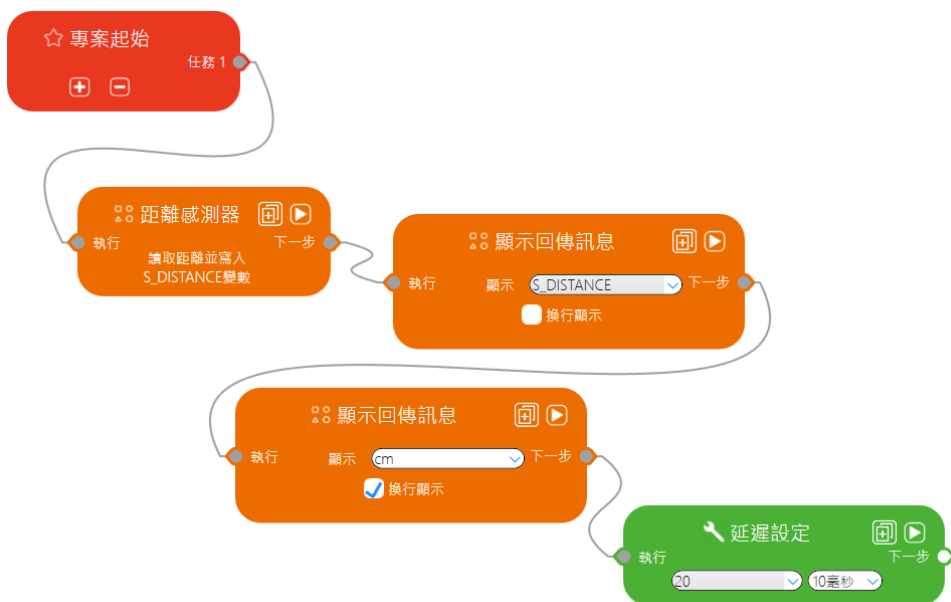
有信號返回，高電位持續時間則為從發射到接收的總時間(微秒)。

```
C Code Window  
int duration = 0;  
digitalWrite(TRIG, LOW);  
delayMicroseconds(2);  
digitalWrite(TRIG, HIGH);  
delayMicroseconds(10);  
digitalWrite(TRIG, LOW);  
duration = pulseIn(ECHO, HIGH);  
S_DISTANCE = duration * 0.034 / 2;
```



Arduino 語法

讀取距離 –



Tarkus VP 畫面

```

void setup() {
    // put your setup code here, to run once:
    pinMode(TrigPin, OUTPUT);
    pinMode(EchoPin, INPUT);
    Serial.begin(9600);
}
  
```

設定腳位模式 TrigPin 為輸出，EchoPin 為輸入，初始化序列埠監控視窗，傳輸速率為 9600 鮑率。

```

void loop() {
    // put your main code here, to run repeatedly:
    distance();
    Serial.print(S_DISTANCE);
    Serial.print("cm");
    Serial.print("\n");
    delay(200);
}
  
```

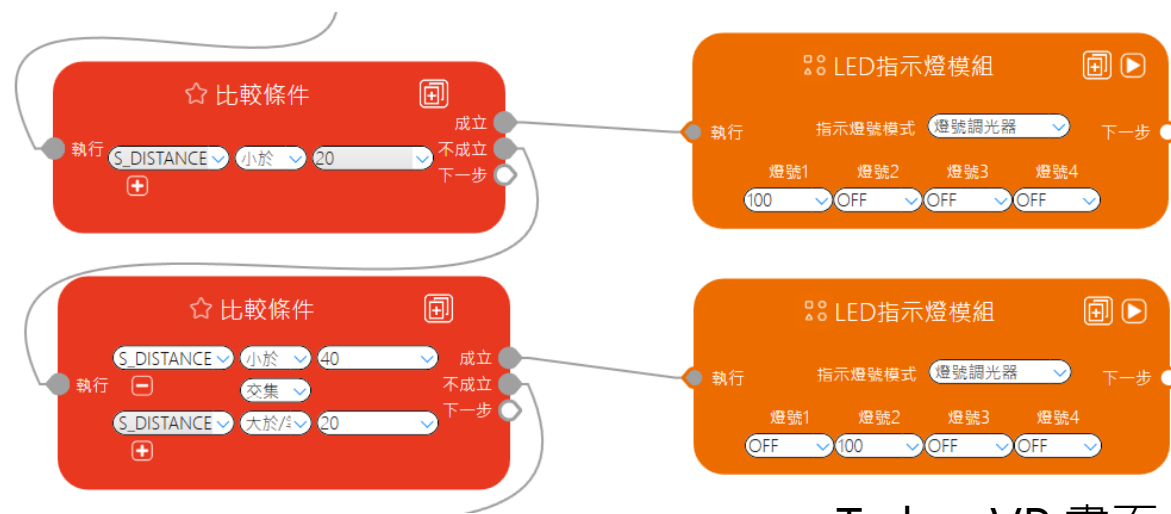
執行函式 distance()，顯示計算完後的距離 S_DISTANCE，顯示字元 cm，換行顯示，延遲 200 毫秒。

Arduino 語法

感測器控制燈光(if-else if 敘述) –

if-else if 會先判斷第一個條件式，如條件為真則執行 {} 內容，若條件式為假則執行第二個 else if 的條件式判斷，以此類推，如所有條件式皆不成立則執行 else 的內容。

```
if(S_DISTANCE < 20)
{
  analogWrite(LED[0],100);
  analogWrite(LED[1],0);
  analogWrite(LED[2],0);
  analogWrite(LED[3],0);
}
else if((S_DISTANCE >= 20)&&(S_DISTANCE < 40))
{
  analogWrite(LED[0],0);
  analogWrite(LED[1],100);
  analogWrite(LED[2],0);
  analogWrite(LED[3],0);
}
```



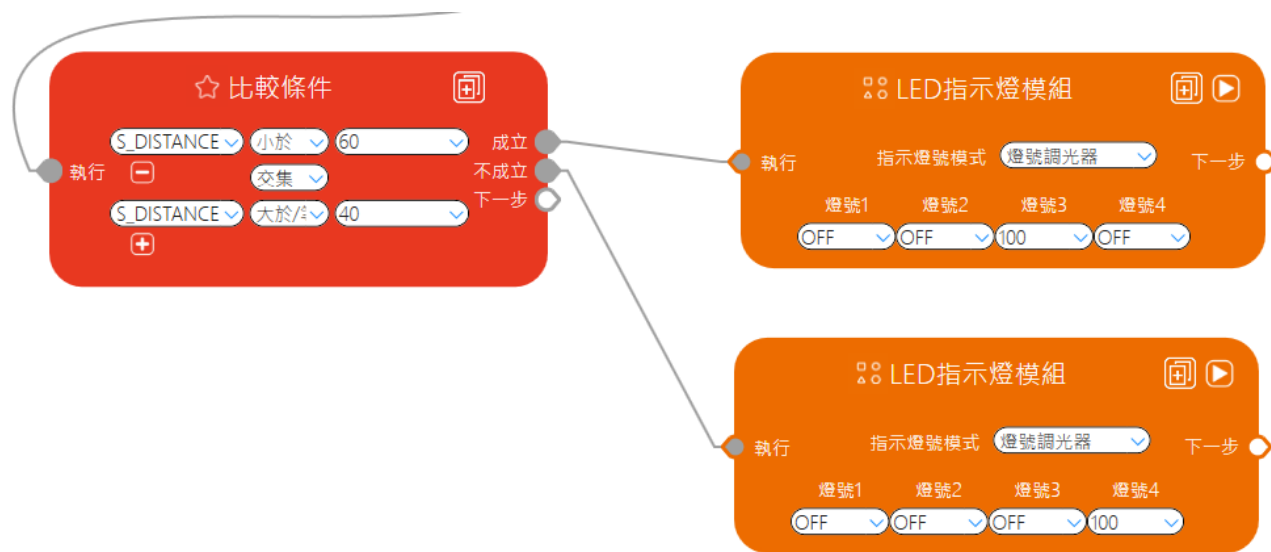
Tarkus VP 畫面

條件：距離小於20公分時亮1號燈，
介於20~40公分時亮2號燈

Arduino 語法

感測器控制燈光(if-else if 敘述) –

```
else if((S_DISTANCE >= 40) && (S_DISTANCE < 60))
{
    analogWrite(LED[0], 0);
    analogWrite(LED[1], 0);
    analogWrite(LED[2], 100);
    analogWrite(LED[3], 0);
}
else{
    analogWrite(LED[0], 0);
    analogWrite(LED[1], 0);
    analogWrite(LED[2], 0);
    analogWrite(LED[3], 100);
}
}
```



條件：距離介於40~60時亮3號燈，
以上條件都不成立時亮4號燈

Tarkus VP 畫面

Arduino 語法

感測器控制燈光(巢狀 if-else 敘述) –

以右圖程式碼為例：

條件1、2、3皆成立時執行敘述 1 ；

條件 1 、2 成立，3 不成立時執行敘述 2 ；

只有條件 1 成立時執行敘述 3 ；

所有條件不成立則執行敘述 4 。

條件：

距離小於 20 亮 0號燈，

距離介於 20 ~ 40 時亮 1 號燈，

距離介於 40 ~ 60 時亮 2 號燈，

距離大於 60 時亮 3 號燈。

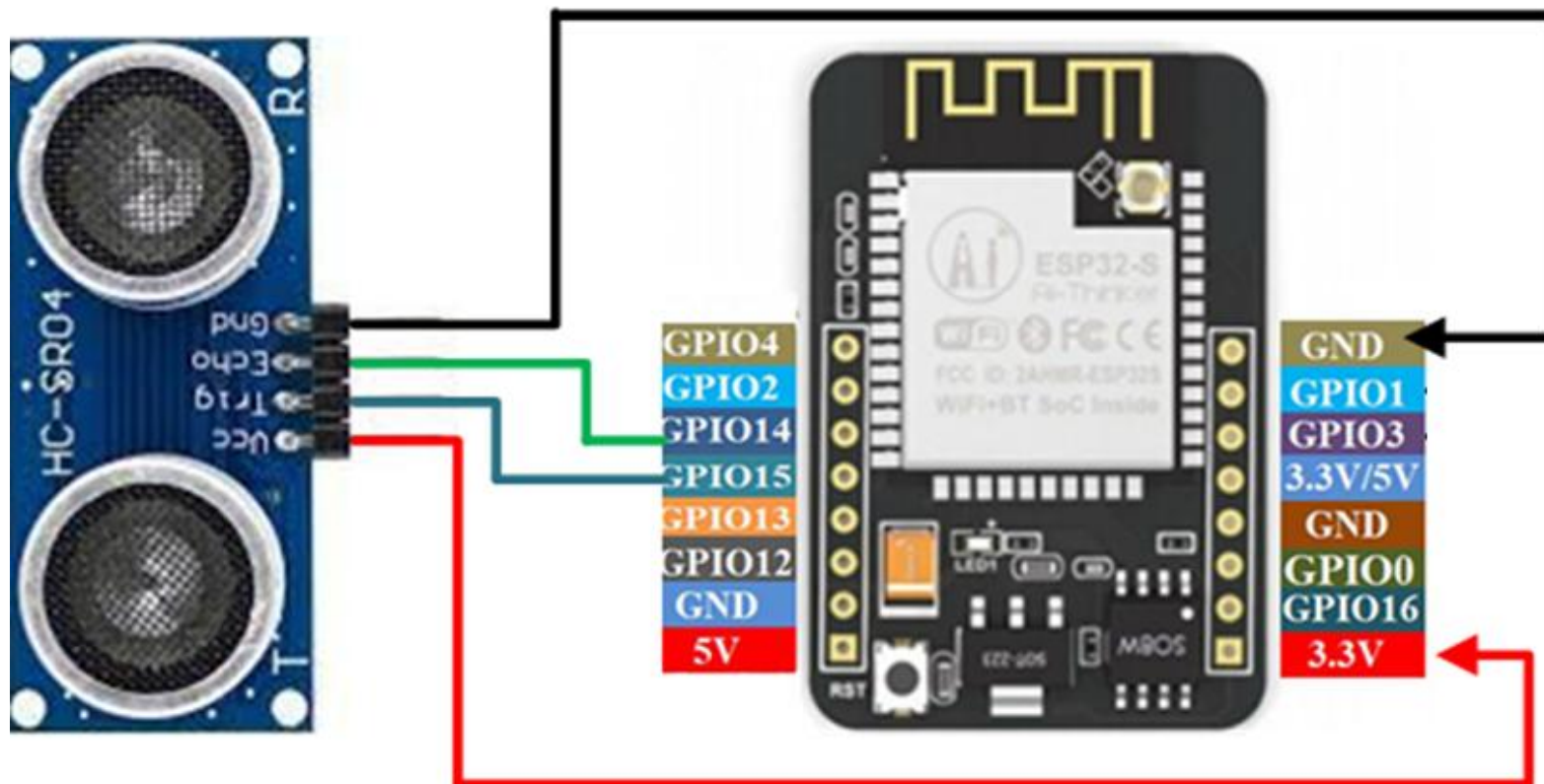
```
distance();  
if(S_DISTANCE>=20)  
    if(S_DISTANCE>=40) 條件  
        if(S_DISTANCE>=60)  
            {digitalWrite(LED[0],LOW);  
             digitalWrite(LED[1],LOW);  
             digitalWrite(LED[2],LOW);  
             digitalWrite(LED[3],HIGH);} 敘述1  
        else  
            {digitalWrite(LED[0],LOW);  
             digitalWrite(LED[1],LOW);  
             digitalWrite(LED[2],HIGH);  
             digitalWrite(LED[3],LOW);} 敘述2  
    else  
        {digitalWrite(LED[0],LOW);  
         digitalWrite(LED[1],HIGH);  
         digitalWrite(LED[2],LOW);  
         digitalWrite(LED[3],LOW);} 敘述3  
else  
    {digitalWrite(LED[0],HIGH);  
     digitalWrite(LED[1],LOW);  
     digitalWrite(LED[2],LOW);  
     digitalWrite(LED[3],LOW);} 敘述4  
}
```

A photograph of three people, two men and one woman, working together at a desk in a modern office. They are looking at a large computer monitor displaying a blue interface with text and a smaller laptop showing a colorful data visualization. One person is pointing at the monitor. A large yellow circle is overlaid on the image, containing the text '距離感測器' and '量測距離'.

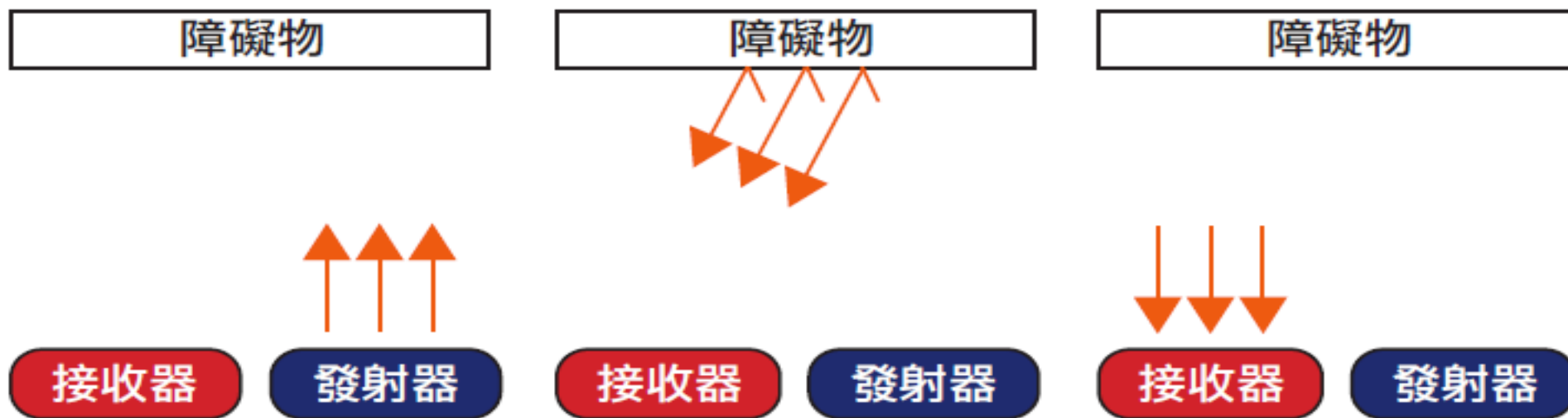
距離感測器
量測距離

硬體連接

超音波	ESP32
VCC	VCC
Trig	GPIO15
Echo	GPIO14
GND	GND



介紹知識

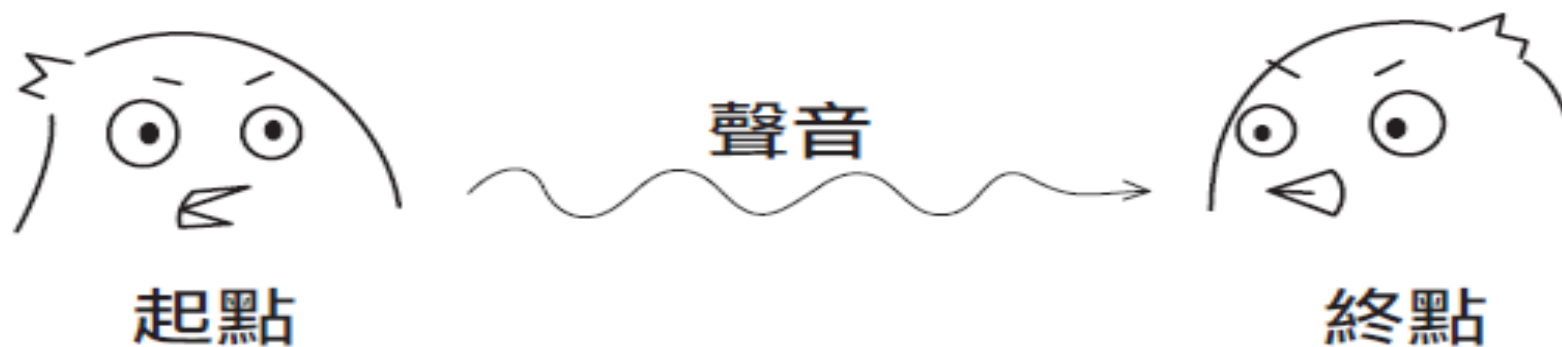


● 原理說明

超音波感測器與丟接球原理相同，感測器的右眼發出訊號，將超音波丟出，遇到障礙物時部分波反射，而感測器的左眼接收反射波。利用整個過程所花費的時間，可以推算出障礙物與反射器的距離。

介紹知識

● 運動學



聲波傳遞的總路徑長。
超音波感測器適用相同原理去計算車子與障礙物之間的距離。

$$\text{距離} = \text{速度} \times \text{時間}$$

每單位時間所走的距離，室溫下聲速是340m/s(每秒走340公尺)。

由起點(講者)至終點(聽者耳朵)所花費的時間。

$$\begin{aligned} \text{距離} \times 2 &= 34000 \text{ cm/s} \times (\text{Time_Value}) \times 1/1000000 \text{ s} \\ \text{距離} &= \text{Time_Value} \times 34 \times 1/1000 \times 1/2 \text{ cm} \end{aligned}$$

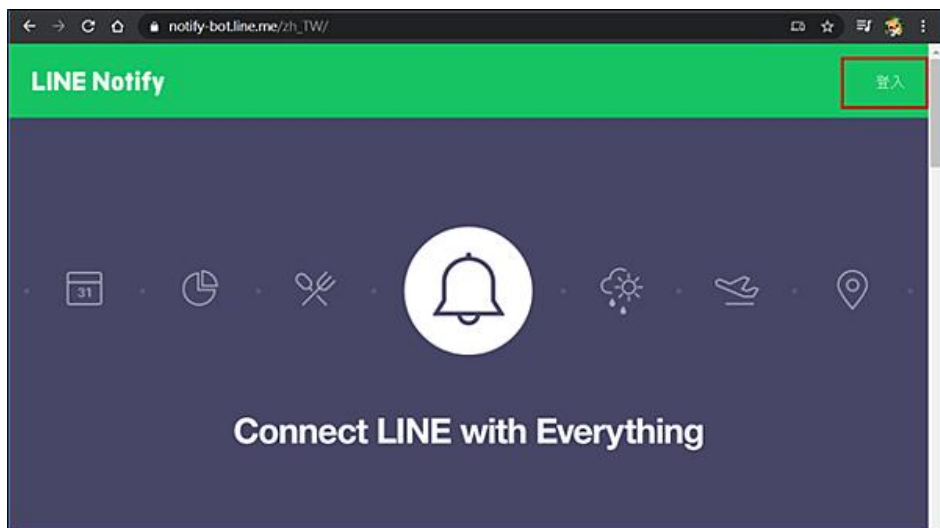
Arduino 實作：測試超音波量測距離

```
int TrigPin = 15; // 發出聲波
int EchoPin = 14; // 接收聲波
float Time_Value;
float CM_Value;
void setup() {
  Serial.begin(115200);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}
void loop() {
  digitalWrite(TrigPin, LOW); // 關閉
  delayMicroseconds(5);
  digitalWrite(TrigPin, HIGH); // 啟動
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW); // 關閉
  Time_Value = pulseIn(EchoPin, HIGH); // 傳回時間
  CM_Value = Time_Value * 34 / 1000 / 2; // 轉換成距離
  Serial.println(CM_Value);
  // Serial.println(" cm");
  delay(100);
}
```

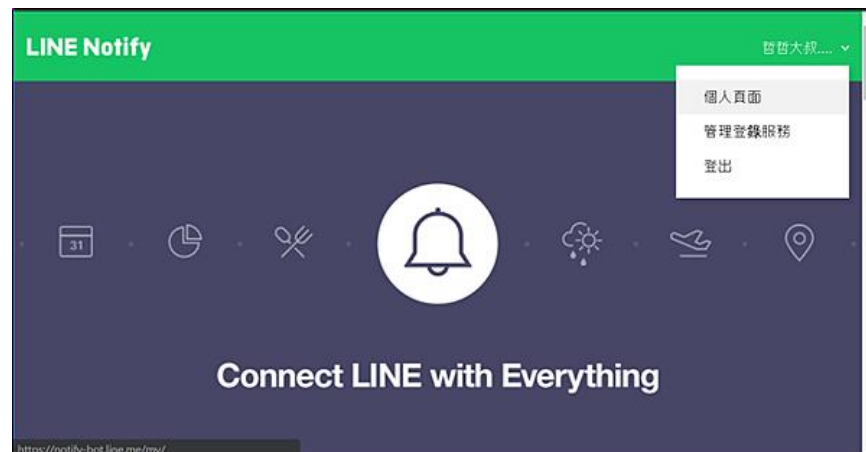
A background image showing three people (two men and one woman) wearing glasses and working on computers in a modern office environment. One person is pointing at a large monitor displaying data. A large yellow circle is overlaid on the image, containing the title text.

距離感測器+ESP32 拍照上傳LINE

LINE權杖的取得



<https://notify-bot.line.me/>



LINE權杖的取得

LINE Notify

哲哲大叔.....

已連動的服務

沒有已連動的服務。

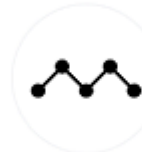
可連動的服務



GitHub >



IFTTT >



Mackerel >

發行存取權杖(開發人員用)

若使用個人存取權杖, 不須登錄網站服務, 即可設定通知。

發行權杖

LINE Notify API Document

發行權杖

×

請填寫權杖名稱(將於傳送提醒時顯示)

倉庫監看機器人

輸入通知機器人的名稱

請選擇您要接收通知的聊天室。

Search by group name



透過1對1聊天接收LINE Notify的通知



104行銷乙班

選擇要通知的群組



10709-IoT物聯網互動班



10711-IoT物聯網互動基礎班第01期



107無人商店的實務應用

※若公開個人存取權杖, 第三者將能取得您所連動的聊天室名稱及個人資料上的姓名。

發行

LINE權杖的取得



LINE權杖的測試



<https://apitester.com/>

1. 輸入Line Notify的相關設定如下圖。
修改傳遞方式POST
2. 輸入Line Notify API網站：<https://notify-api.line.me/api/notify>
3. 輸入傳遞的訊息內容：message=這是測試
點選「+」號，增加一個表頭Request Header，
我們會將密碼放在表頭與Line API做驗證
-輸入表頭名稱：Authorization
-輸入認證內容：Bearer 權杖密碼，要注意的是
Bearer與權杖密碼中間有一個空白。

完成設定後後，按下下方的藍色Test按鈕，即可在手機收到LINE傳來的訊息。

Arduino 實作: 距離感測器啟動拍照上傳LINE

宣告: 變數/引入函式庫/PIN腳位名稱

```
//The maximum size image is XGA(1024x768).  
// Enter your WiFi ssid and password  
const char* ssid = "SSID"; //your network SSID  
const char* password = "Password"; //your network  
password  
String myLineNotifyToken = "LINE權杖密碼"; //Line Notify  
Token · You can refer this post to get Line token :  
https://t.ly/LZwKn  
int trigPin = 15; //請將Trig接GPIO15  
int echoPin = 14; //Echo Pin 接GPIO14  
#include <WiFi.h>  
#include <WiFiClientSecure.h>  
#include "soc/soc.h"  
#include "soc/rtc_cntl_reg.h"  
#include "esp_camera.h"
```

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

```
#define CAMERA_MODEL_AI_THINKER
```

Arduino 實作: 距離感測器啟動拍照上傳LINE

宣告: 函式

```
String sendImage2LineNotify(String msg) {
    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get(); //取得相機影像放置fb
    if (!fb) {
        delay(100);
        Serial.println("Camera capture failed, Reset");
        ESP.restart();
    }
    WiFiClientSecure client_tcp; //啟動SSL wificlient
    Serial.println("Connect to notify-api.line.me");
    if (client_tcp.connect("notify-api.line.me", 443)) {
        Serial.println("Connection successful");
        String head = "--Taiwan\r\nContent-Disposition: form-
data; name=\"message\"; \r\n\r\n" + msg + "\r\n--
Taiwan\r\nContent-Disposition: form-data;
name=\"imageFile\"; filename=\"esp32-
cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
        String tail = "\r\n--Taiwan--\r\n";
        uint16_t imageLen = fb->len;
        uint16_t extraLen = head.length() + tail.length();
        uint16_t totalLen = imageLen + extraLen;
        //開始POST傳送訊息
        client_tcp.println("POST /api/notify HTTP/1.1");
        client_tcp.println("Connection: close");
        client_tcp.println("Host: notify-api.line.me");
        client_tcp.println("Authorization: Bearer " +
myLineNotifyToken);
        client_tcp.println("Content-Length: " + String(totalLen));
```

```
        client_tcp.println("Content-Type: multipart/form-data;
boundary=Taiwan");
        client_tcp.println();
        client_tcp.print(head);
        uint8_t *fbBuf = fb->buf;
        size_t fbLen = fb->len;
        Serial.println("Data Sending....");
        //照片 · 分段傳送
        for (size_t n = 0; n < fbLen; n = n + 2048) {
            if (n + 2048 < fbLen) {
                client_tcp.write(fbBuf, 2048);
                fbBuf += 2048;
            } else if (fbLen % 2048 > 0) {
                size_t remainder = fbLen % 2048;
                client_tcp.write(fbBuf, remainder);
            }
        }
        client_tcp.print(tail);
        client_tcp.println();
        String getResponse = "", Feedback = "";
        boolean state = false;
        int waitTime = 3000; // 依據網路調整等候時間 · 3000代表
最多等3秒
        long startTime = millis();
        delay(1000);
        Serial.print("Get Response");
        while ((startTime + waitTime) > millis()) {
            Serial.print(".");
```

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

Arduino 實作：距離感測器啟動拍照上傳LINE

Void Setup() 設定

```
void setup() {
  Serial.begin(115200);
  //初始化相機結束
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  config.jpeg_quality = 10; //10-63 lower number means
  higher quality
  config.fb_count = 2;
```

```
  config.frame_size = FRAMESIZE_XGA;// FRAMESIZE_ +
  UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
  //Line notify don't accept bigger than XGA
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }
  //初始化相機結束 · 開始網路連線
  WiFi.mode(WIFI_STA);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  long int StartTime = millis();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    if ((StartTime + 10000) < millis()) break;
  }
  Serial.println("");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("");
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Reset");
    delay(1000);
```

```
    ESP.restart();//連線不成功 · 則重新開機
  }
  pinMode(trigPin, OUTPUT); //Define inputs and outputs
  pinMode(echoPin, INPUT);
}
```

Arduino 實作: 距離感測器啟動拍照上傳LINE

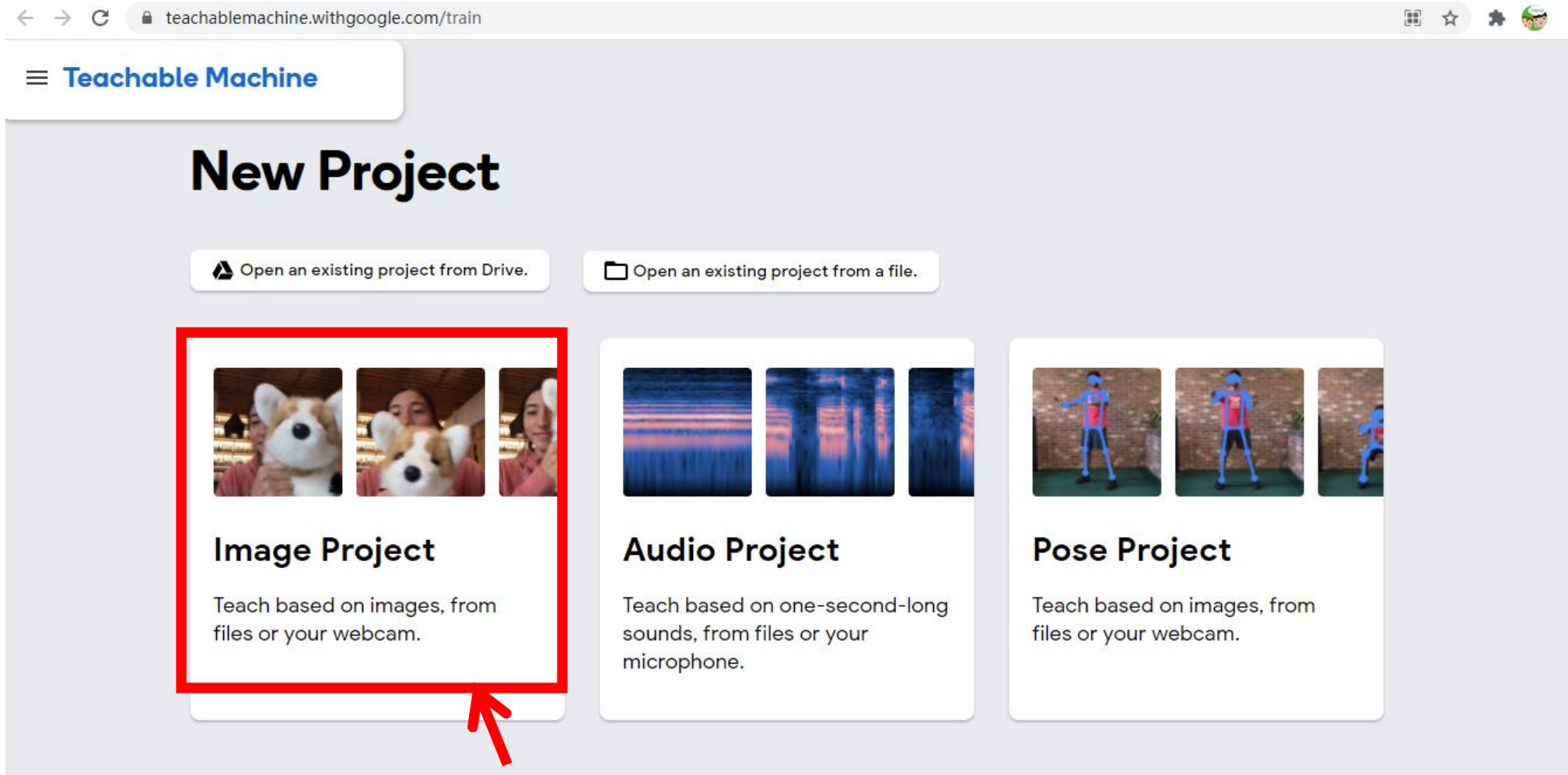
Void loop() 主程式

```
void loop() {  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(5);  
  digitalWrite(trigPin, HIGH);  // 給 Trig 高電位，持續 10微秒  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  long cm = pulseIn(echoPin, HIGH); // 收到高電位時的時間  
  cm = (cm / 2) / 29.1;  // 將時間換算成距離 cm 或 inch  
  Serial.println(cm);  
  if (cm <= 30) {  
    Serial.println("starting to Line");  
    String payload = sendImage2LineNotify("There is someone coming....");  
    Serial.println(payload);  
    delay(2000);  
  }  
  delay(100); //You could only send up to 50 images to Line Notify in one hour.  
}
```

A background image showing three people (two men and one woman) wearing glasses and looking at computer monitors in a modern office or lab setting. One person is pointing at a large monitor displaying data. A large yellow circle is overlaid on the image, containing white text.

AI辨識模型+ESP32
將判定結果
回傳至ESP32

Teachable Machine



<https://teachablemachine.withgoogle.com/train>

Teachable Machine

← → ↻ teachablemachine.withgoogle.com/train/image

≡ Teachable Machine

me 

18 Image Samples

hand 

30 Image Samples

 Add a class

Training

Model Trained

Advanced 

Epochs: 50  

Batch Size: 16  

Learning Rate: 0.001  

Reset Defaults 

Under the hood 

Preview  Export Model

Input  ON 

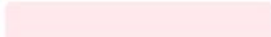




↓

Output

me 

hand 

可增加多種訓練的模型種類

Teachable Machine

← → ↻ teachablemachine.withgoogle.com/train/image

≡ Teachable Machine

me 

18 Image Samples



hand 

30 Image Samples



 Add a class

Training

Model Trained

Advanced 

Epochs: 50   

Batch Size: 16  

Learning Rate: 0.001   

 Reset Defaults 

可調整Epochs質, 每個套圖訓練的次數, 越多則Model越準確

Preview  Export Model

Input  ON File 

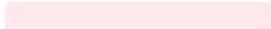
 Choose images from your files, or drag & drop here

 Import images from Google Drive

↓

Output

me 

hand 

Teachable Machine

← → ↻ teachablemachine.withgoogle.com/train/image

≡ Teachable Machine

me ✎

18 Image Samples



Webcam



Upload



hand ✎

30 Image Samples



Webcam



Upload



+ Add a class

按下Train Model之後
待處理程序完成，
可進入Preview

Training

Model Trained

Advanced

Epochs: 50

Batch Size: 16

Learning Rate:

0.001

Reset Defaults

Under the hood

Preview

Export Model

Input ☐ ON File

Choose images from your
files, or drag & drop here

Import images from
Google Drive

Output

me 100%

hand

Preview可選擇
視訊或圖片來
測試訓練出來
的模型

Teachable Machine

← → ↻ teachablemachine.withgoogle.com/train/image

≡ Teachable Machine

me 

18 Image Samples



hand 

30 Image Samples



 Add a class

Training

Model Trained

Advanced 

Epochs: 50  

Batch Size: 16  

Learning Rate: 0.001  

Reset Defaults 

Under the hood 

Preview 

Input  ON File 

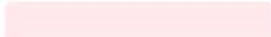




↓

Output

me 

hand 

輸出步驟1

Teachable Machine

Export your model to use it in projects. ×

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow Lite ⓘ

Export your model:

☒ Upload (shareable link) ☐ Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/HdosbGTow/> Copy

When you upload your model, Teachable Machine hosts it at this link for free. (FAQ: [Who can use my model?](#))

✓ Your cloud model is up to date.

Code snippets to use your model:

Javascript p5.js [Contribute on Github](#)

Learn more about how to use the code snippet on [github](#).

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-
image.min.js"></script>
<script type="text/javascript">
```

輸出步驟2

複製連結

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告: 變數/引入函式庫/PIN腳位名稱

```
//輸入Wi-Fi帳號
const char* ssid = "*****"; //Wi-Fi帳號
const char* password = "*****"; //Wi-Fi密碼
```

```
//輸入AP端連線帳號密碼
const char* apssid = "ESP32-CAM";
const char* appassword = "12345678"; //AP端密碼至少要八個字元以上
```

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "esp_camera.h" //視訊
#include "soc/soc.h" //用於電源不穩不重開機
#include "soc/rtc_cntl_reg.h" //用於電源不穩不重開機
String Feedback=""; //回傳客戶端訊息
//指令參數值
String
Command="",cmd="",P1="",P2="",P3="",P4="",P5="",P6="",P7="",P8
="",P9="";
```

```
//指令拆解狀態值
byte ReceiveState=0,cmdState=1,strState=1,questionstate=0,
equalstate=0,semicolonstate=0;
```

```
//ESP32-CAM模組腳位設定
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22
WiFiServer server(80);
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告: 函式

```
void ExecuteCommand()
{
    //Serial.println("");
    //Serial.println("Command: "+Command);
    if (cmd!="getstill") {
        Serial.println("cmd= "+cmd+" ,P1= "+P1+" ,P2= "+P2+"
,P3= "+P3+" ,P4= "+P4+" ,P5= "+P5+" ,P6= "+P6+" ,P7=
"+P7+" ,P8= "+P8+" ,P9= "+P9);
        Serial.println("");
    }

    //自訂指令區
    if (cmd=="your cmd") {
        // You can do anything.
        // Feedback="<font color=\red\">Hello
World</font>";
    }
    else if (cmd=="ip") {
        Feedback="AP IP: "+WiFi.softAPIP().toString();
        Feedback+=" ";
        Feedback+="STA IP: "+WiFi.localIP().toString();
    }
    else if (cmd=="mac") {
        Feedback="STA MAC: "+WiFi.macAddress();
    }
    else if (cmd=="resetwifi") {
        WiFi.begin(P1.c_str(), P2.c_str());
    }
}
```

```
Serial.print("Connecting to ");
Serial.println(P1);
long int StartTime=millis();
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    if ((StartTime+5000) < millis()) break;
}
Serial.println("");
Serial.println("STAIP: "+WiFi.localIP().toString());
Feedback="STAIP: "+WiFi.localIP().toString();
}
else if (cmd=="restart") {
    ESP.restart();
}
else if (cmd=="digitalwrite") {
    ledcDetachPin(P1.toInt());
    pinMode(P1.toInt(), OUTPUT);
    digitalWrite(P1.toInt(), P2.toInt());
}
else if (cmd=="analogwrite") {
    if (P1=="4") {
        ledcAttachPin(4, 4);
        ledcSetup(4, 5000, 8);
        ledcWrite(4,P2.toInt());
    }
    else {
```

```
        ledcAttachPin(P1.toInt(), 5);
        ledcSetup(5, 5000, 8);
        ledcWrite(5,P2.toInt());
    }
}
else if (cmd=="flash") {
    ledcAttachPin(4, 4);
    ledcSetup(4, 5000, 8);

    int val = P1.toInt();
    ledcWrite(4,val);
}
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告: 函式

```
else if (cmd=="framesize") {
  sensor_t * s = esp_camera_sensor_get();
  if (P1=="0")
    s->set_framesize(s, FRAMESIZE_QQVGA);
  else if (P1=="3")
    s->set_framesize(s, FRAMESIZE_HQVGA);
  else if (P1=="4")
    s->set_framesize(s, FRAMESIZE_QVGA);
  else if (P1=="5")
    s->set_framesize(s, FRAMESIZE_CIF);
  else if (P1=="6")
    s->set_framesize(s, FRAMESIZE_VGA);
  else if (P1=="7")
    s->set_framesize(s, FRAMESIZE_SVGA);
  else if (P1=="8")
    s->set_framesize(s, FRAMESIZE_XGA);
  else if (P1=="9")
    s->set_framesize(s, FRAMESIZE_SXGA);
  else if (P1=="10")
    s->set_framesize(s, FRAMESIZE_UXGA);
  else
    s->set_framesize(s, FRAMESIZE_QVGA);
}
else if (cmd=="quality") {
  sensor_t * s = esp_camera_sensor_get();
  int val = P1.toInt();
  s->set_quality(s, val);
```

```
}
else if (cmd=="contrast") {
  sensor_t * s = esp_camera_sensor_get();
  int val = P1.toInt();
  s->set_contrast(s, val);
}
else if (cmd=="brightness") {
  sensor_t * s = esp_camera_sensor_get();
  int val = P1.toInt();
  s->set_brightness(s, val);
}
else if (cmd=="serial") {
  if (P1!=""&P1!="stop") Serial.println(P1);
  if (P2!=""&P2!="stop") Serial.println(P2);
  Serial.println();
}
else {
  Feedback="Command is not defined.";
}
if (Feedback=="") Feedback=Command;
}}
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

Void Setup() 設定

```
void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //關閉
  電源不穩就重開機的設定
```

```
  Serial.begin(115200);
  Serial.setDebugOutput(true); //開啟診斷輸出
  Serial.println();
```

//視訊組態設定

```
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
```

```
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10; //0-63 lower number means
    higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12; //0-63 lower number means
    higher quality
    config.fb_count = 1;
  }
```

```
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }
```

```
  //drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_QVGA);
  //UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
  設定初始化影像解析度
```

```
  //閃光燈
  ledcAttachPin(4, 4);
  ledcSetup(4, 5000, 8);
```

```
  WiFi.mode(WIFI_AP_STA);
```

```
  //指定Client端靜態IP
  //WiFi.config(IPAddress(192, 168, 201, 100),
  IPAddress(192, 168, 201, 2), IPAddress(255, 255, 255, 0));
```

```
  WiFi.begin(ssid, password); //執行網路連線
```

```
  delay(1000);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

```
  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    if ((StartTime+10000) < millis()) break; //等待10秒連線
  }
```

```
  if (WiFi.status() == WL_CONNECTED) { //若連線成功
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

Void Setup() 設定

```
WiFi.softAP((WiFi.localIP().toString()+"_"+(String)apssid).c_str(), appassword); //設定SSID顯示客戶端IP
Serial.println("");
Serial.println("STAIP address: ");
Serial.println(WiFi.localIP());

for (int i=0;i<5;i++) { //若連上WIFI設定閃光燈快速閃爍
  ledcWrite(4,10);
  delay(200);
  ledcWrite(4,0);
  delay(200);
}
}
else {

WiFi.softAP((WiFi.softAPIP().toString()+"_"+(String)apssid).c_str(), appassword);

for (int i=0;i<2;i++) { //若連不上WIFI設定閃光燈慢速閃爍
  ledcWrite(4,10);
  delay(1000);
  ledcWrite(4,0);
  delay(1000);
}
}

//指定AP端IP
WiFi.softAPConfig(IPAddress(192, 168, 4, 1), IPAddress(255, 255, 255, 0));
Serial.println("");
Serial.println("APIP address: ");
Serial.println(WiFi.softAPIP());

pinMode(4, OUTPUT);
digitalWrite(4, LOW);

server.begin();
}
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告:自訂網頁內容

```
//自訂網頁首頁管理介面
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-
scale=1">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.
min.js"> </script>
<script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/
tf.min.js"> </script>
<script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image
@0.8/dist/teachablemachine-image.min.js"> </script>
<script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@
0.8/dist/teachablemachine-pose.min.js"> </script>
</head>
<body>
<div id="container"> </div>
<img id="ShowImage" src="" style="display:none">
<canvas id="canvas" style="display:none"> </canvas>
<table>
<tr>
<td><input type="button" id="restart"
value="Restart"> </td>
<td colspan="2"><input type="button" id="getStill"
value="Get Still" style="display:none"> </td>
```

```
</tr>
<tr>
<td>Model kind</td>
<td colspan="2">
<select id="kind">
<option value="image">image</option>
<option value="pose">pose</option>
</select>
</td>
</tr>
<tr>
<td>Model Path</td>
<td colspan="2"><input type="text" id="modelPath"
value=""> </td>
</tr>
<tr>
<td></td>
<td colspan="2"><button type="button" id="btnModel"
onclick="LoadModel();">Start Recognition</button> </td>
</tr>
<tr>
<td>MirrorImage</td>
<td colspan="2">
<select id="mirrorimage">
<option value="1">yes</option>
<option value="0">no</option>
</select>
</td>
</tr>
</table>
```

```
<td>Resolution</td>
<td colspan="2">
<select id="framesize">
<option value="UXGA">UXGA(1600x1200)</option>
<option value="SXGA">SXGA(1280x1024)</option>
<option value="XGA">XGA(1024x768)</option>
<option value="SVGA">SVGA(800x600)</option>
<option value="VGA">VGA(640x480)</option>
<option value="CIF">CIF(400x296)</option>
<option value="QVGA"
selected="selected">QVGA(320x240)</option>
<option value="HQVGA">HQVGA(240x176)</option>
<option value="QQVGA">QQVGA(160x120)</option>
</select>
</td>
</tr>
<tr>
<td>Flash</td>
<td colspan="2"><input type="range" id="flash" min="0"
max="255" value="0"> </td>
</tr>
<tr>
<td>Quality</td>
<td colspan="2"><input type="range" id="quality" min="10"
max="63" value="10"> </td>
</tr>
<tr>
<td>Brightness</td>
<td colspan="2"><input type="range" id="brightness"
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告:自訂網頁內容

```
min="-2" max="2" value="0"></td>
</tr>
<tr>
<td>Contrast</td>
<td colspan="2"><input type="range" id="contrast" min="-2"
max="2" value="0"></td>
</tr>
<tr>
<td>Rotate</td>
<td align="left" colspan="2">
<select
onchange="document.getElementById('canvas').style.transform
='rotate('+this.value+')';">
<option value="0deg">0deg</option>
<option value="90deg">90deg</option>
<option value="180deg">180deg</option>
<option value="270deg">270deg</option>
</select>
</td>
</tr>
</table>
<div id="result" style="color:red"><div>
</body>
</html>

<script>
var getStill = document.getElementById('getStill');
var ShowImage = document.getElementById('ShowImage');
var canvas = document.getElementById("canvas");
var context = canvas.getContext("2d");
```

```
var mirrorimage = document.getElementById("mirrorimage");
var flash = document.getElementById('flash');
var modelPath = document.getElementById('modelPath');
var result = document.getElementById('result');
var kind = document.getElementById('kind');
var myTimer;
var restartCount=0;
async function LoadModel() {
if (modelPath.value=="") {
result.innerHTML = "Please input model path.";
return;
}

result.innerHTML = "Please wait for loading model.";

const URL = modelPath.value;
const modelURL = URL + "model.json";
const metadataURL = URL + "metadata.json";

if (kind.value=="image") {
Model = await tmImage.load(modelURL, metadataURL);
}
else if (kind.value=="pose") {
Model = await tmPose.load(modelURL, metadataURL);
}
maxPredictions = Model.getTotalClasses();
result.innerHTML = "";

getStill.style.display = "block";
getStill.click();
```

```
}

getStill.onclick = function (event) {
clearInterval(myTimer);
myTimer = setInterval(function(){error_handle();},5000);
ShowImage.src=location.origin+'/?getstill='+Math.random();
}

function error_handle() {
restartCount++;
clearInterval(myTimer);
if (restartCount<=2) {
//message.innerHTML = "Get still error. <br>Restart ESP32-
CAM "+restartCount+" times.";
myTimer = setInterval(function(){getStill.click();},6000);
}
else
message.innerHTML = "Get still error. <br>Please close the
page and check ESP32-CAM.";
}

ShowImage.onload = function (event) {
clearInterval(myTimer);
restartCount=0;
canvas.setAttribute("width", ShowImage.width);
canvas.setAttribute("height", ShowImage.height);
canvas.style.display = "block ";
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告:自訂網頁內容

```

if (mirrorimage.value==1) {
    context.translate((canvas.width + ShowImage.width) / 2, 0);
    context.scale(-1, 1);
    context.drawImage(ShowImage, 0, 0, ShowImage.width,
ShowImage.height);
    context.setTransform(1, 0, 0, 1, 0, 0);
}
else

context.drawImage(ShowImage,0,0,ShowImage.width,ShowImage.height);
predict();
}

restart.onclick = function (event) {
    fetch(location.origin+'/?restart=stop');
}
framesize.onclick = function (event) {

fetch(document.location.origin+'/?framesize='+this.value+';stop
');
}
flash.onchange = function (event) {
    fetch(location.origin+'/?flash='+this.value+';stop');
}
quality.onclick = function (event) {

fetch(document.location.origin+'/?quality='+this.value+';stop');
}
brightness.onclick = function (event) {

```

```

fetch(document.location.origin+'/?brightness='+this.value+';stop');
}
contrast.onclick = function (event) {

fetch(document.location.origin+'/?contrast='+this.value+';stop');
}

async function predict() {
    var data = "";
    var maxClassName = "";
    var maxProbability = "";

    canvas.setAttribute("width", ShowImage.width);
    canvas.setAttribute("height", ShowImage.height);
    context.drawImage(ShowImage, 0, 0, ShowImage.width,
ShowImage.height);

    if (kind.value=="image")
        var prediction = await Model.predict(canvas);
    else if (kind.value=="pose") {
        var { pose, posenetOutput } = await
Model.estimatePose(canvas);
        var prediction = await Model.predict(posenetOutput);
    }

    if (maxPredictions>0) {
        for (let i = 0; i < maxPredictions; i++) {
            if (i==0) {

```

```

maxClassName = prediction[i].className;
maxProbability = prediction[i].probability;
}
else {
    if (prediction[i].probability>maxProbability) {
        maxClassName = prediction[i].className;
        maxProbability = prediction[i].probability;
    }
}
data += prediction[i].className + "," +
prediction[i].probability.toFixed(2) + "<br>";
}
result.innerHTML = data;
result.innerHTML += "<br>Result: " + maxClassName + "," +
maxProbability;

$.ajax({url:
document.location.origin+'/?serial='+maxClassName+"," +maxPr
obability+';stop', async: false});
}
else
    result.innerHTML = "Unrecognizable";

getStill.click();
}
</script>
rawliteral";

```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

Void loop() 主程式

```
void loop() {

Feedback="";Command="";cmd="";P1="";P2="";P3="";P4="";P5="";P6="";P7="";P8="";P9="";

ReceiveState=0,cmdState=1,strState=1,questionstate=0,equalstate=0,semicolonstate=0;

WiFiClient client = server.available();

if (client) {
String currentLine = "";

while (client.connected()) {
if (client.available()) {
char c = client.read();

getCommand(c); //將緩衝區取得的字元拆解出指令參數

if (c == '\n') {
if (currentLine.length() == 0) {

if (cmd=="getstill") {
//回傳JPEG格式影像
camera_fb_t * fb = NULL;
fb = esp_camera_fb_get();
if(!fb) {
```

```
Serial.println("Camera capture failed");
delay(1000);
ESP.restart();
}

client.println("HTTP/1.1 200 OK");
client.println("Access-Control-Allow-Origin: *");
client.println("Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept");
client.println("Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS");
client.println("Content-Type: image/jpeg");
client.println("Content-Disposition: form-data; name=\"imageFile\"; filename=\"picture.jpg\"");
client.println("Content-Length: " + String(fb->len));
client.println("Connection: close");
client.println();

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n=0;n<fbLen;n=n+1024) {
if (n+1024<fbLen) {
client.write(fbBuf, 1024);
fbBuf += 1024;
}
else if (fbLen%1024>0) {
size_t remainder = fbLen%1024;
```

```
client.write(fbBuf, remainder);
}
}

esp_camera_fb_return(fb);

pinMode(4, OUTPUT);
digitalWrite(4, LOW);
}
```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

Void loop() 主程式

```

else {
  //回傳HTML首頁或Feedback
  client.println("HTTP/1.1 200 OK");
  client.println("Access-Control-Allow-Headers:
Origin, X-Requested-With, Content-Type, Accept");
  client.println("Access-Control-Allow-Methods:
GET,POST,PUT,DELETE,OPTIONS");
  client.println("Content-Type: text/html;
charset=utf-8");
  client.println("Access-Control-Allow-Origin: *");
  client.println("Connection: close");
  client.println();

  String Data="";
  if (cmd!="")
    Data = Feedback;
  else {
    Data = String((const char *)INDEX_HTML);
  }
  int Index;
  for (Index = 0; Index < Data.length(); Index =
Index+1000) {
    client.print(Data.substring(Index, Index+1000));
  }

  client.println();
}

```

```

}

Feedback="";
break;
} else {
  currentLine = "";
}
}
else if (c != '\r') {
  currentLine += c;
}

if ((currentLine.indexOf("/")!=-
1)&&(currentLine.indexOf(" HTTP")!=-1)) {
  if (Command.indexOf("stop")!=-1) { //若指令中含關鍵
字stop立即斷線 ->
http://192.168.xxx.xxx/?cmd=aaa;bbb;ccc;stop
    client.println();
    client.println();
    client.stop();
  }
  currentLine="";
  Feedback="";
  ExecuteCommand();
}
}
}
}

```

```

delay(1);
client.stop();
}
}

```

Arduino 實作: ESP32+ AI辨識模型判定結果回傳

宣告: 函式

```
//拆解命令字串置入變數
void getCommand(char c)
{
    if (c=='?') ReceiveState=1;
    if ((c==' ')|| (c=='\r') || (c=='\n')) ReceiveState=0;

    if (ReceiveState==1)
    {
        Command=Command+String(c);

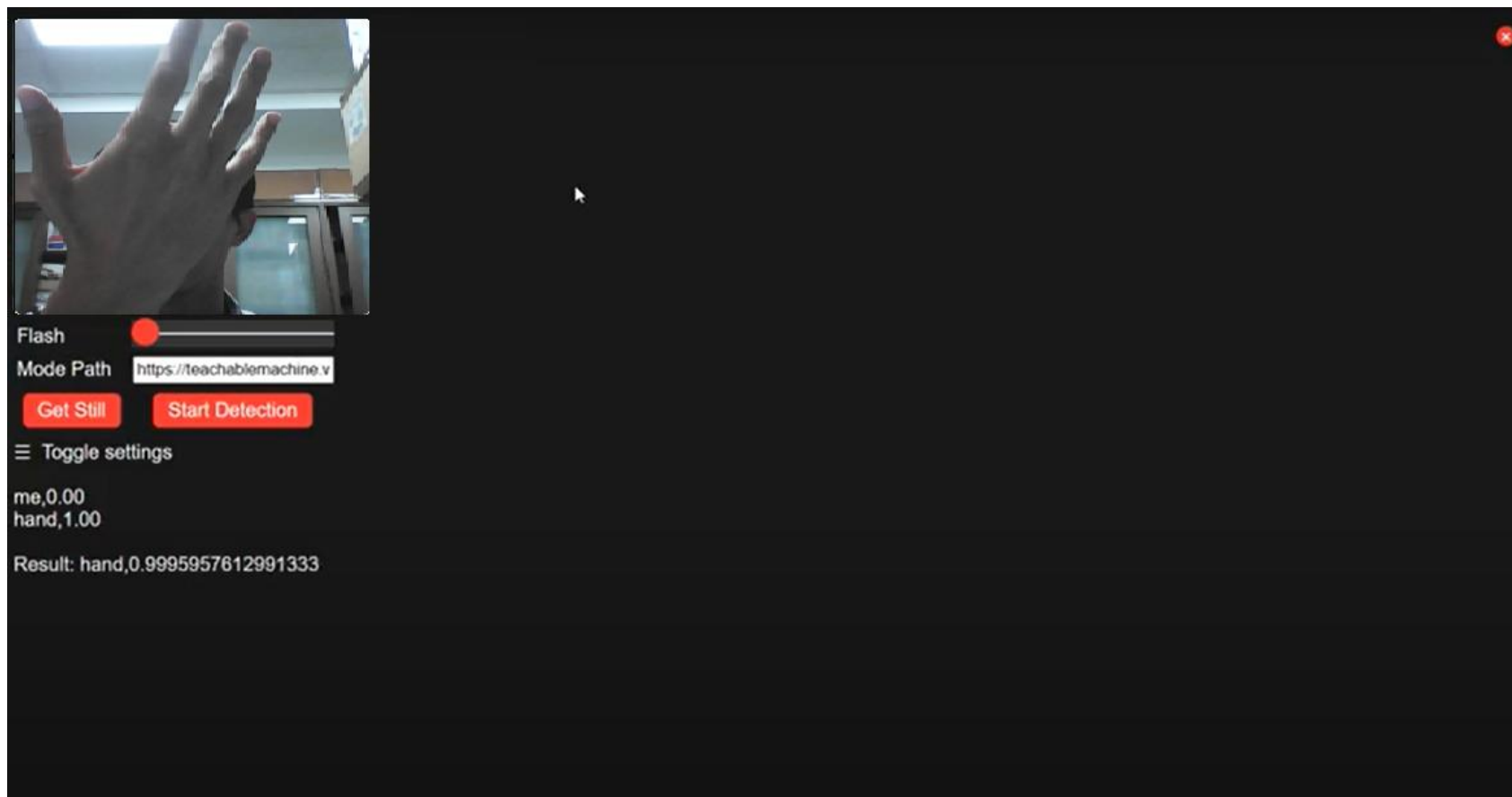
        if (c=='=') cmdState=0;
        if (c==';') strState++;

        if ((cmdState==1)&&((c!='?')||(questionstate==1)))
        cmd=cmd+String(c);
        if
        ((cmdState==0)&&(strState==1)&&((c!=' ')||(equalstate=
        =1))) P1=P1+String(c);
        if ((cmdState==0)&&(strState==2)&&(c!=';'))
        P2=P2+String(c);
        if ((cmdState==0)&&(strState==3)&&(c!=';'))
        P3=P3+String(c);
        if ((cmdState==0)&&(strState==4)&&(c!=';'))
        P4=P4+String(c);
        if ((cmdState==0)&&(strState==5)&&(c!=';'))
        P5=P5+String(c);
        if ((cmdState==0)&&(strState==6)&&(c!=';'))
```

```
P6=P6+String(c);
    if ((cmdState==0)&&(strState==7)&&(c!=';'))
    P7=P7+String(c);
    if ((cmdState==0)&&(strState==8)&&(c!=';'))
    P8=P8+String(c);
    if
    ((cmdState==0)&&(strState>=9)&&((c!=';')||(semicolonsta
    te==1))) P9=P9+String(c);

    if (c=='?') questionstate=1;
    if (c=='=') equalstate=1;
    if ((strState>=9)&&(c!=';')) semicolonstate=1;
    }
}
```

Teachable Machine with ESP32

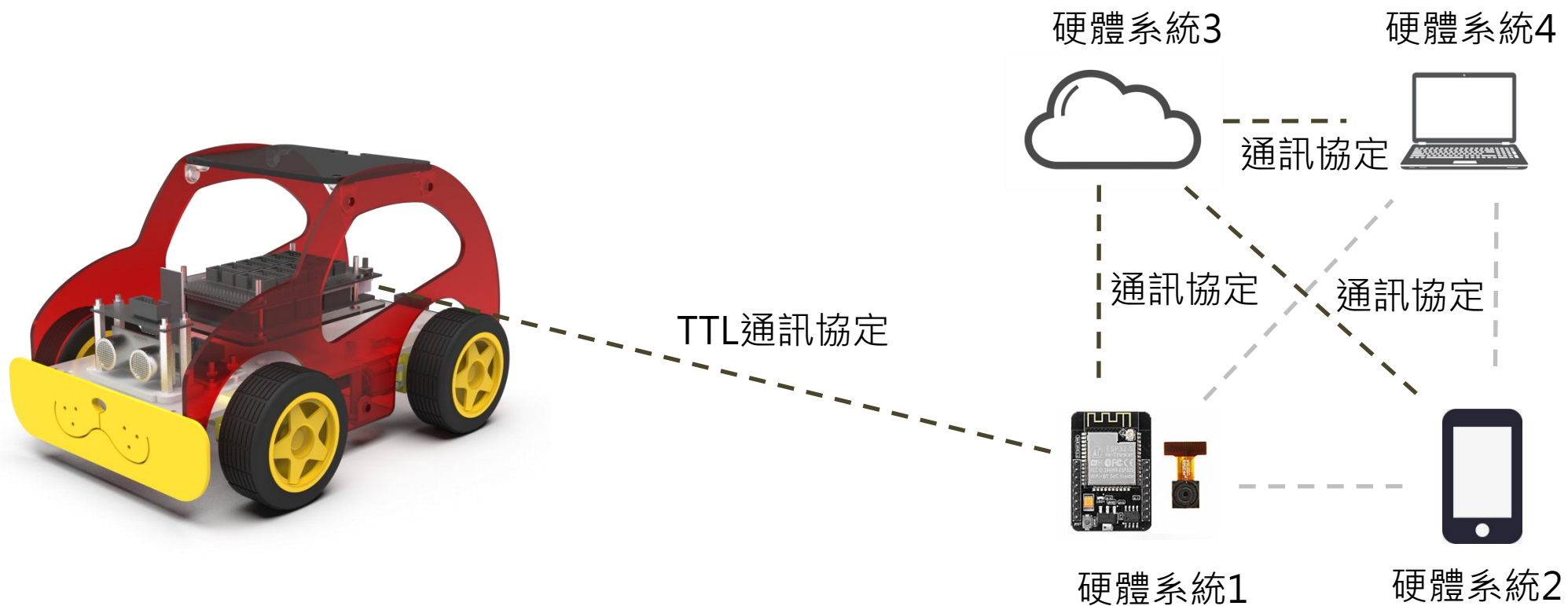


ESP32影
像透過模
型輸出辨
識結果

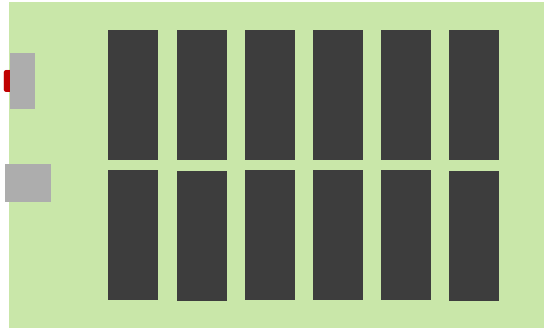
A photograph of three students, two girls and one boy, all wearing glasses, looking intently at a large computer monitor. One girl is pointing at the screen. The monitor displays a blue interface with text and a bar chart. A yellow circle is overlaid on the image, containing the text '專題思考' and '自動車的應用'.

專題思考
自動車的應用

專題實作



BeeCar主板GPIO定義



PIN definitions:

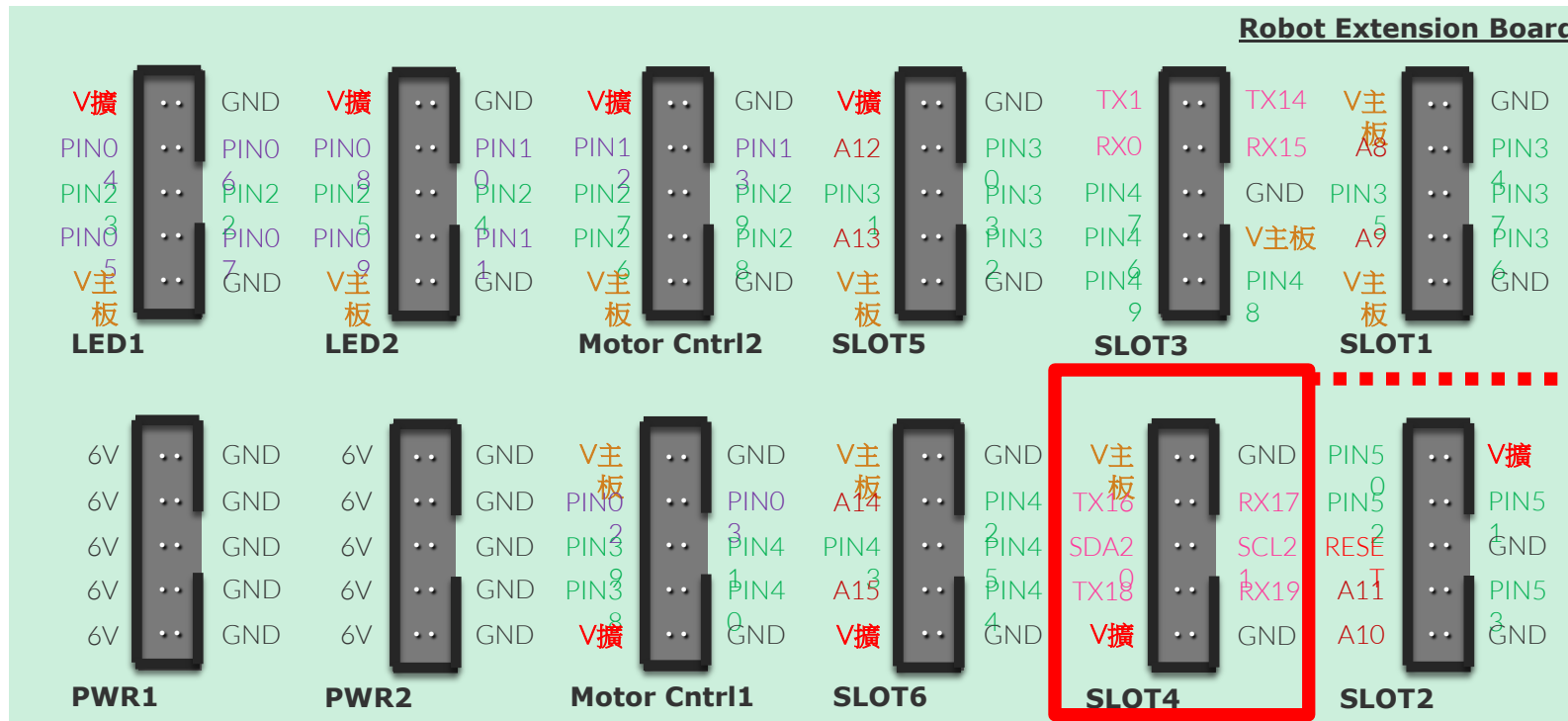
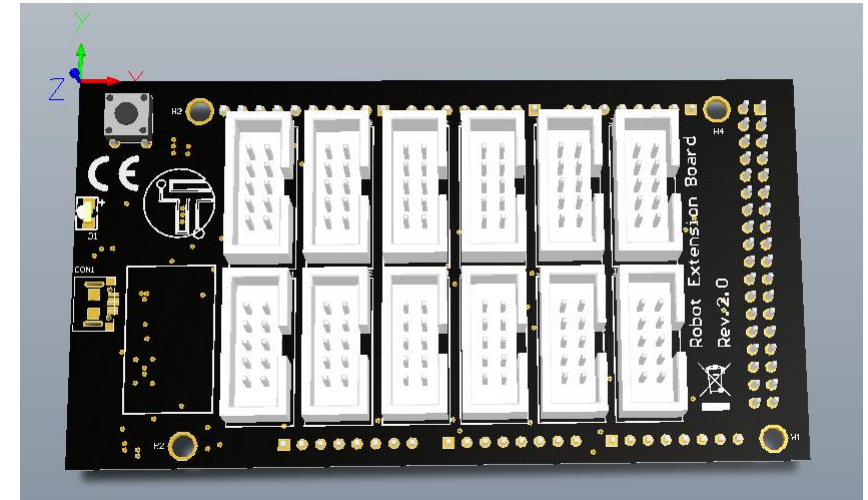
PWM PIN: 2~13

Digital PIN: 22~49

Analog PIN: A8~A15

TTL/SDI: 14~21

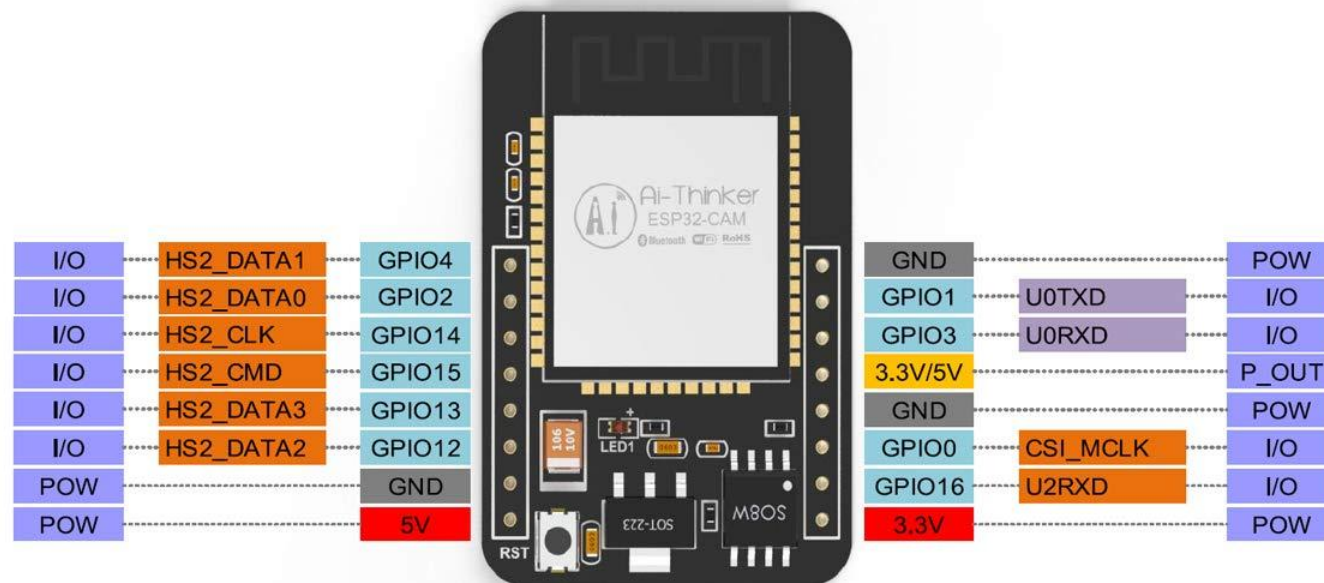
V主板/GND



TTL通信連接



專題實作



Mega2560	ESP32
TX1 (PIN18)	PIN3
RX1 (PIN19)	PIN1
5V	5V
GND	GND

UART
channel

透過UART, ESP32與Arduino2560進行通信

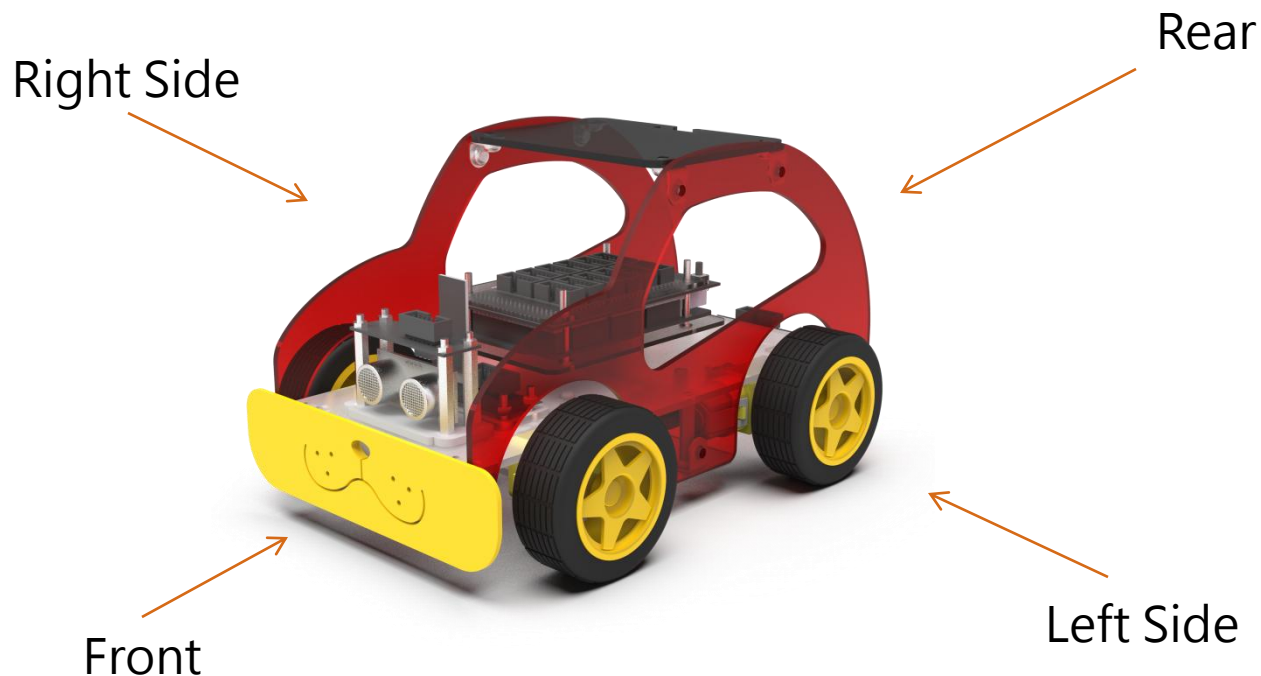
CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/闪光灯	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

Pin Definition

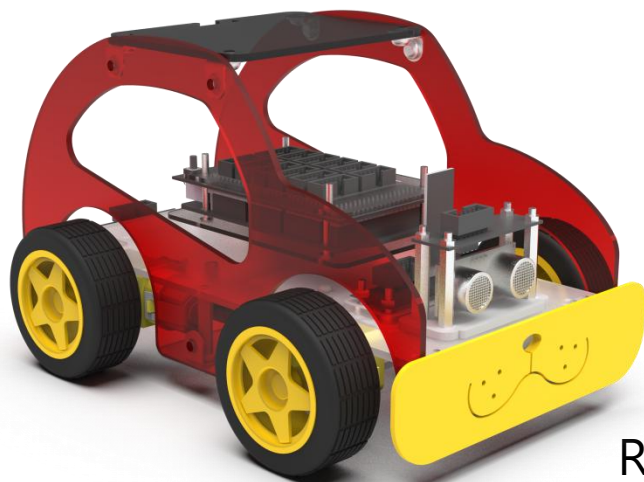
專題實作

將Teachable Machine訓練模型應用在BeeCar小車，製作出簡單的影像辨識自動會車系統：

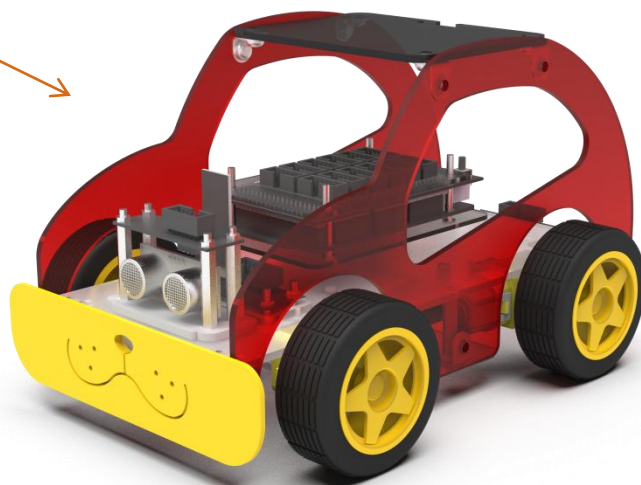
1. 將車子的四個面分別定義四種不同的Class
2. 訓練模型判斷特徵
3. 區分出不同會車方向，Arduino分別定義動作
4. 可加入距離感測器判斷示於車子動作邏輯之中。



專題實作



Right Side



實例:

1. 比如A車從右側駛來, 須立即停止讓B車通行。
2. 若A車由前方正面駛來, 向右微轉改面行進方向。
3. 若A車由後方駛來, 須減速與前車保持距離。

專題實作

可以運用到自動車功能上嗎？

1. 辨識車身方向
2. 偵測車體距離
3. 預測車子行進方向

A photograph of three students, two girls and one boy, looking intently at a large computer monitor. The monitor displays a blue-themed interface with text and a bar chart. One girl is pointing at the screen. A yellow circle is overlaid on the image, containing the text '專題思考 未來的應用'.

專題思考
未來的應用

專題思考

總結上述所學：

1. 是否可以改動Code將辨識結果回傳至Line。
2. 可以聯想到什麼生活應用。

專題思考

可以運用到自動車功能上嗎？

1. 辨識交通燈號誌
2. 辨識行人穿越
3. 辨識路上標線、路邊標示



TARKUS
i n n o v a t i o n

THANK YOU!



www.tarkustech.com