

超入門！

AI 聲音姿態影像辨識大應用

賴秉樑 debugger



PERSONAL INFO

姓名 NAME

賴秉樑 debugger

學歷 EDUCATION

台科大資工、電子雙學位
中興資訊科學與工程碩士

經歷 EXPERIENCE

國立大學電子系、資工系講師
職業訓練、產投電腦講師
竹科半導體研發工程師

個人興趣 INTEREST

玩數學、打電腦

我的座右銘 MOTTO

動手做、樂趣多

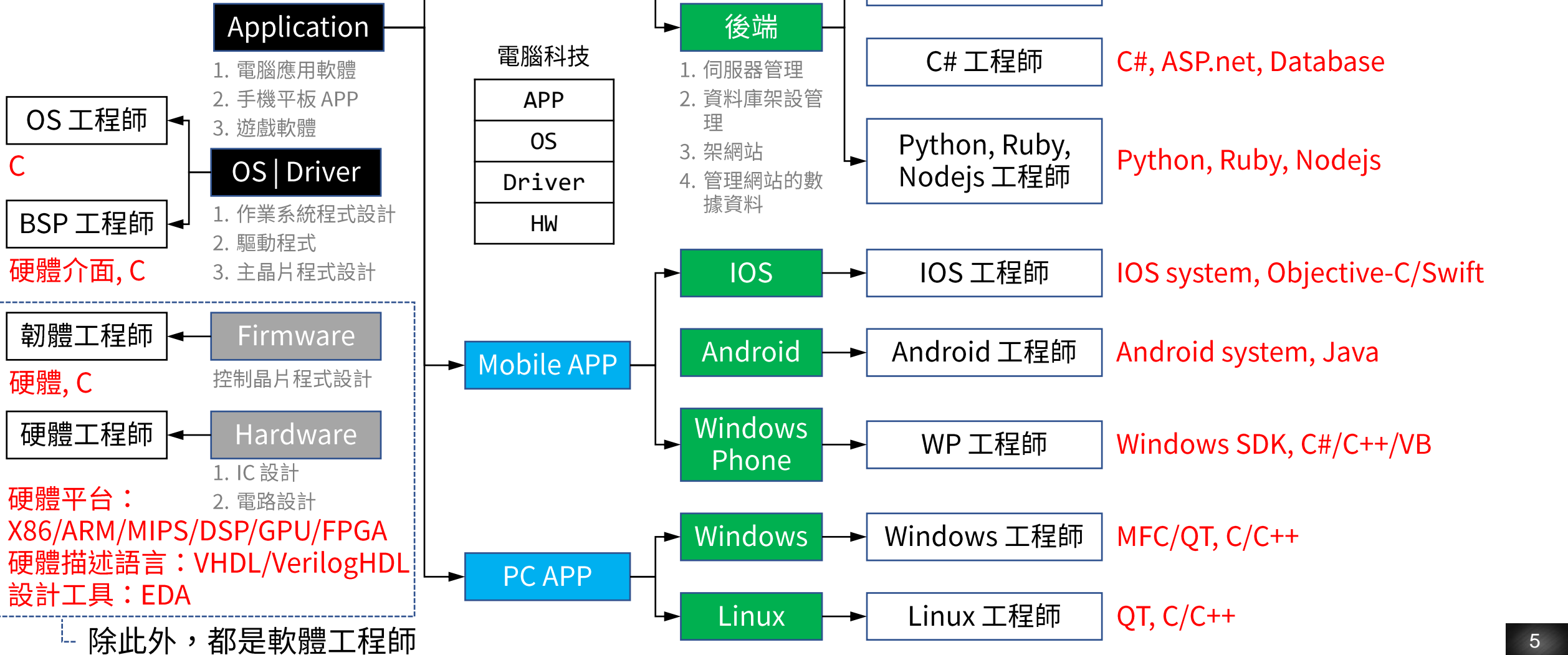
Outline

- 美好的程式世界
- 智慧物聯網 (The Artificial Internet of Things, IoT)
- 9 個小時的課程：AI 聲音姿態影像辨識大應用

Coder, Hacker, and Maker

- 程式設計師 (coder, 或 programmer)
 - ✓ 主要透過編輯程式，簡稱編程 (coding)，它可以指在程式設計**某個專業領域的專業人士**，或是從事軟體撰寫，程式開發、維護的專業人員。
- 駭客 (hacker)
 - ✓ 除了**精通**程式設計、作業系統的人可以被視作駭客，對硬體裝置做創新的工程師通常也被認為是駭客，精通網路入侵的人也被看作是駭客。
- 創客 (maker)
 - ✓ 又稱自造者。是一群酷愛科技、熱衷實踐的人群，他們以分享技術、**激發的創造力**與交流思想為樂。

科技業工程師 主要分類



運算思維為何很重要？ (1/2)

- 學會了運算思維，讓我們也能擁有電腦科學家面對問題時，所持有的科學方法。各種領域都需要運算思維，例如：
- **科學與工程領域**
 - ✓ 利用運算模擬建築結構，以確認安全性。
 - ✓ 利用運算預測氣象，以增加準確性。
- **金融領域**
 - ✓ 利用運算研究經濟大數據。
 - ✓ 利用運算完成自動交易。

運算思維為何很重要？ (2/2)

- **人文與社會領域**

- ✓ 利用運算分析，並優化廣告投放策略。
- ✓ 利用運算分析人口老化趨勢，與醫療資源分布。

- **藝術領域**

- ✓ 利用運算建構三維動畫。
- ✓ 利用運算創作數位音樂。

- **工業設計**

- ✓ 利用運算實現工業 4.0。
- ✓ 利用運算實現更多的增值應用，例如：自動化與智慧化。

科技業的下一個神話

電子商務 X 社群網路 X 串流媒體



物聯網 X 人工智慧 X 區塊鏈

數位浪潮帶動新經濟發展

	1985	1995	2000	2005	2010	2016~
	電腦世代	網路世代	行動世代	雲端世代	IoT 世代	
主要應用/模式	資訊系統管理、系統整合等	電子商務、入口網站網路搜尋、線上影音、Open API 等	社群媒體、行動App、行動影音、Open API、分享等	SaaS、PaaS、IaaS 等雲端服務、巨量資料分析服務	虛擬整合/跨業創新應用、智慧化服務、智慧工廠等	
主要產品	大型主機、桌上型電腦、應用軟體等	桌上/筆記型電腦連網裝置等	智慧型行動電話、平板電腦等	資料中心設備、資料分析工具等	感測器、智慧穿戴、機器人、無人車等	
產業典範轉移	出版、媒體、影視…	旅遊、零售銷售、行銷、廣告…	3C 硬體、行銷、廣告、住宿…	金融、資服、3C 硬體…	交通運輸、生產製造、居家生活、醫療照護、城市建設	
重大改變	Wintel 架構	網路服務興起 消費習慣改變	行動優先、軟體使用地點改變	軟體銷售方式、軟體開發方式改變	產品即服務、軟硬整合、異業整合、世代整合、實體與網路整合	

世界首富正在影響我們的未來

二大世界首富，與未來的首富



傑夫·貝佐斯 (Jeff Bezos)

Amazon



伊隆·馬斯克 (Elon Musk)

Tesla、SpaceX、無聊公司、
SolarCity、Neuralink



中本聰 (Satoshi Nakamoto)

比特幣之父

馬斯克建構的未來世界

Tesla : 特斯拉汽車



SpaceX：太空探索技術公司



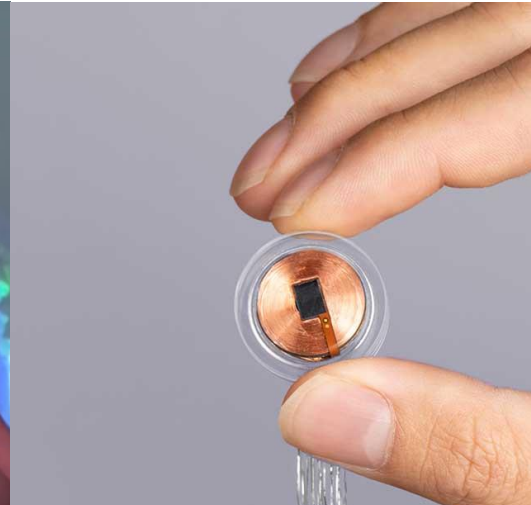
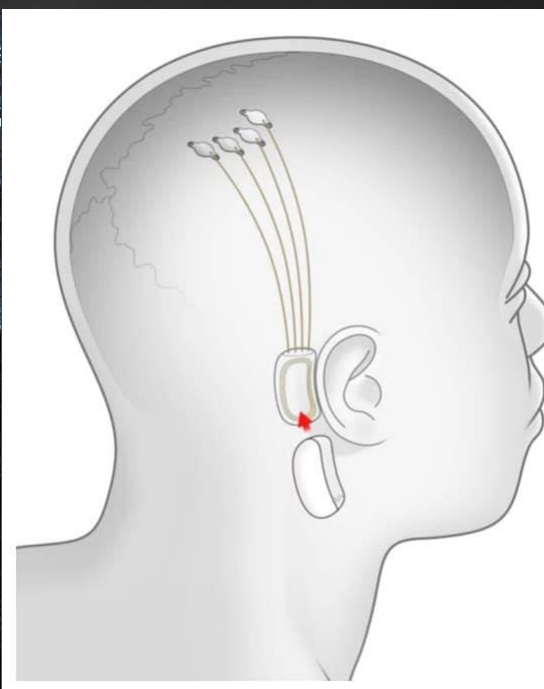
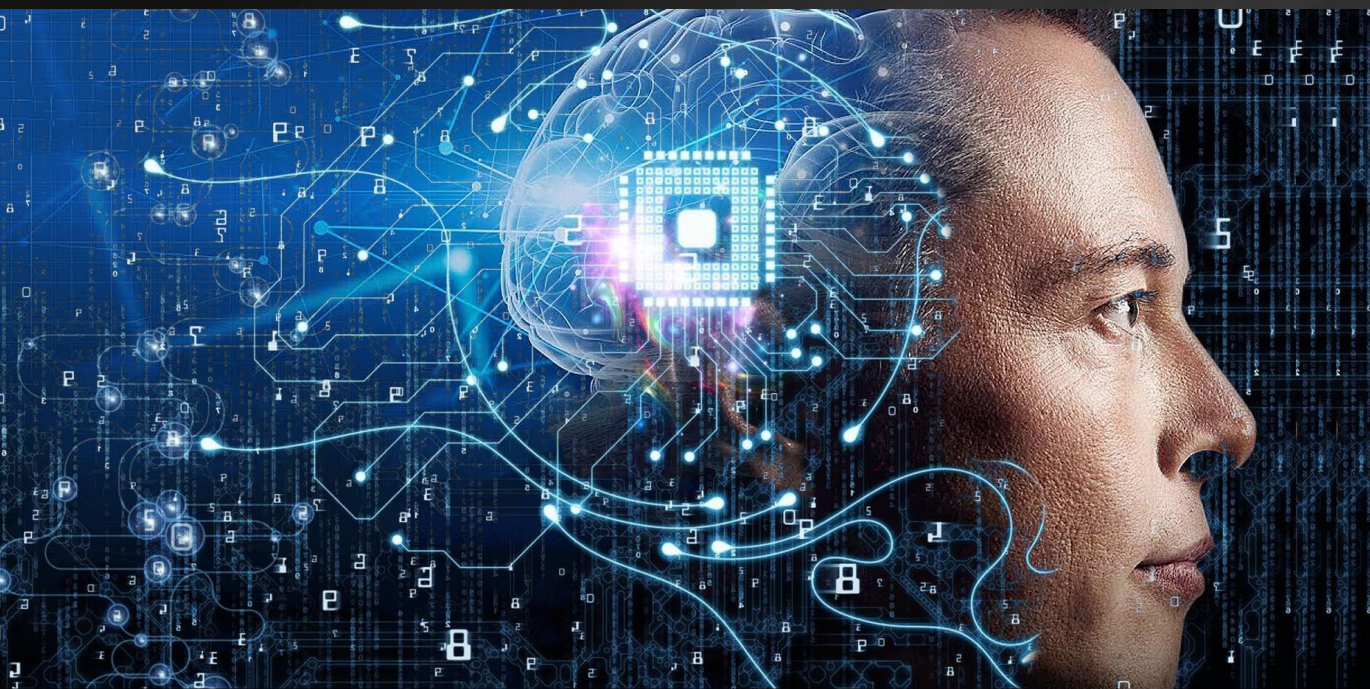
Boring Company：基礎設施和隧道建設公司



SolarCity：全美最大的太陽能發電公司



Neuralink：神經科技和腦機接口公司



跨領域的創新

我的親戚實例

2014 紅點獎：Hear me 助視障者輕鬆錄音

- 傳達設計獎最佳獎。全台第一個獲紅點獎的手機 App。
 - ✓ 視障者多以錄音的方式記錄生活細節，手機的錄音設計未考慮視障者。
 - ✓ 適合盲人使用的 App「聽，見」，用語音開啟軟體，敲擊手機螢幕，即可暫停或中止錄音，更能以滑動方式標記，方便回頭重聽時快速找到重點。
 - ✓ 結合地理資訊定位系統，使用者可錄製生活訊息或個人心情，上傳分享。



台科大工商業設計研究所學生學生林奕岑

善用程式，讓你更突出（國際大廠專案設計師）



台灣動畫界享譽盛名的 FUI 動態圖像設計師 – Yoshiki Lai 賴志彥

工商業的自動化

AccuPick 3D：機器學習實現工業 4.0



所羅門 AccuPick 3D 智能取放系統，具備辨識與分類能力

便利商店：人工智慧、物聯網實現無人商店



Fami News

Category	Value
個人	36
蝦皮	50
奇摩	26
九易	13

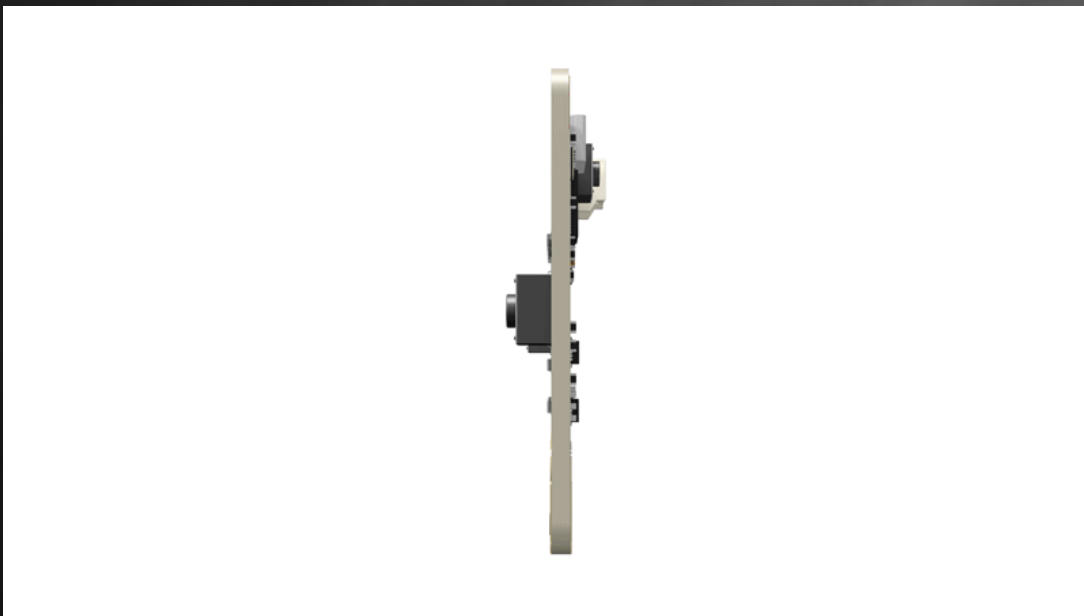
這是RFID技術搭配物流端的標籤



主流的創客微型開發裝置

主流的單晶片：micro:bit (適合國小生)

- micro:bit 是一塊沒有外殼的開發板。由英國廣播公司 (BBC) 設計用於英國的青少年程式教育。具備以下特點：
 1. 體積小、耗電低、便宜，主控板市價約 450~550 元，配件也很便宜。
 2. 主控板基本功能完整，可額外結合許多硬體，創造更多樂趣。
 3. 能夠使用積木式程式 (Blocks)、JavaScript 或 MicroPython 編寫。



主流的單晶片：Arduino（適合小四 ~ 玩家）

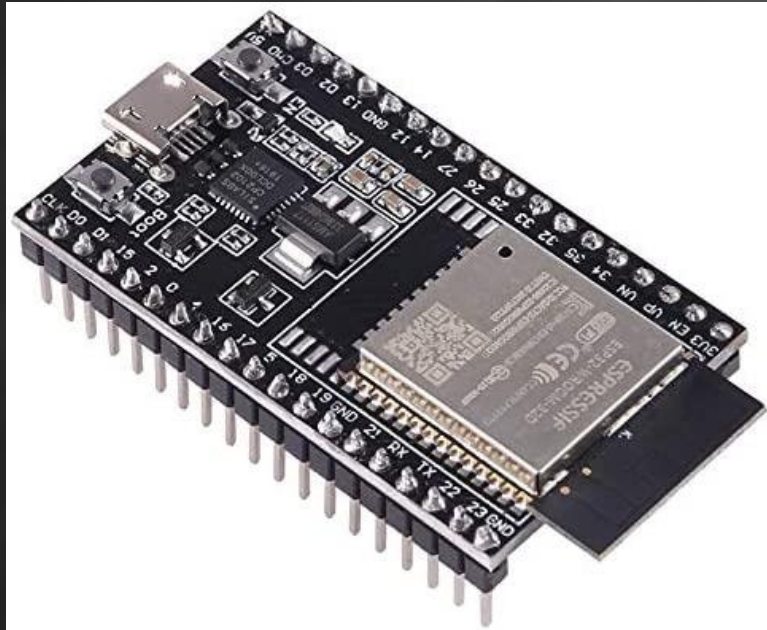
- Arduino 是一家製作開源硬體和開源軟體的公司，該公司負責設計和製造單板微控制器和微控制器套件，用於構建數位裝置和互動式物件。



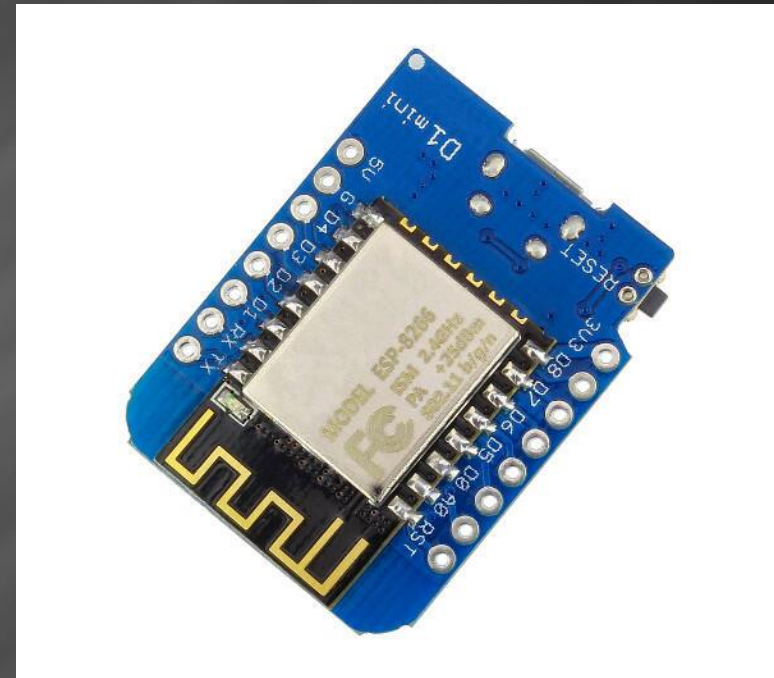
Arduino Uno SMD R3

主流的單晶片：ESP 系列（適合小四 ~ 玩家）

- ESP 系列由上海樂鑫信息科技所開發，基於這個 Wi-Fi IoT 晶片發展出的開發套件系列，這一、兩年紅透半邊天，甚至給其他通訊晶片大廠很大的壓力。



ESP32



ESP8266 (D1 mini)

主流的單晶片：樹莓派（適合專業玩家）

- 樹莓派（Raspberry Pi），簡稱 Pi，是基於 Linux 的單板電腦，由英國樹莓派基金會開發。目的是以低價的硬體，及自由軟體促進學校的電腦科學教育，使得軟體開發變得非常上手。



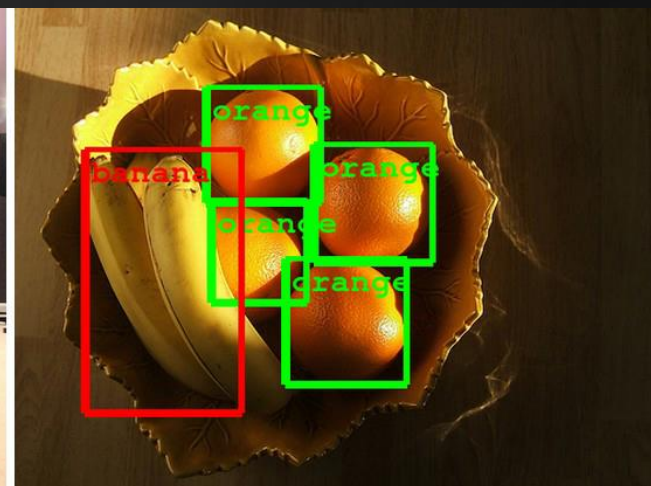
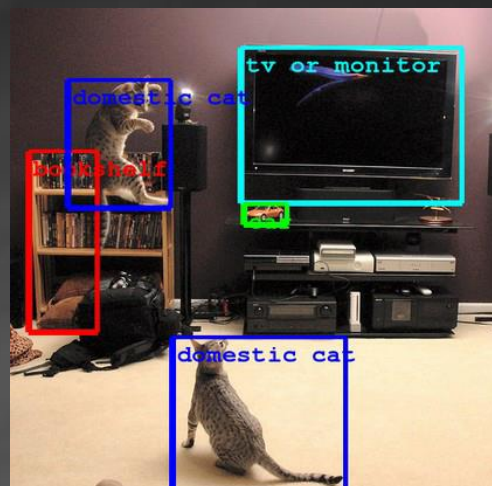
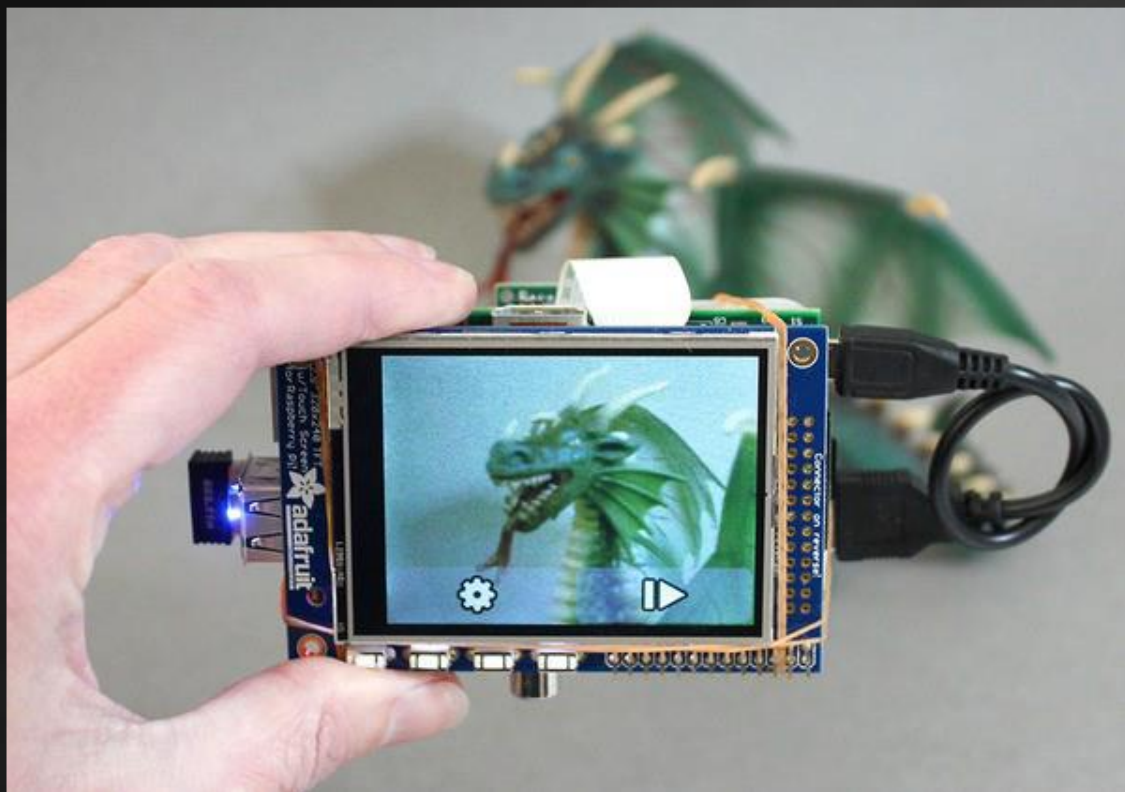
樹莓派的應用

影像處理

IP Camera (網路攝影機)



具有影像辨識的雲端相機



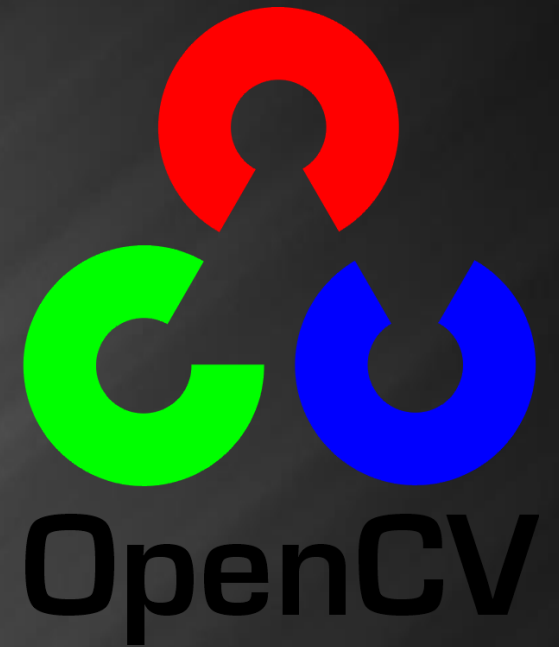
可做影像辨識的相機



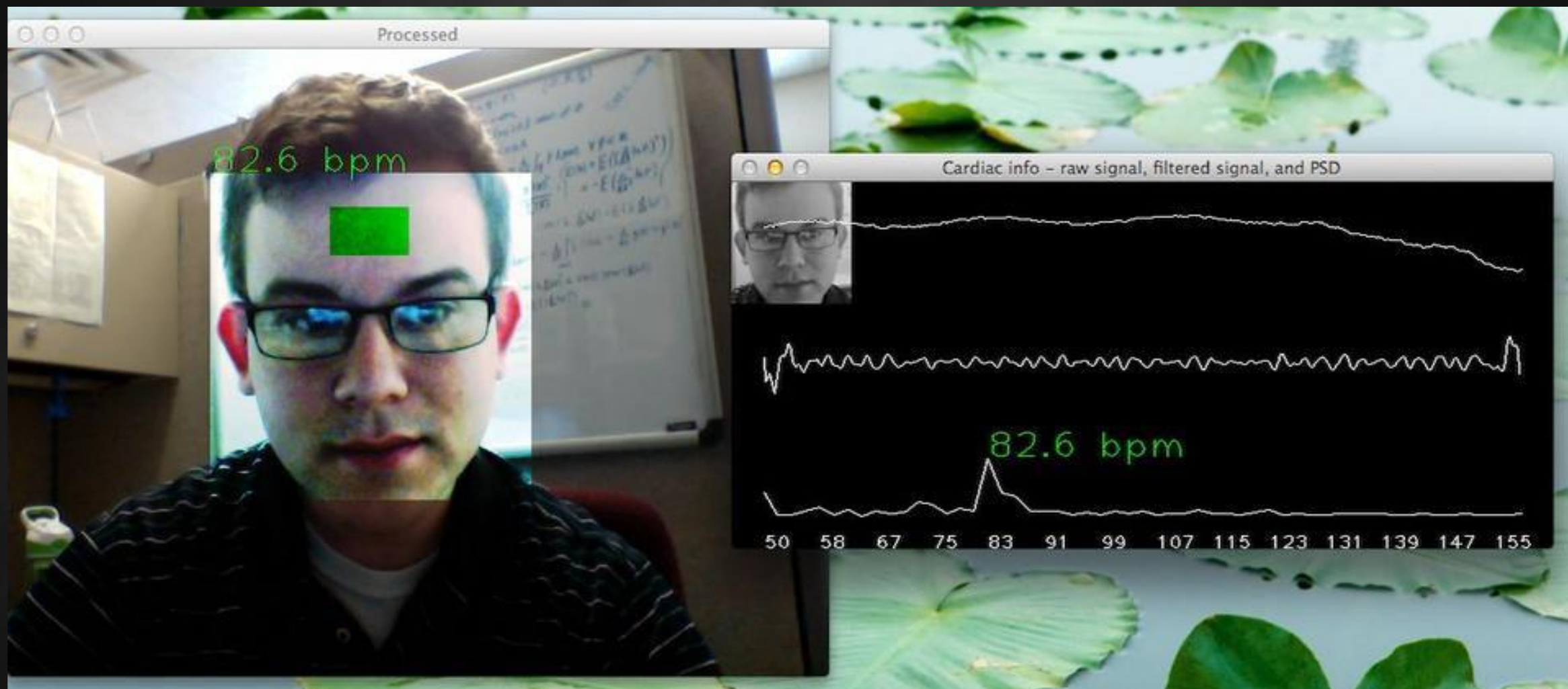
Pi 立得



人臉偵測與追蹤

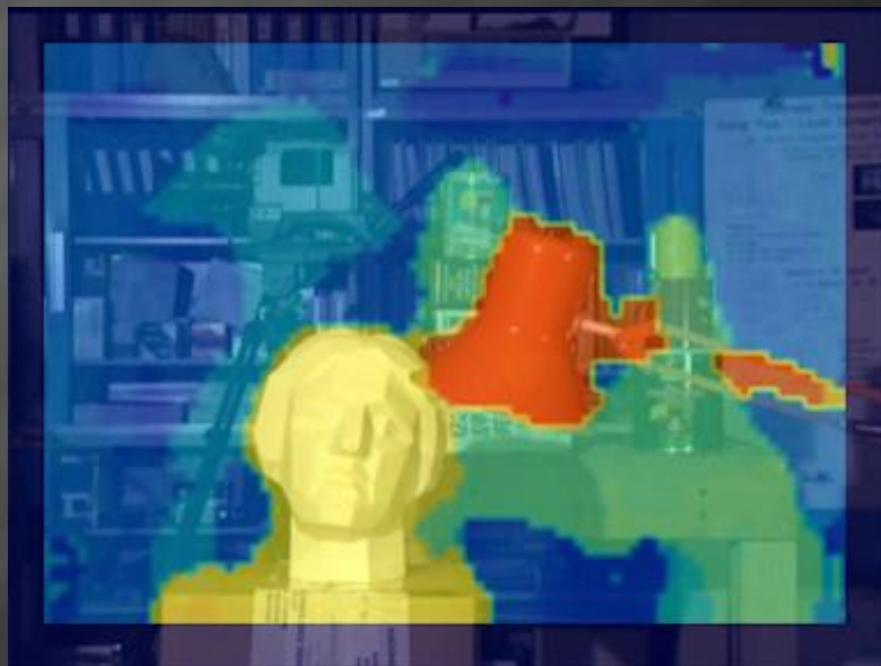
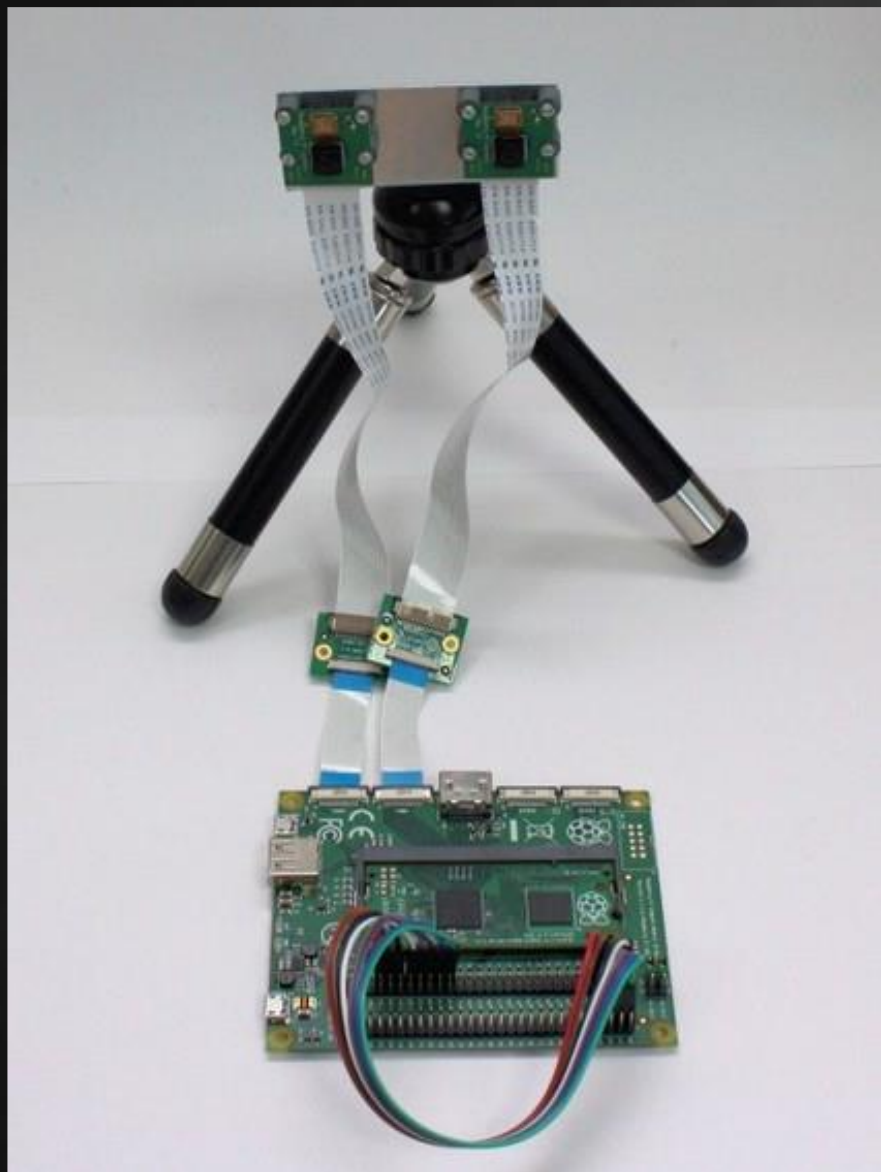


脈博辨識

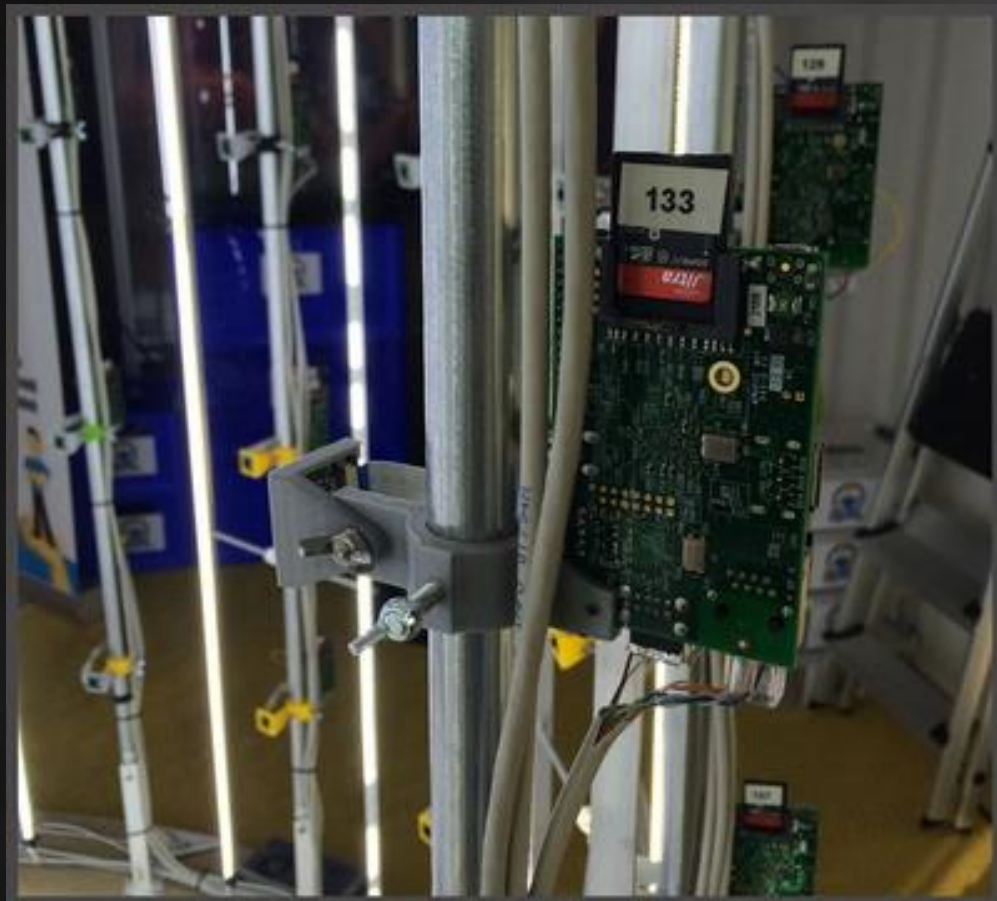


<https://github.com/thearn/webcam-pulse-detector>

3D 建模計算深度 / 2 Cameras



3D 掃描 / 50 Cameras



因為有這些程式
生活更美、更好



Hello, World! 最像英文的 **Python**



Python

```
# 印出 Hello World! 字串物件  
print("Hello World!")
```

C

```
/* 印出 Hello World! 字串物件*/  
include <stdio.h>  
  
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```

C++

```
//印出 Hello World! 字串物件  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    cout << "Hello World" << endl;  
    return 0;  
}
```

Java

```
//印出 Hello World! 字串物件  
public class HelloWorld{  
  
    public static void main(String []args){  
        System.out.println("Hello World");  
    }  
}
```

程式語言：五大語法結構



Sequence

循序執行



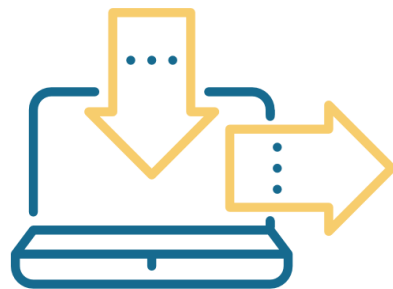
Conditional
Statements

if 判斷式



Loops

迴圈



Input / Output

輸入/輸出



Functions

函數

Outline

- 美好的程式世界
- 智慧物聯網 (The Artificial Internet of Things, AIoT)
- 9 個小時的課程：AI 聲音姿態影像辨識大應用

何謂 AIoT ?

Artificial Intelligence



If...then...

learn...

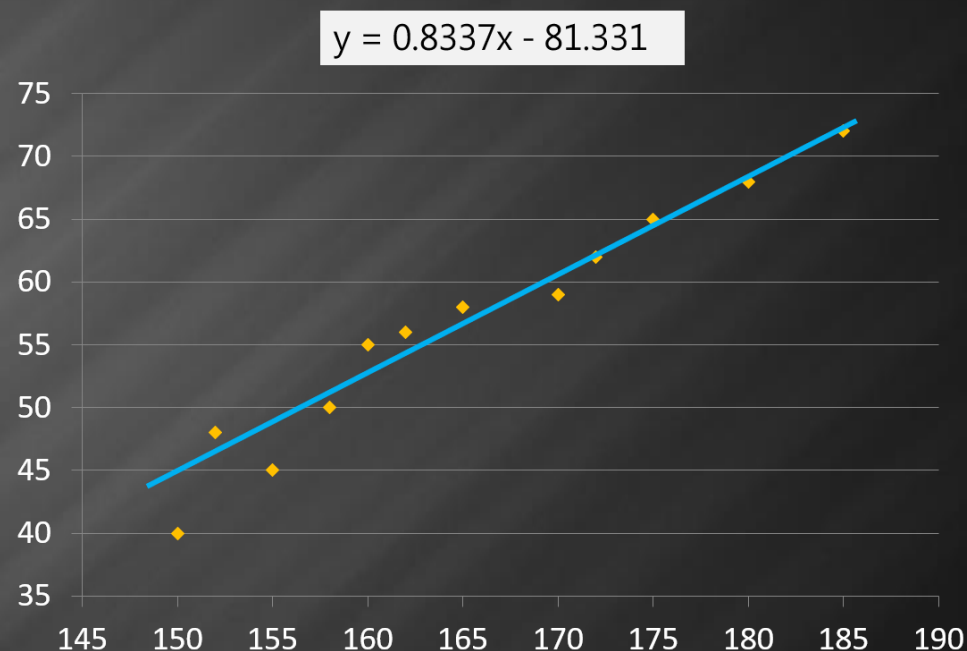
AI 的主流技術：機器學習

- 機器學習，即準備一些問題與對應的答案給電腦後，讓它自行找出其中的規則，並且有能力針對類似的問題給出正確的答案。
 - ✓ 歸功於網路龐大的資訊量，或各式感測器收集的資料。

某一班學生的身高和體重是否相關？

能否用身高來推測某位學生的體重？

身高	體重
150	40
152	48
155	45
158	50
160	55
162	56
165	58
170	59
172	62
175	65
180	68
185	72



人工智慧發展簡史

第一波

1950-1960

符號邏輯

把人的**思考邏輯**放進電腦

由領域專家寫下決策邏輯。

人類還沒辦法清楚理解自己的思考過程，如何告訴電腦？

失敗



第二波

1980-1990

專家系統

把人的**所有知識**放進電腦

由領域專家寫下經驗規則。

太多難題人類無法解答、無法寫成規則、無法以程式碼表示。

失敗



第三波

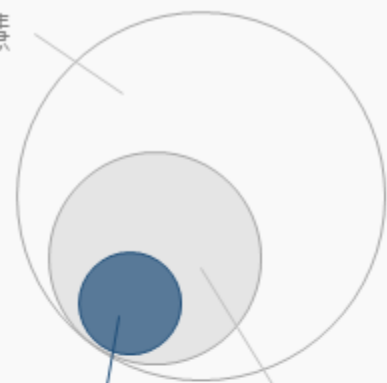
2010-Present

機器學習

把人的**所有看見**放進電腦

由領域專家提供歷史資料，讓電腦自己歸納規則。

人工智慧



機器學習
(第三波人工智慧的代表技術)

深度學習
(機器學習技術中成長最快、表現最佳)

專家系統

專家定義規則

傳統機器學習

(與深度學習區隔)

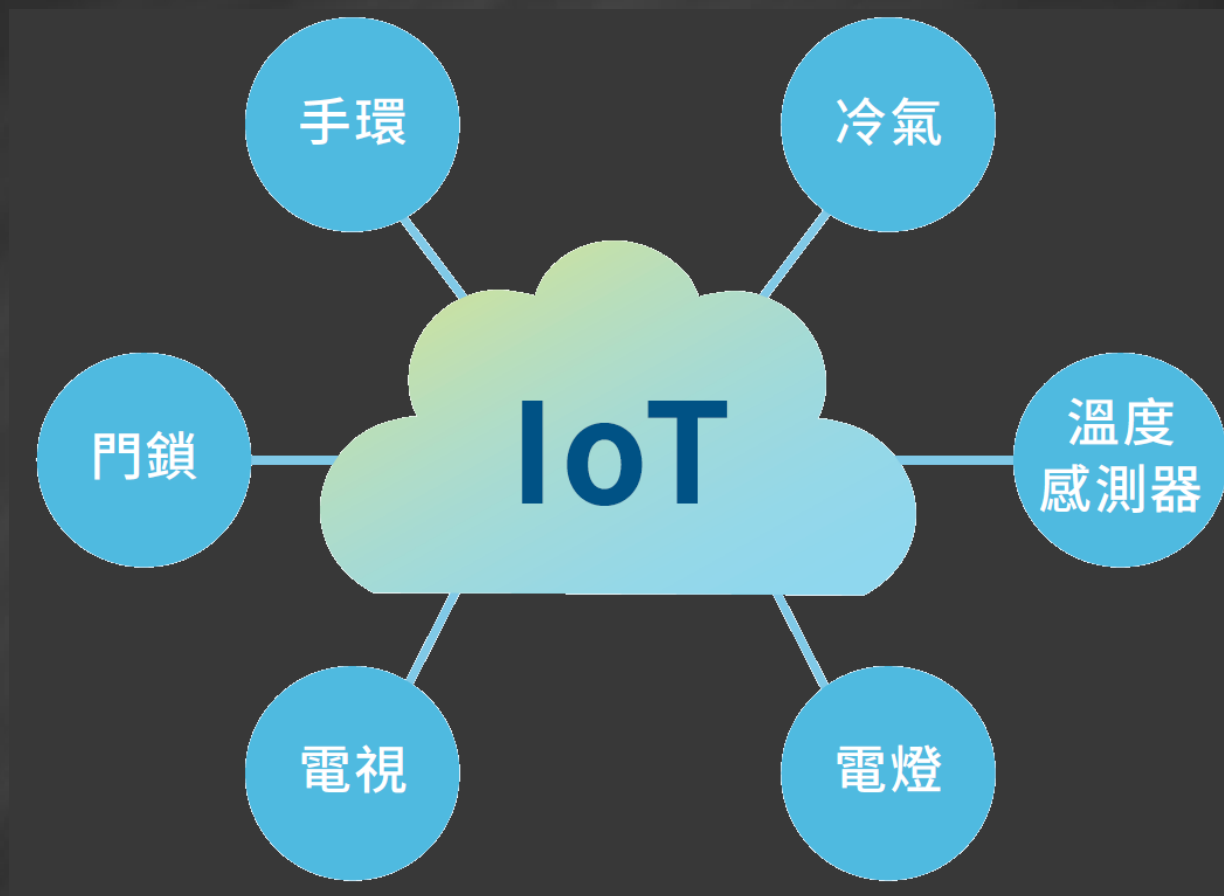
電腦定義規則
專家定義特徵

深度學習

(多層類神經網路)

電腦定義規則 (更準)
電腦定義特徵

何謂 AIoT ?



物聯網應用涵蓋多元

公共應用

裝置數量：
1,000,000~100,000,000



企業應用

裝置數量：
100~1,000,000



企業應用：

1. 智慧製造：製造、營造
2. 智慧醫療：醫療院所、健檢中心、居家健康照護、健康社區、遠距醫療
3. 智慧建築：一般家用住宅、企業建築
4. 智慧物流：便利商店、批發與零售業

家庭應用

裝置數量：10~100



家庭應用：

智慧家庭：室內照明，安全監控，空調，智慧居家，智能廚房、溫度控制、濕度控制、節能控制、影音播放

個人應用

裝置數量：1~10



個人應用：

智慧裝置：隨身助理、生理量測、行人導航…

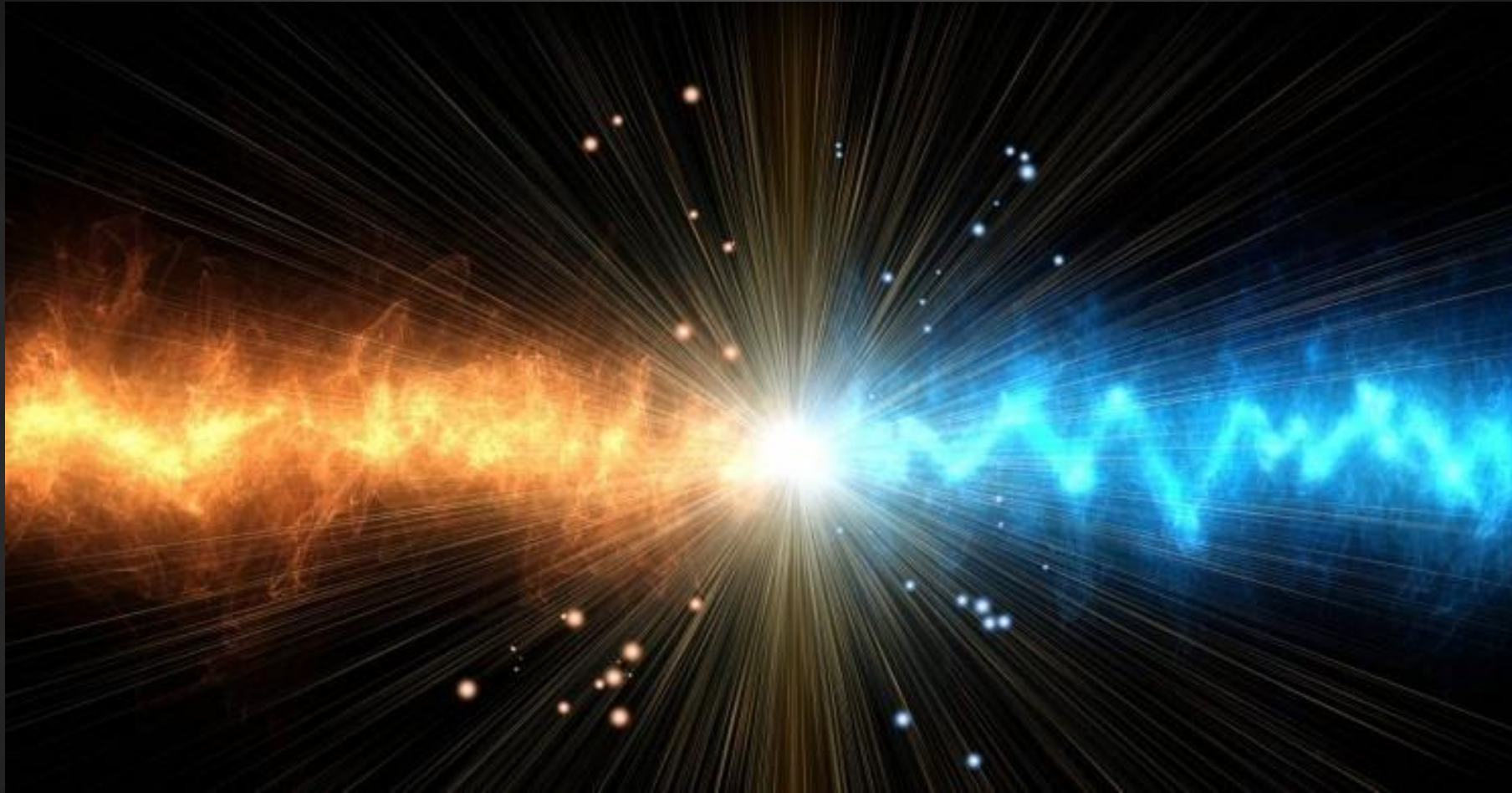
政府應用：

1. 智慧交通：公共運輸（計程車、大客車）、車隊管理、交通資訊、智慧交通控制
2. 智慧能源：一般家庭、企業、工廠
3. 智慧安全：保全、防災橋樑監控、水質監測、水位監控、地震偵測、颱風預測
4. 智慧農業：農產品產銷履歷、農漁、畜牧、病蟲害防疫、溫室環控
5. 智慧教育：擴增實境

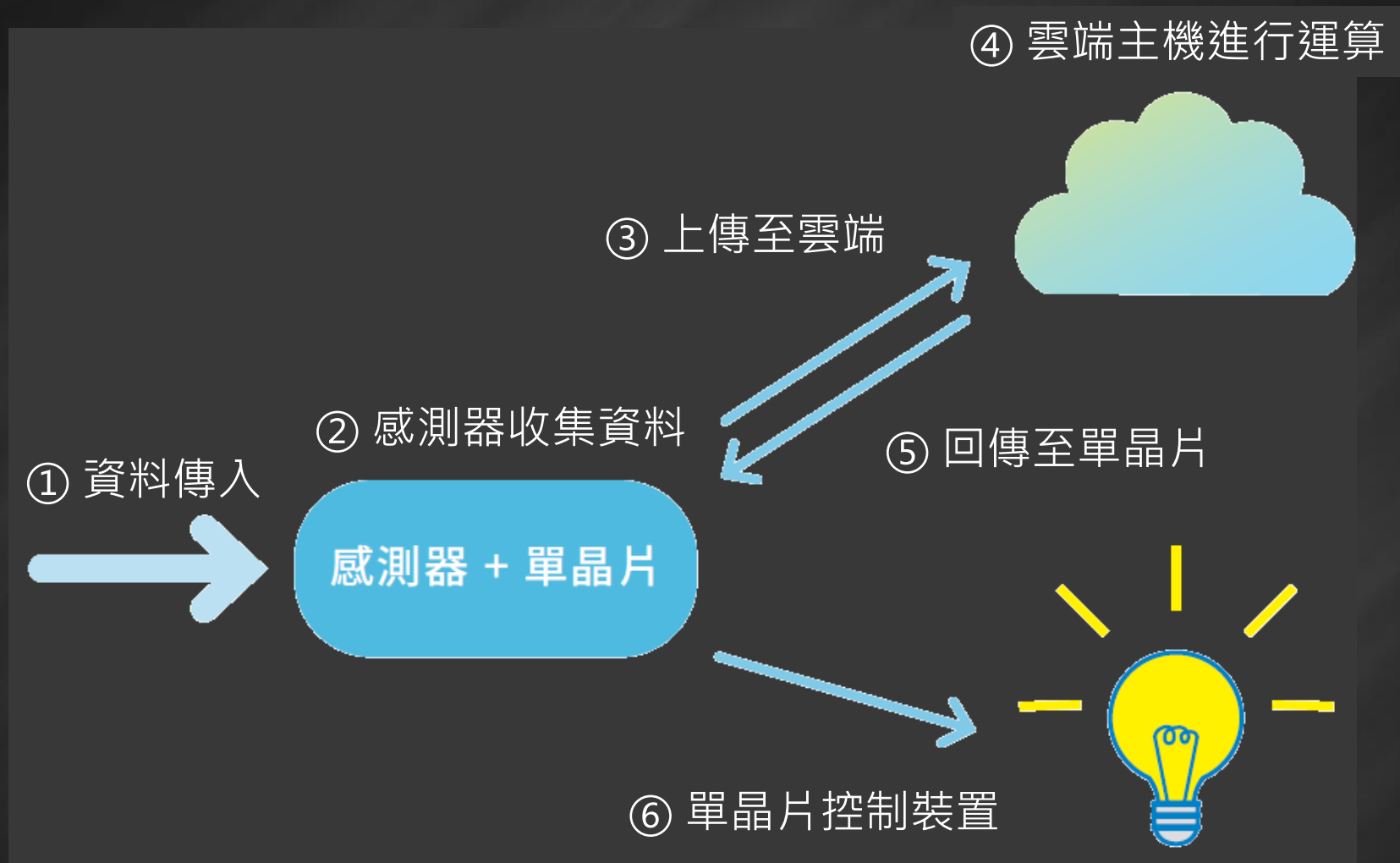
物聯網會不會泡沫？

- 物聯網發展不如早期那麼興奮，在《**The internet of things myth**》一書中，作者 10 年前預估物聯網產業會非常興奮，預估到 2020 年全球互聯網裝置將達 500 億個。時至今日，專家估計今年只有 110 億個，也就是只達到當初預期 22%，遠不如預期。
- 但時間遲早會來到，只是會延後若干年，產業發展將類似 2000 年網路泡沫，不合理的樂觀會變成泡沫，但如今網路影響生活是事實。
- IoT 也可能走相似的路徑，神話很大的概率，最終將實現。

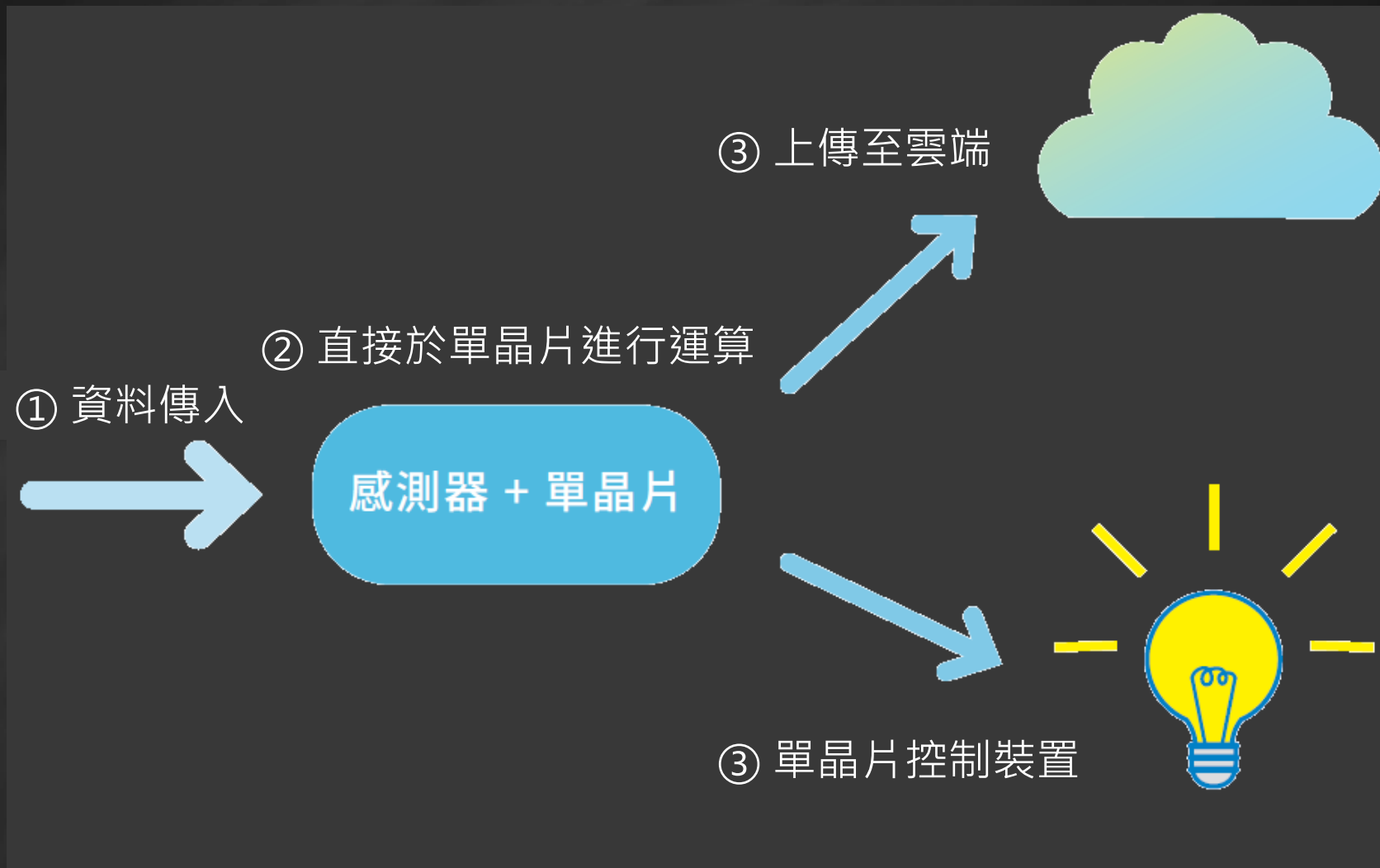
何謂 AIoT ?



雲端運算



邊緣運算



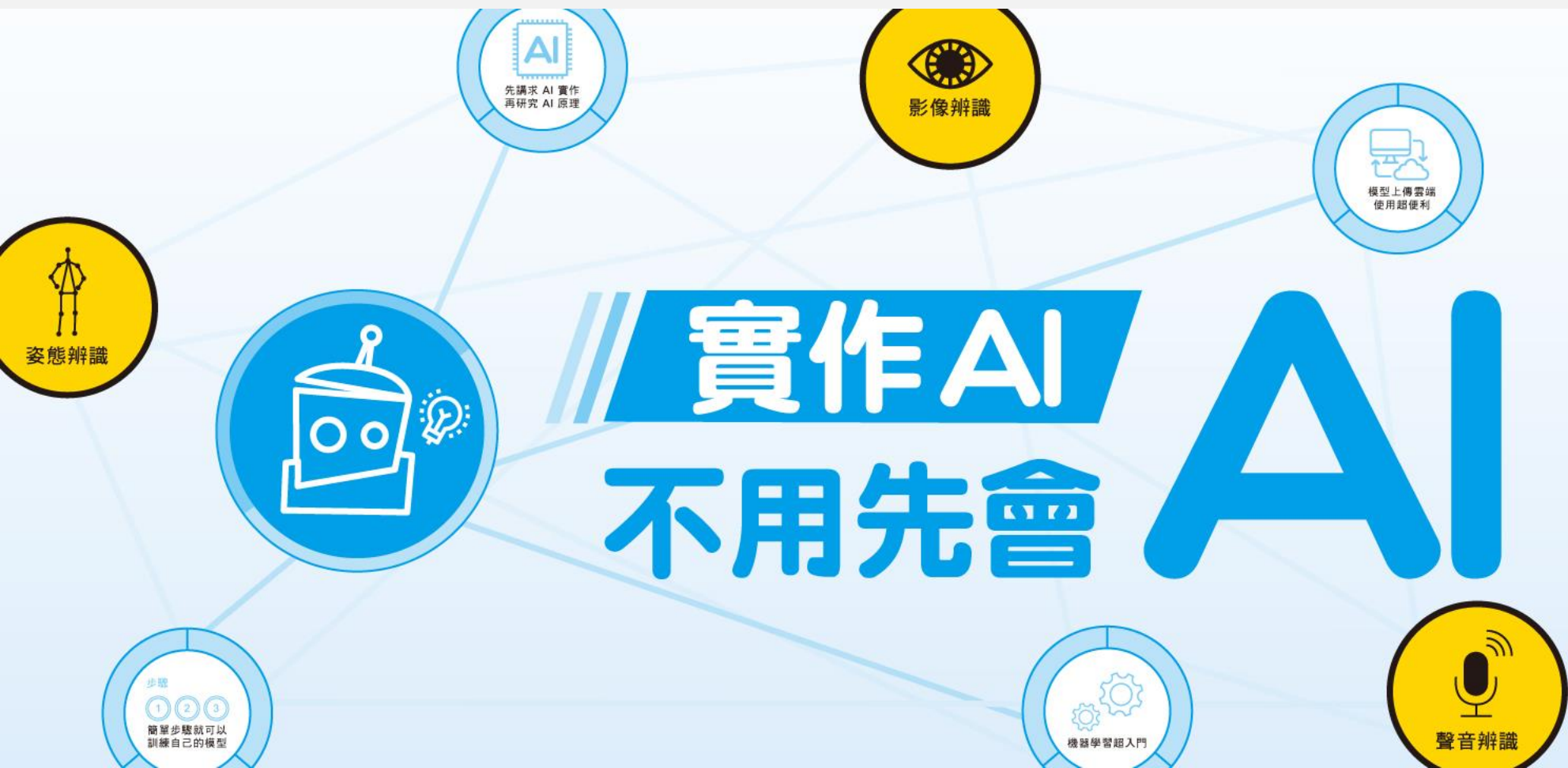
Outline

- 美好的程式世界
- 智慧物聯網 (The Artificial Internet of Things, IoT)
- 9 個小時的課程：AI 聲音姿態影像辨識大應用

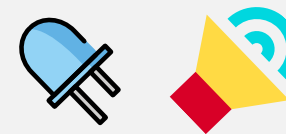
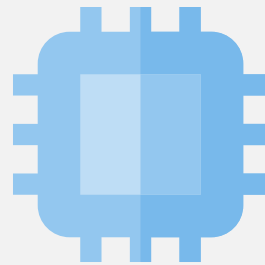
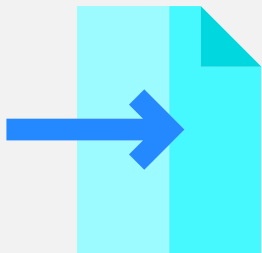
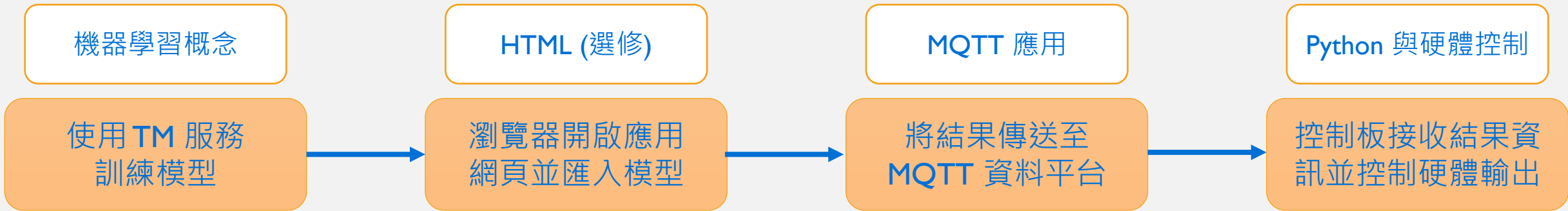
課程大綱

- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用

今天要用到的套件



本課程的學習流程



課程大綱

- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



00 實驗環境設定

視訊鏡頭、開發環境與驅動程式

課程必備視訊鏡頭 (WEBCAM) 、麥克風

<https://bit.ly/3yEVgF9>



安裝第三方軟體, 讓手機成為電腦鏡頭

Google

droidcam



全部 影片 圖片 新聞 購物 更多

工具

約有 690,000 項結果 (搜尋時間 : 0.31 秒)

<https://www.dev47apps.com> 翻譯這個網頁

Dev47Apps

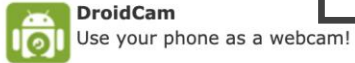


DroidCam turns your phone/tablet into a webcam for your PC. Use it with chat programs like Zoom, MS Teams, and Skype. Main Features: – Chat using “DroidCam ...

<https://www.kocpc.com.tw> Android App分享介紹

Dev47Apps

安裝電腦端軟體



DroidCam turns your phone/tablet into a webcam for your PC. Use it with chat programs like Zoom, MS Teams, and Skype.

Main Features:

- Chat using "DroidCam Webcam" on your computer, including Sound and
- Connect over WiFi or USB cable.
- Unlimited free usage at standard definition.
- Keep using your phone with DroidCam in background.
- Simple, safe, efficient, and trusted by millions.

DroidCamX Pro Features:

- Switch to HD Mode for 720p/1080p high definition.
- Camera controls: toggle light, enable continuous video recording.
- Rotate, Flip & Mirror the video to better match your screen.
- Adjust brightness, contrast, add video delay.
- USB-Only mode for extra privacy and security.
- Option to auto-mute phone calls.
- No Ads.

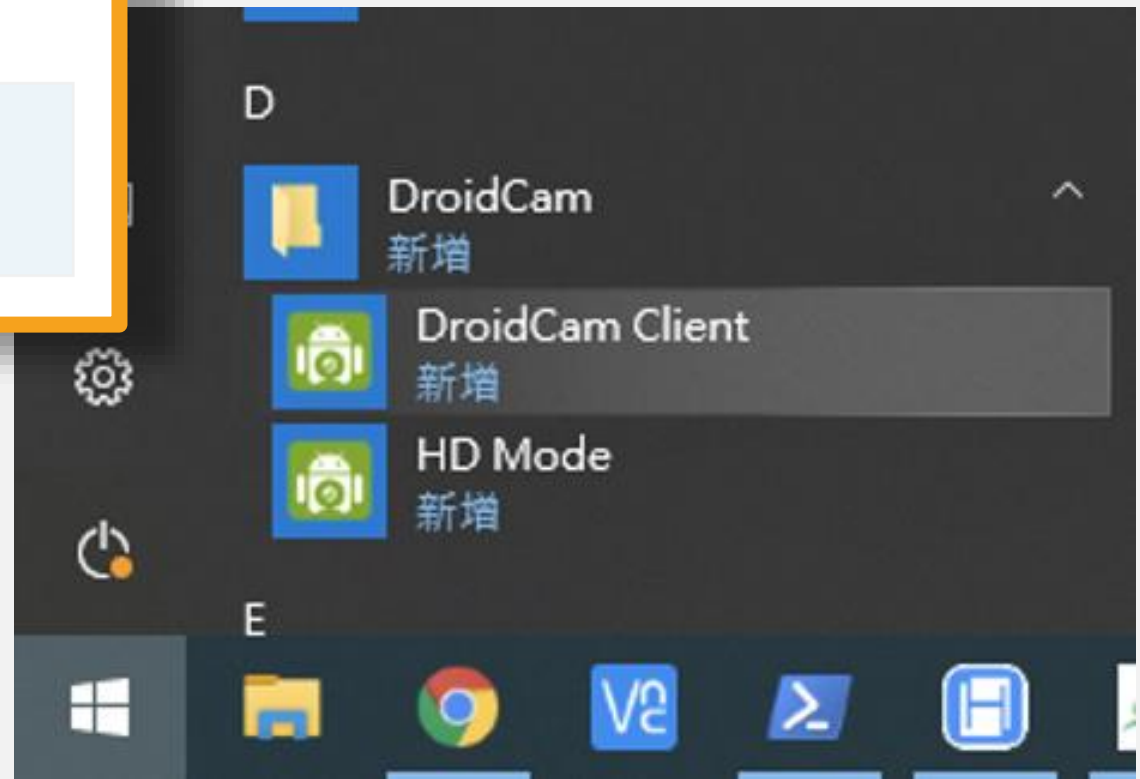
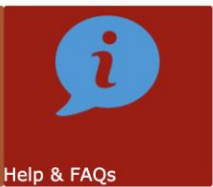


Windows



DroidCam Client

v6.4.3 Download



DroidCam Client

DroidCam

Connect

Connect over USB

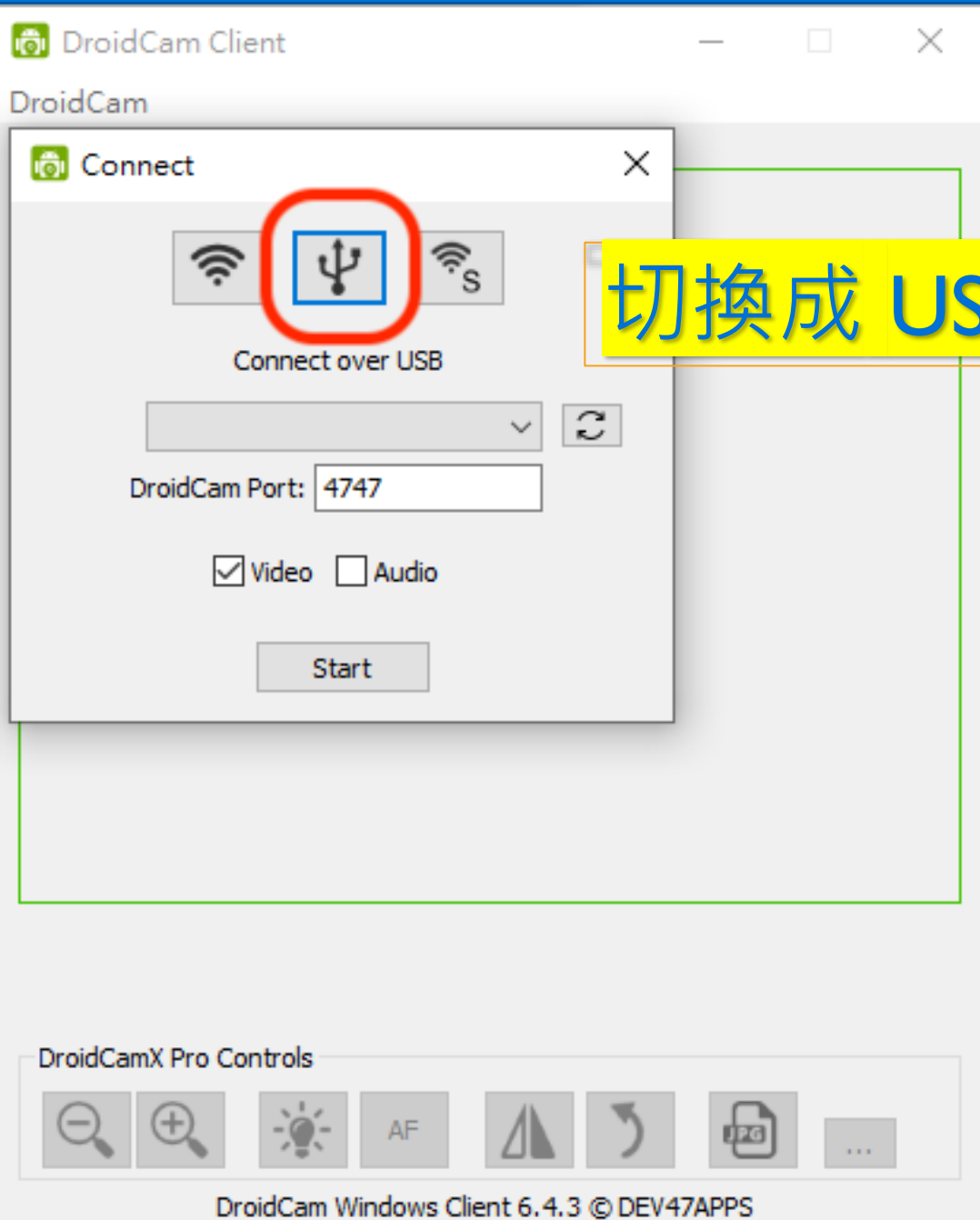
DroidCam Port: 4747

Video Audio

Start

DroidCamX Pro Controls

DroidCam Windows Client 6.4.3 © DEV47APPS

The image shows a screenshot of the DroidCam Client software interface. A 'Connect' dialog box is open, showing three connection options: Wi-Fi, USB, and Wi-Fi S. The USB icon is highlighted with a red circle and a blue square. A yellow callout box with the text '切換成 USB 模式' (Switch to USB mode) is overlaid on the right side of the dialog. Below the dialog, the 'DroidCamX Pro Controls' panel is visible, containing various camera control icons like zoom, light, AF, and a menu. The bottom of the window displays the version information: 'DroidCam Windows Client 6.4.3 © DEV47APPS'.

切換成 USB 模式

安裝手機端軟體

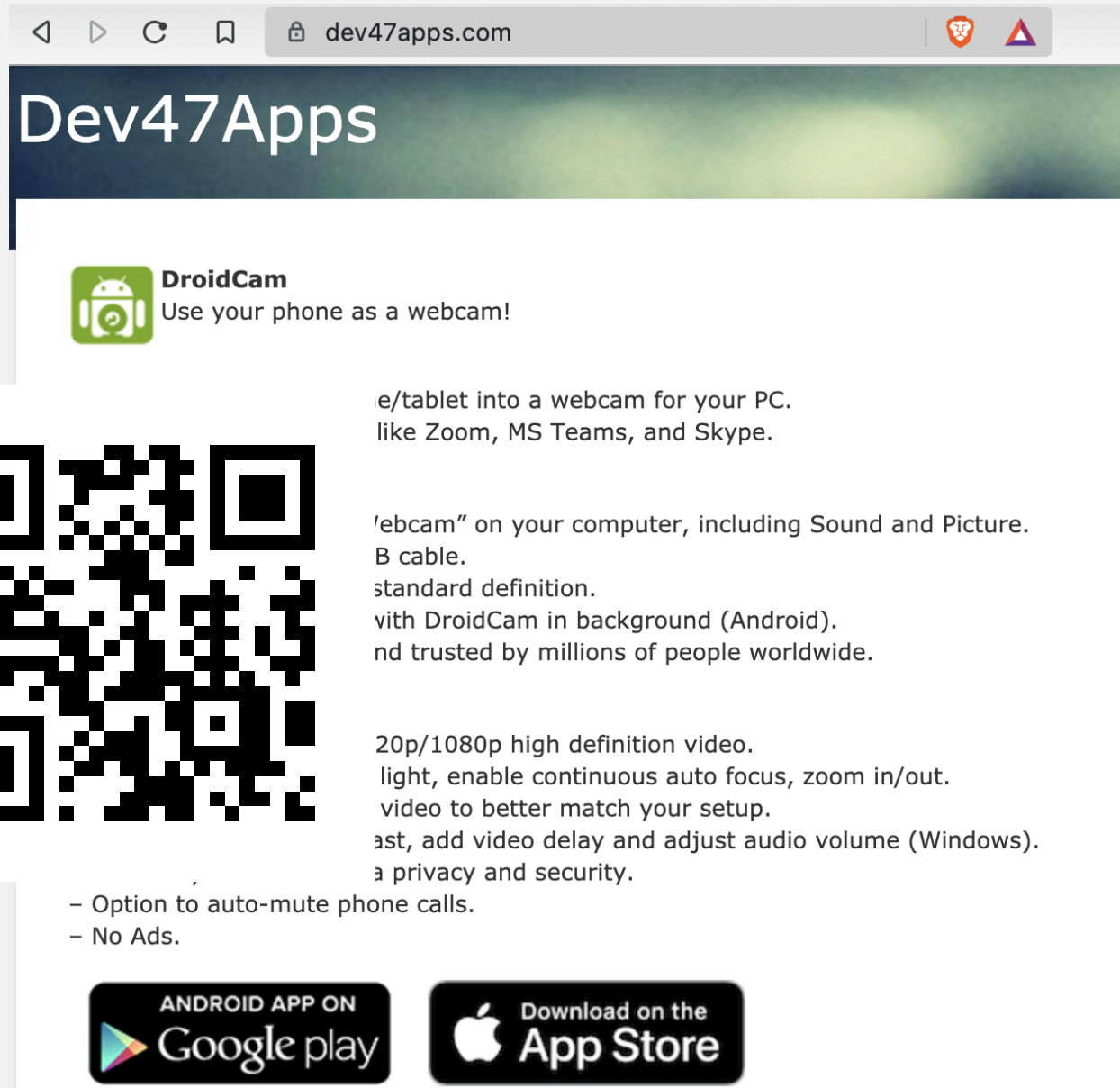
手機連線

<https://www.dev47apps.com/>


或掃描 QR code

選擇自己的手機作業系統

下載並安裝



Dev47Apps


 **DroidCam**
Use your phone as a webcam!


...e/tablet into a webcam for your PC.
like Zoom, MS Teams, and Skype.

...ebcam" on your computer, including Sound and Picture.
B cable.
standard definition.
with DroidCam in background (Android).
nd trusted by millions of people worldwide.

20p/1080p high definition video.
light, enable continuous auto focus, zoom in/out.
video to better match your setup.
ast, add video delay and adjust audio volume (Windows).
a privacy and security.

- Option to auto-mute phone calls.
- No Ads.

ANDROID APP ON
 Google play

Download on the
 App Store

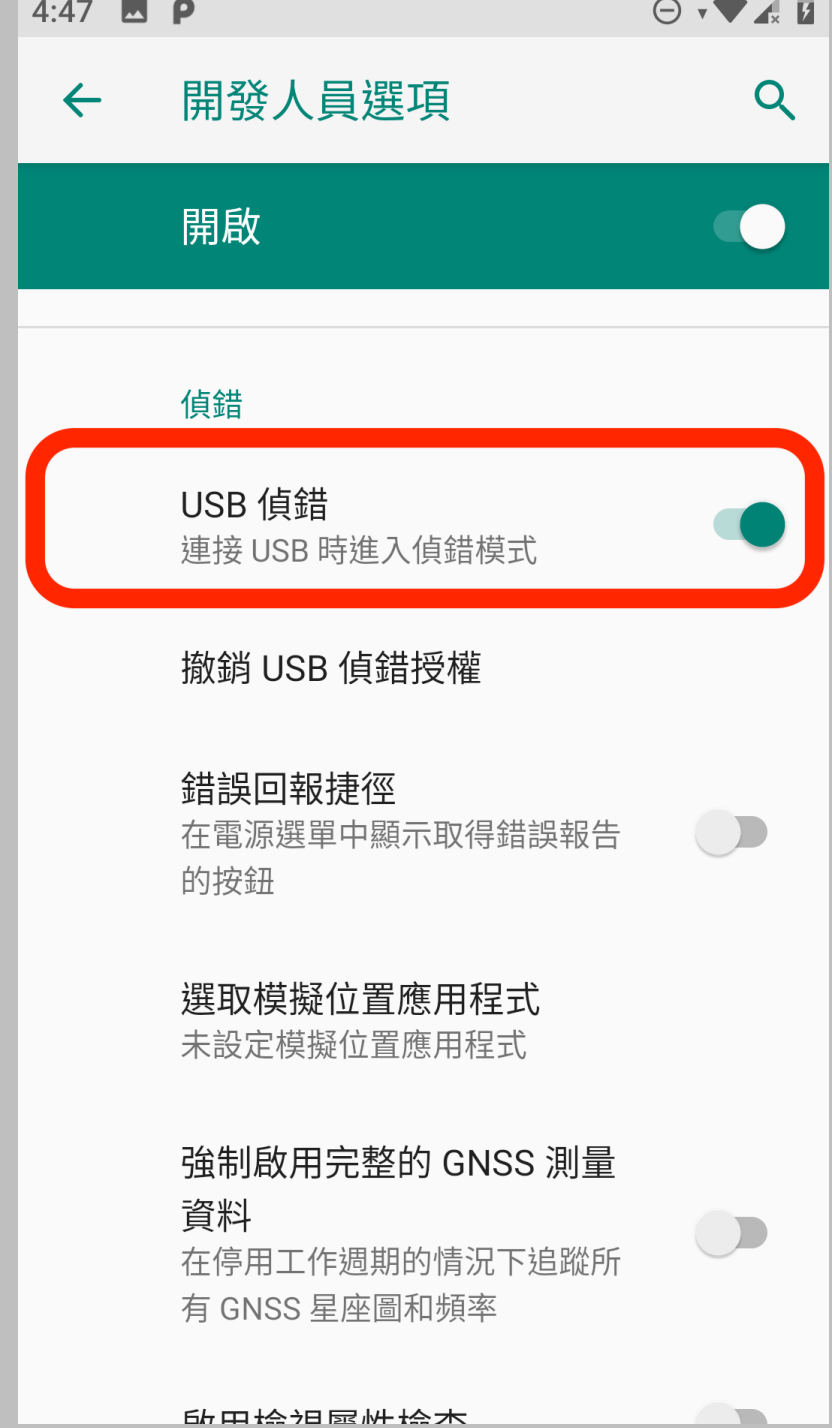
開啟 ANDROID 手機
開發人員模式

『手機設定 / 系統 / 關於手機』



開啟 USB 偵錯

『手機設定 / 系統 / 進階
/ 開發人員選項』



IOS 手機連接電腦需安裝 ITUNES

使用 iOS 手機不需額外設定

但電腦需要額外安裝 iTunes 軟體才能連接

Windows 電腦根據作業系統分別連線下載, 並依指示安裝 :

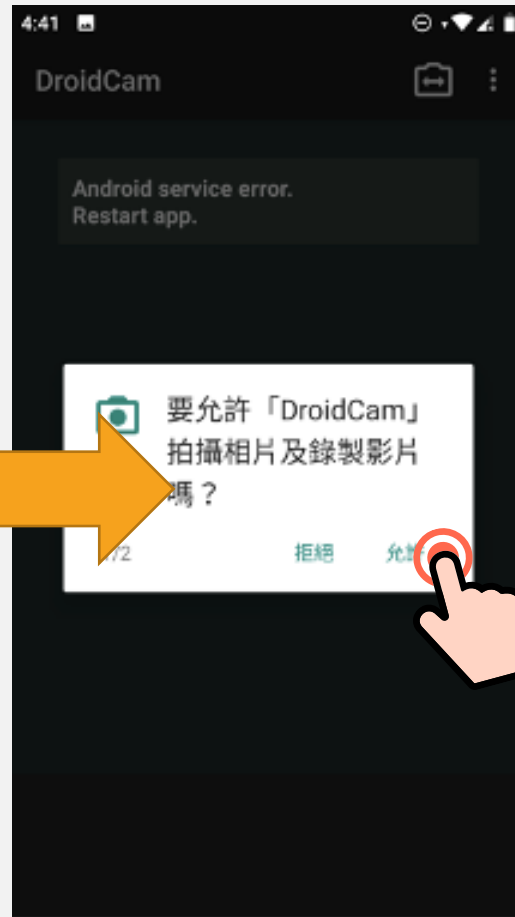
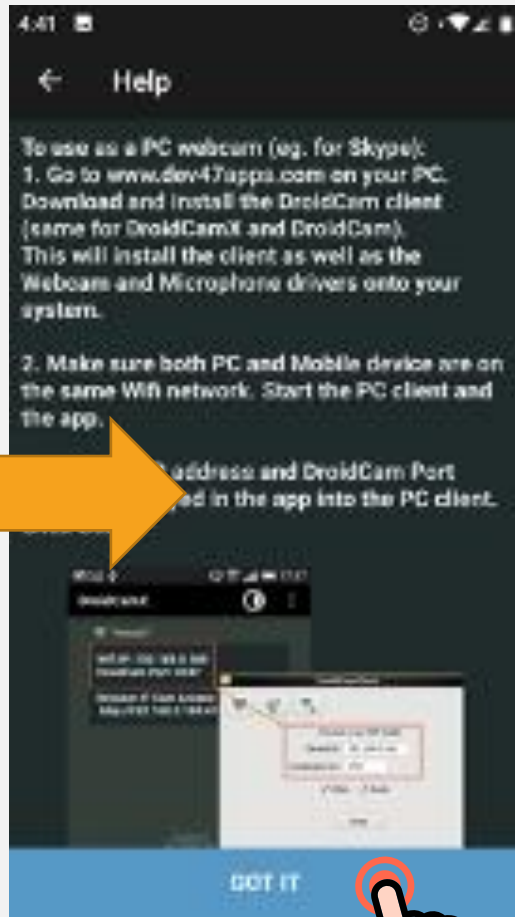
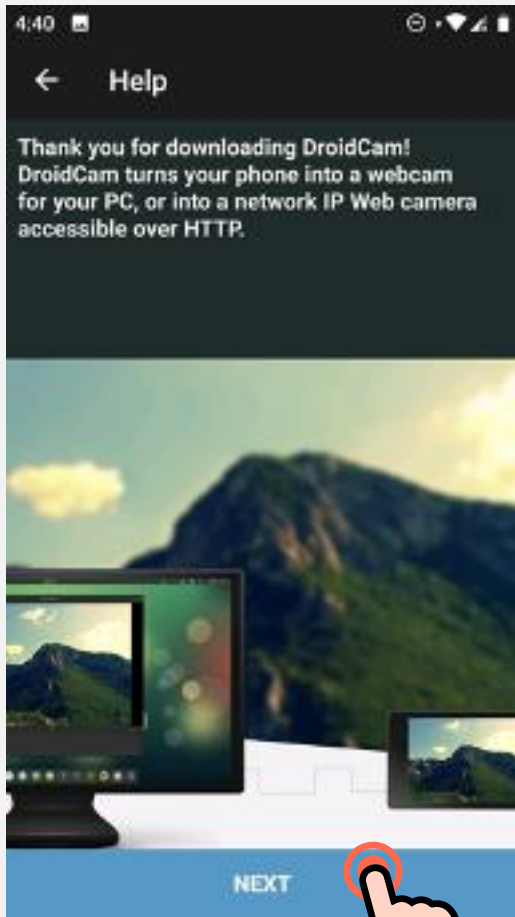
64 位元 :

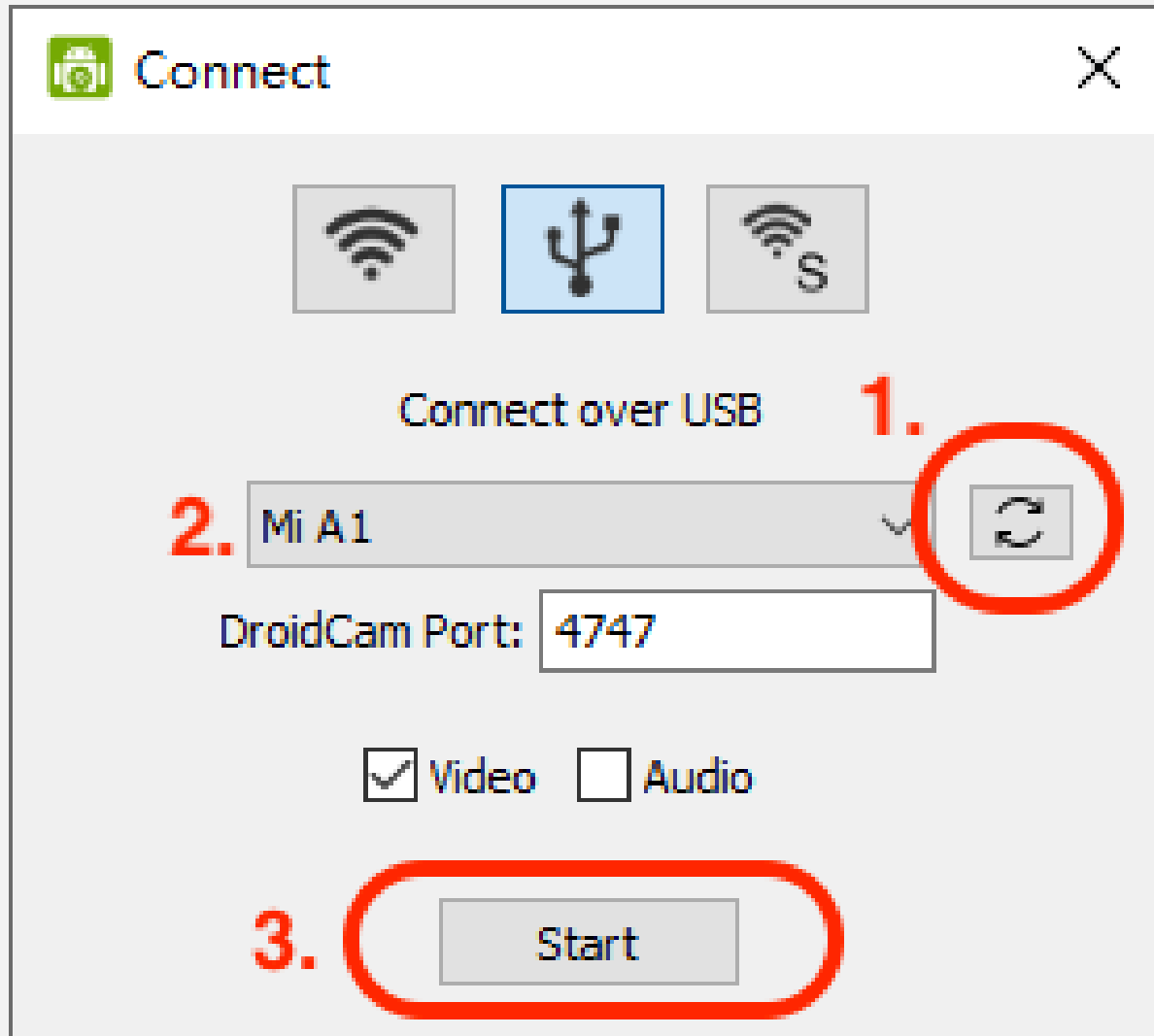
<https://www.apple.com/itunes/download/win64>

32 位元 :

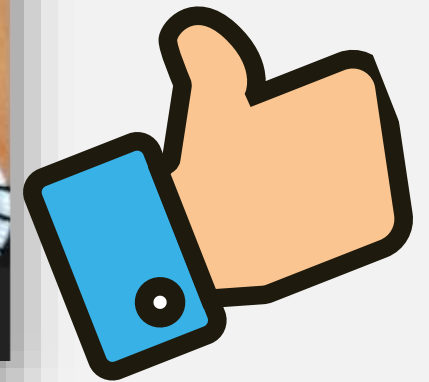
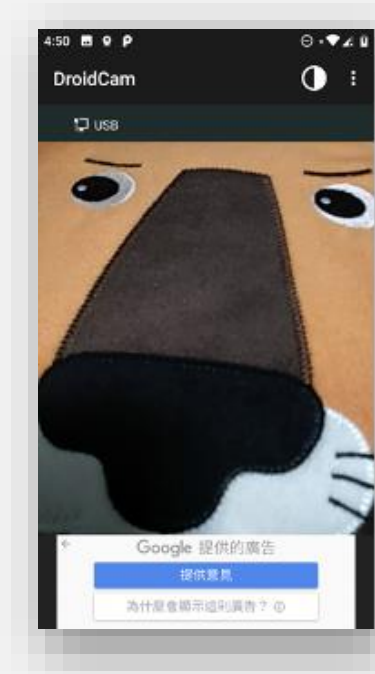
<https://www.apple.com/itunes/download/win32>







1. 按此重新整理
2. 確認裝置列表
3. 按 Start



課程大綱

- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



01 AI 影像辨識

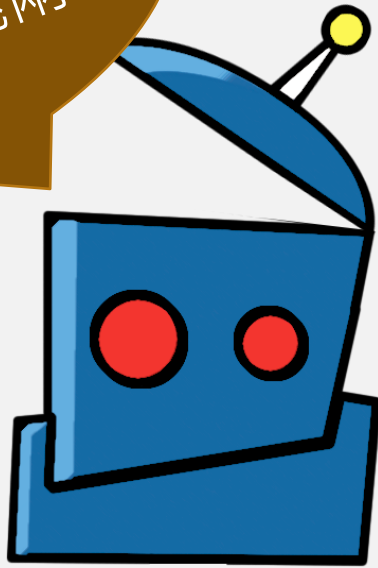
Teachable Machine 建立模型

人工智慧

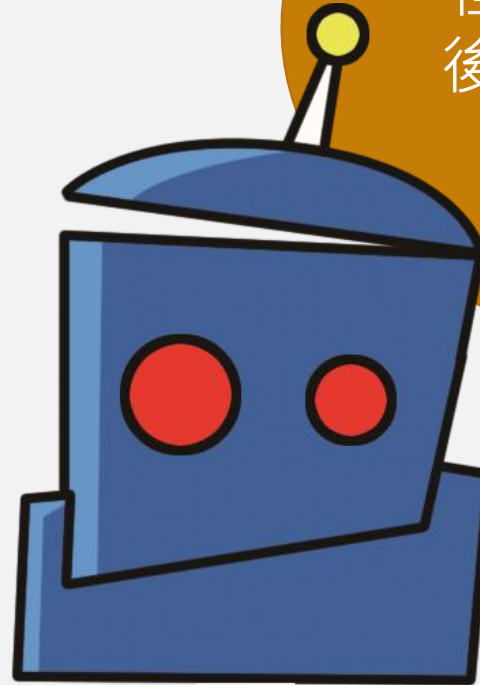
早期的 AI

現在的 AI

如果烏龜在我一個單位的距離出現,我就往前跳兩個單位...



如果烏龜在我前面我可以：
1.踩他 2.跳過去 3.撞他
但是在我玩過了 50 次之後發現,只要不要撞它,我都可以過關。



LAB 手勢影像辨識

使用 Teachable Machine 訓練模型並上傳至雲端

Export your model to use it in projects. ✕

Tensorflow.js ⓘ

Tensorflow ⓘ

Tensorflow Lite ⓘ

Export your model:

Upload (shareable link)

Download

Update my cloud model

Your sharable link:

<https://teachablemachine.withgoogle.com/models/7qkpiwVGK/>

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

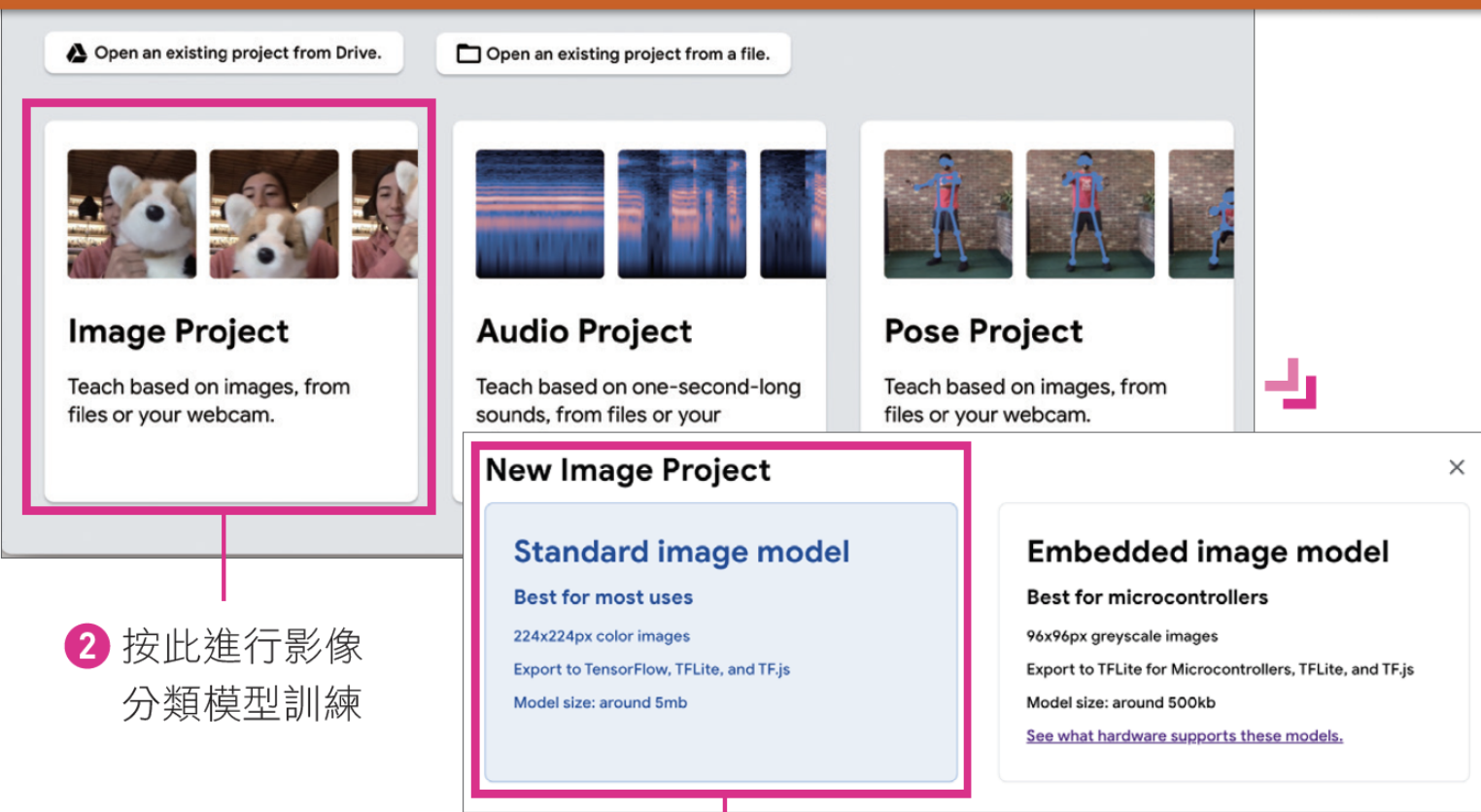
Learn more about how to use the code snippet on [github](#).

```
<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">
  // More API functions here
```

Copy

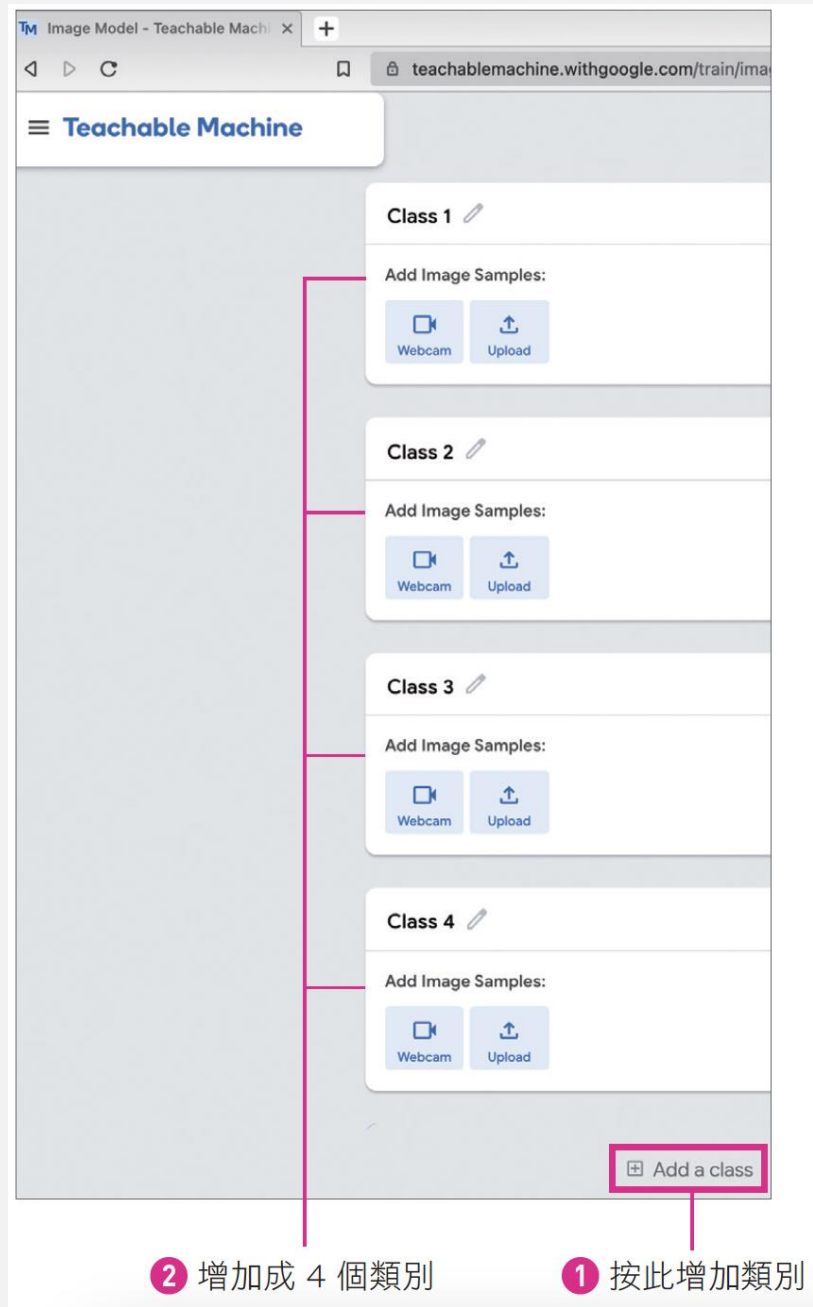
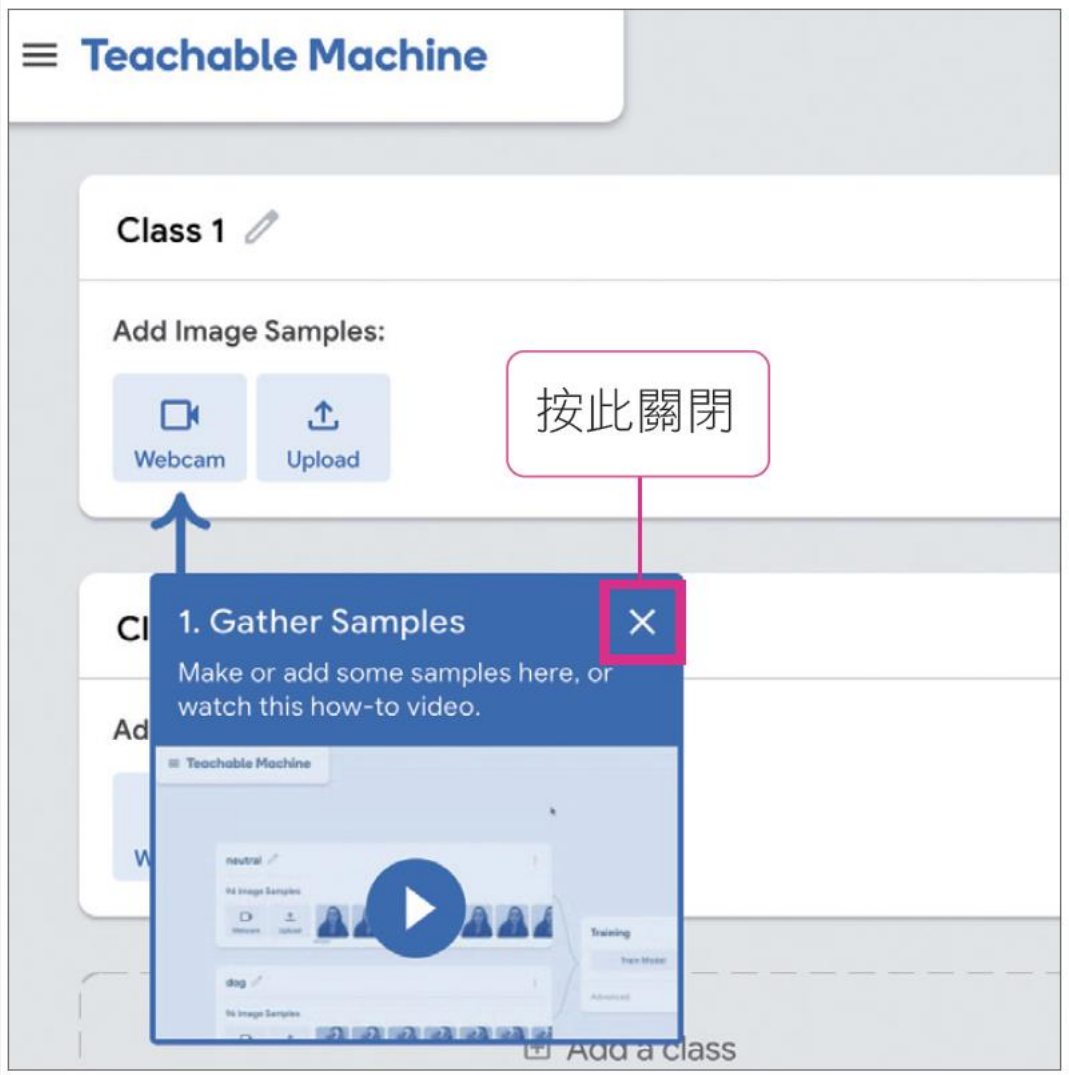


https://teachablemachine.withgoogle.com/train



2 按此進行影像分類模型訓練

3 選擇標準模型




剪刀

石頭

布

沒有

剪刀 


Add Image Samples:



Webcam



Upload

石頭 

Add Image Samples:



Webcam



Upload

布 


Add Image Samples:



Webcam



Upload

沒有 

Add Image Samples:



Webcam



Upload

剪刀 ✎

Add Image Samples:

Webcam Upload



剪刀 ✎

Webcam

Switch Webcam

27 Image Samples

Hold to Record



剪刀 ✎

30 Image Samples

Webcam Upload

石頭 ✎

30 Image Samples

Webcam Upload

布 ✎

30 Image Samples

Webcam Upload

沒有 ✎

30 Image Samples

Webcam Upload

Training

Train Model

Advanced



Training

Training...

00:07 - 47 / 50

Advanced




Preview

Export Model

Input ON Webcam

Switch Webcam

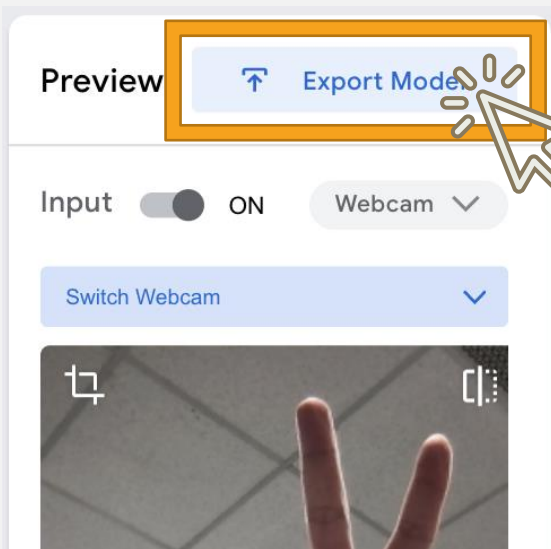


Output

剪刀	100%
石頭	
布	
沒有	

TM 的 TRAINING 參數

- Epochs :
 - 訓練的回合設定，訓練回合數和時間成正比。若要針對細微變化的差異進行辨識時則要提高回合數，會助於模型辨識效能，但特別要注意的是過高的回合數可能會使模型產生「過適 Overfitting」的問題，若是過低的回合數會產生「乏適 Underfitting」的問題，但無論是甚麼問題都會降低模型辨識效能。
- Batch Size :
 - 適當的批次大小設定有助於模型的優化，而且可以提高訓練的速度，及減少訓練誤差。
- Learning Rate (介於 0 ~ 1) :
 - 學習率設定影響著訓練模型尋找最佳解的過程中是否會收斂或發散。若設定過大，雖會快速收斂，但也可能難以收斂產生震盪甚至發散；相對的，設定過小會導致尋找最佳解時緩慢收斂。若有興趣，可以搜尋『梯度下降法』。
- 參考影片：<https://www.youtube.com/watch?v=NRfyTE0xV84&t=2022s>



Export your model to use it in projects.

Tensorflow.js *i* Tensorflow *i* Tensorflow Lite *i*

Export your model:

Upload (shareable link) Download **Update my cloud model**

Your sharable link:

https://teachablemachine.withgoogle.com/models/7qkpiwVGK/ 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">
  // More API functions here
```

Copy

LAB 剪刀石頭布

將訓練好的模型匯入預先寫好的應用網頁



https://flagtech.github.io/FM633A/rsp_game/



Export your model to use it in projects.

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow

Export your model:

Upload (shareable link) Download

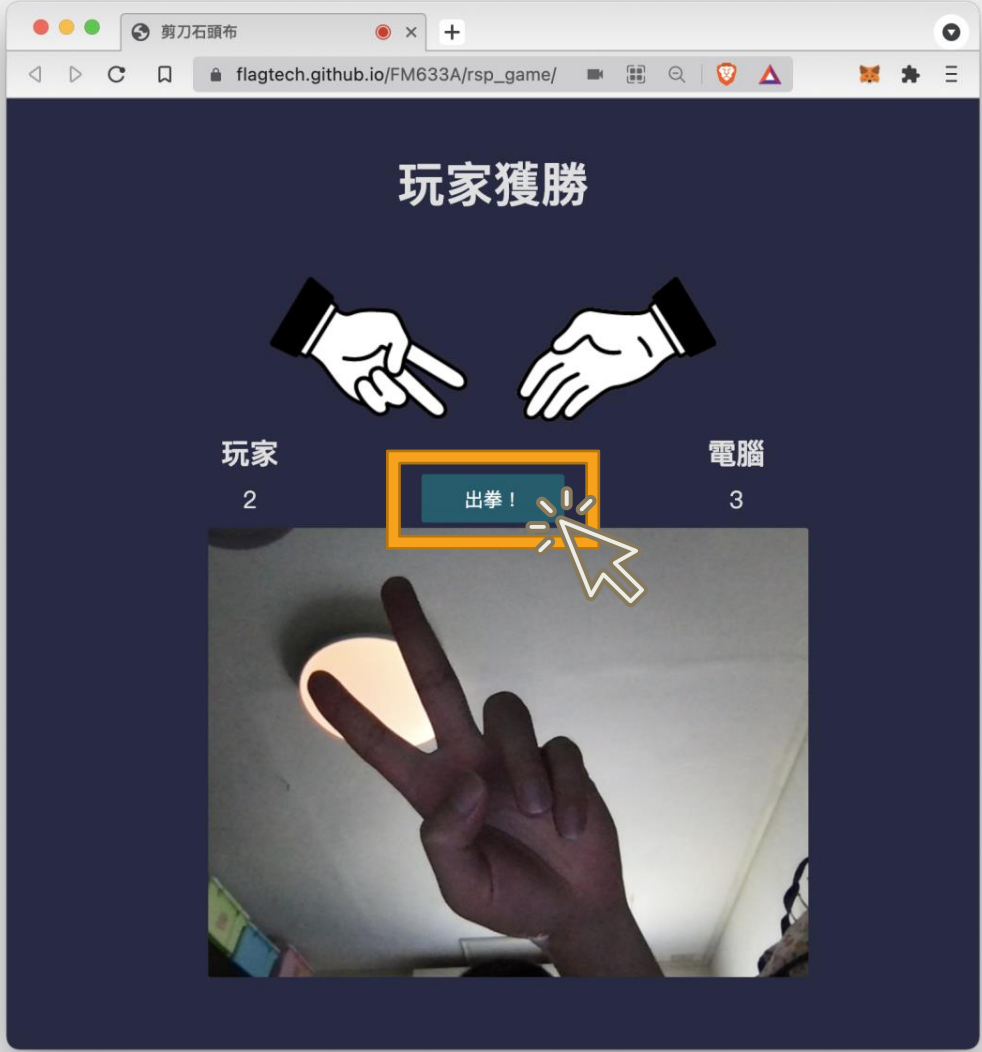
Your sharable link:

<https://teachablemachine.withgoogle.com/models/7qkpiwVGK/> 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.

LAB 匯入模型至應用網頁實驗結果



課程大綱

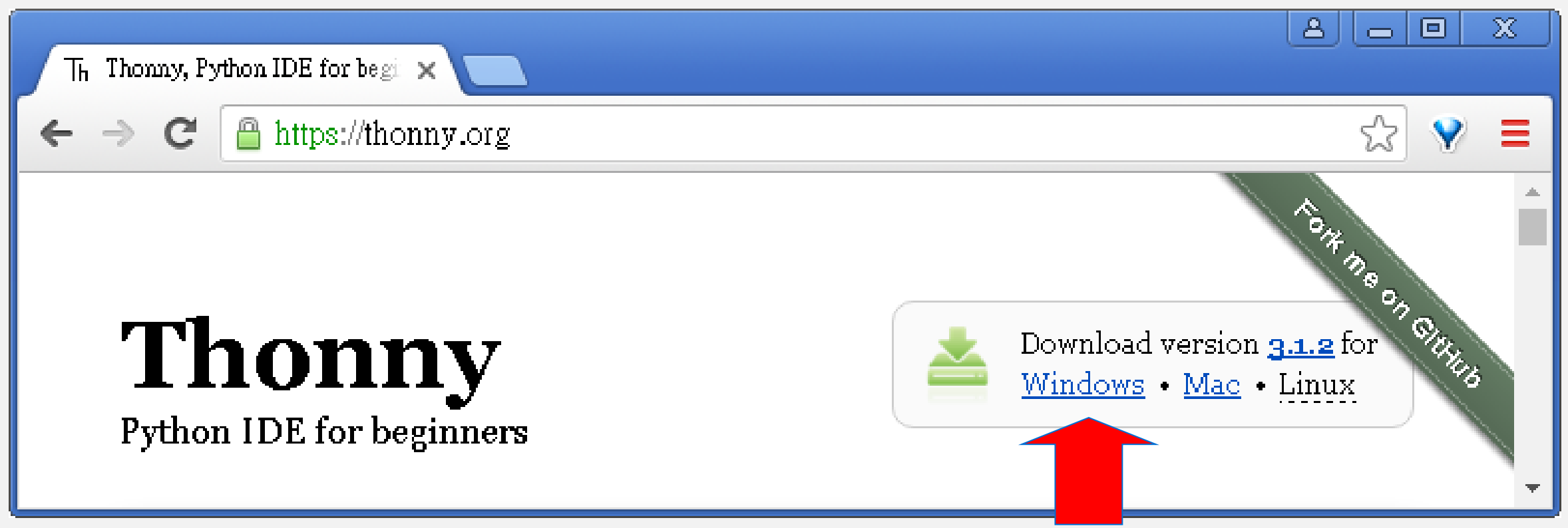
- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



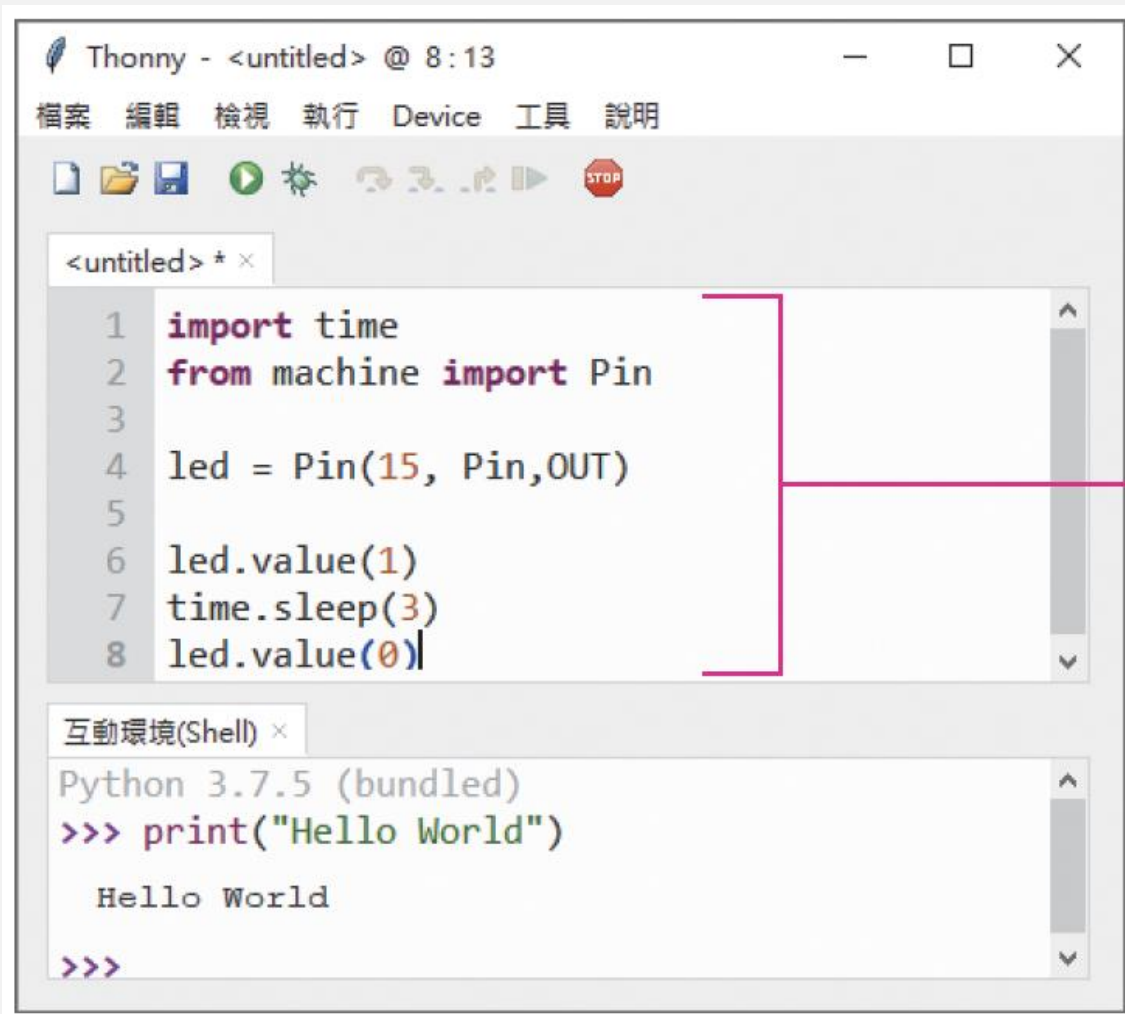
02 互動中心- 控制板

Python 與 DI mini 控制板

- 下載網址：<https://thonny.org>



THONNY 開發環境



在此區域撰寫程式

更改 THONNY 開發環境的文字大小

The image shows a Thonny IDE window with the 'Tools' menu open. The 'Options...' menu item is highlighted. An orange arrow points from this menu item to the 'Thonny Options' dialog box. The dialog box has several tabs, with 'Theme and Font' selected. In this tab, the 'Editor font' is set to 'Consolas' and the size is '13'. A dropdown menu is open next to the size, showing options from 13 to 27, with '14' selected. At the bottom of the dialog, the 'Confirm' button is highlighted. Annotations with red circles and lines point to these key elements.

1 執行選單的『工具 / 選項...』命令 開啟設定視窗

2 切換到主題和字型頁面

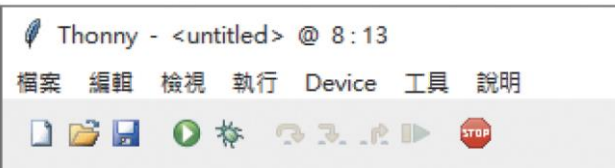
3 在此處選擇字型大小

4 按確認鈕儲存設定

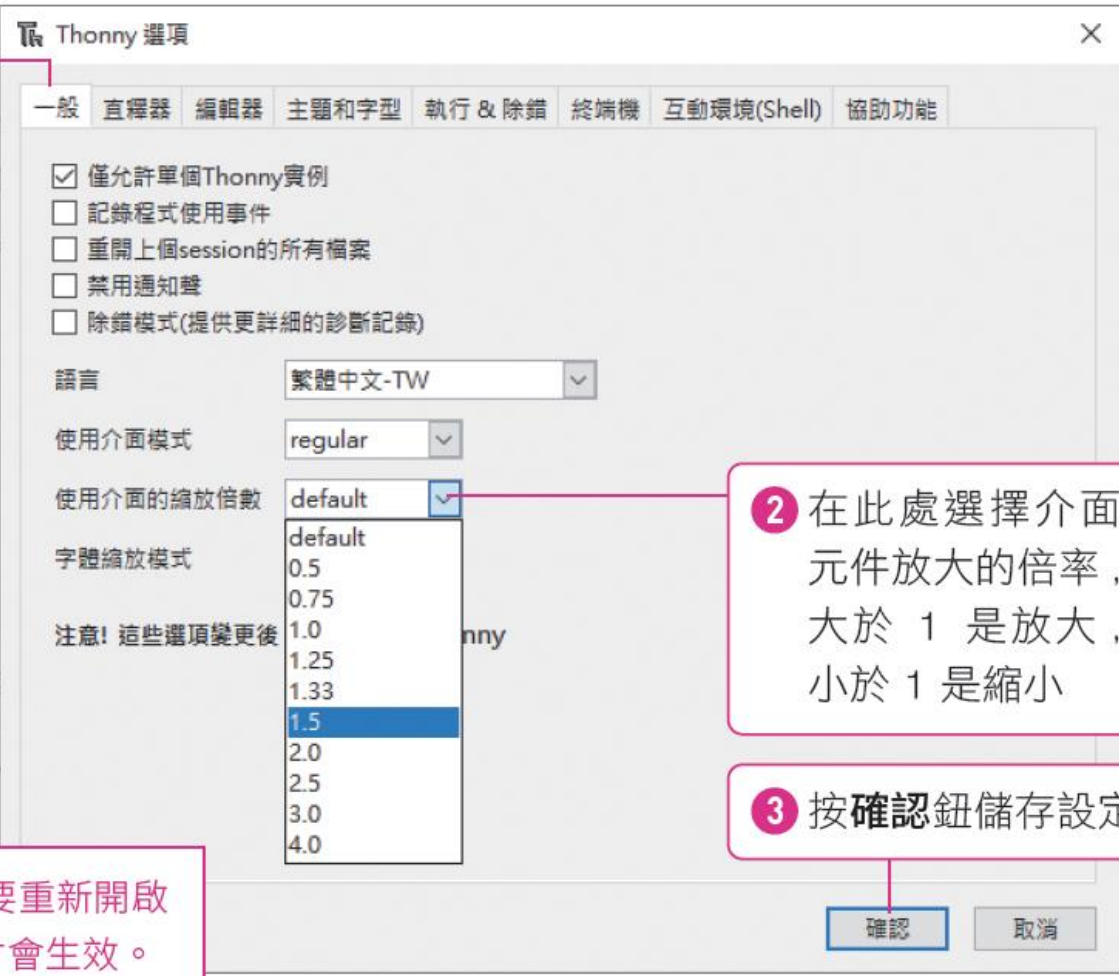
```
def foo(bar):  
    if bar is None: # 這是註解  
        print('The answer is', 33)  
  
unclosed_string = "blah, blah"  
  
>>> %Run demo.py  
Enter an integer: 2.5  
ValueError: invalid literal for int() with base 10:  
'2.5'
```

注意! 有些樣式項目在重新啟動Thonny後才會套用

更改介面上的按鈕大小



1 切換到一般頁面

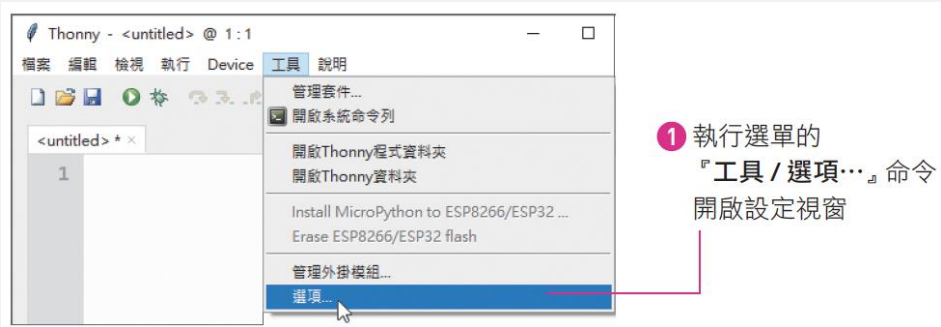


2 在此處選擇介面元件放大的倍率，大於 1 是放大，小於 1 是縮小

3 按**確認**鈕儲存設定

⚠ 此設定需要重新開啟 Thonny 才會生效。

顯示程式碼行號





Python 最基礎

Python 物件、資料型別、變數、匯入模組

物件

- 英文一般寫句子時，會以名詞 + 動詞。Python 是以物件.方法來描述。

文章寫作	寫 Python 程式	
車子	<code>car</code>	car 物件
車子向前進	<code>car.start()</code>	car 物件的 start 方法

- 方法後面會加上括號()`()`，有些方法需要加入額外的參數。
 - ✓ 例如：`car.go(100)`，車子加速到 100。
- 若方法有多個參數，以逗點分隔。
 - ✓ 例如：`car.left(50, 30)`，以 50 的速度，向左轉 30 度。

練習：字串物件

- 在互動模式中，輸入下列敘述：(>>> 指的是在互動模式中，執行單行敘述)

```
>>> "abc".upper() —— 使用字串物件 "abc" 的 upper() 方法，  
                        將字串轉成大寫  
'ABC'  
>>> "abc".find('b') —— find() 方法尋找 'b' 的位置  
                        (從 0 開始)  
1  
>>> "abc".replace('b', 'z') —— replace() 方法將所有的  
                        'b' 取代成 'z'  
'azc'
```

- 不同的物件會有不同的方法。例如：字串物件與整數物件。

資料型別

- 除字串物件以雙引號或單引號來表示，寫程式常有整數與浮點數(小數)物件，例如：111 與 11.1。

```
>>> 111 + 111 ————— 整數物件相加
```

```
222
```

```
>>> "111" + "111" ————— 字串物件串接
```

```
'111111'
```

- 上述 + 的運算，因物件的資料不同而產生不同的結果。物件的種類，程式語言稱之為『物件型態』或『資料型態』(Data Type)。

練習：要分清楚資料型別

- 兩個資料型別若不同，可能會導致程式錯誤。

```
>>> 111 + "111" —— 不同型別的資料相加，發生錯誤  
Traceback (most recent call last):
```

```
  File "<ipython-input-6-4832c22160be>", line 1, in  
<module>  
    111 + "111"
```

```
TypeError: unsupported operand type(s) for +: 'int'  
and 'str'
```

型別轉換

- 兩個資料型別若要運算，可以使用型別轉換。

```
>>> str(111) + "111"  —— str() 可轉換物件為字串型別
'111111'
>>> 111 + int("111") —— int() 可轉換物件為整數型別
222
```

- 整數與浮點數的數學運算

- ✓ + (加)、- (減)、* (乘)、/ (取商)、// (取商，整數)、% (取餘數)、** (指數)。
- ✓ Python 允許整數與浮點數直接運算，執行下列程式：

加法 (+)

```
>>> 10 + 5.5  
15.5
```

減法 (-)

```
>>> 10 - 5.5  
4.5
```

乘法 (*)

```
>>> 10 * 5.5  
55.0
```

除法取商 (/)

```
>>> 10 / 5.5  
1.8181818181818181
```

除法取整數商 (//)

```
>>> 10 // 5.5  
1.0
```

取餘數 (%)

```
>>> 10 % 5.5  
4.5
```

指數 (**)

```
>>> 4 ** 0.5, 8 ** (1/3)  
(2.0, 2.0)
```

TIP

1. 整數與浮點數做運算，結果一定為浮點數。
2. 只有整數與整數做除法，結果為浮點數。

常用的變數運算

- 把整數加上特定的值：

```
>>> x = 1
>>> x = x + 1
>>> x
2
```

- 常用的簡式：

簡式	意義
$x += 2$	$x = x + 2$
$x -= 2$	$x = x - 2$
$x *= 2$	$x = x * 2$

簡式	意義
$x /= 2$	$x = x / 2$
$x //= 2$	$x = x // 2$
$x %= 2$	$x = x \% 2$

- 『變數』(variable) 就像是掛在物件的名牌，幫物件取名之後，讓我們方便識別物件與操作，其語法為：

變數名稱 = 物件

- 例如：

```
>>> n1 = 123456789 —— 將整數物件 123456789 指派給變數 n1
>>> n2 = 987654321 —— 將整數物件 987654321 指派給變數 n2
>>> n1 + n2 —— 實際上是 123456789 + 987654321
1111111110
```

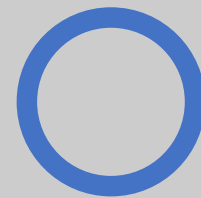
寫出可讀性高的程式 (1/2)

- 使用有意義的變數 (variable) 的名稱。

```
a = 3.1  
b = 2.2  
c = a * b * b
```



```
pi = 3.1  
radius = 2.2  
# 使用公式計算圓面積  
circle_area = pi * radius * radius
```



寫出可讀性高的程式 (2/2)

- 將程式加上註解。
 - ✓ 註解可以幫助其他人了解這個程式。
- 較佳的註解：

```
# 註解 1：使用公式計算圓面積
```

- 不好閱讀的註解：

```
# 註解 2：將圓周率乘半徑乘半徑
```

內建函式

- 『**函式**』 (function) 是一段預先寫好的程式，方便重複使用。而程式先將經常使用到的功能以函式的形式先寫好，稱為『**內建函式**』。
- 例如：print() 是最常用的顯示函數：

```
>>> print("abc") —— 顯示字串物件
```

```
abc
```

```
>>> print("abc".upper()) —— 顯示字串物件.方法的執行結果
```

```
ABC
```

```
>>> print(111 + 111) —— 顯示整數物件運算的結果
```

```
222
```

匯入模組

- 內建函式不就越多越好？若內建函式無限制增加，會導致啟動速度越來越慢，執行時佔用的記憶體越來越多。
- 『**模組**』 (module)，就是將同一類的函式打包成模組，預設不會啟用。需要時，再用**匯入** (import) 的方式啟用。預先寫好的稱為『**內建模組**』。

```
>>> import time —— 匯入時間相關的 time 模組
```

```
>>> time.sleep(3) —— 執行 time 模組的 sleep() 函式，暫停 3 秒
```

```
>>> from time import sleep —— 從 time 模組裡匯入 sleep() 函式
```

```
>>> sleep(5) —— 執行 sleep() 函式，暫停 5 秒
```

練習：匯入模組

程式

暫停 3 秒後，印出 Hello World! 字串物件

```
# 暫停 3 秒後，印出 Hello World!  
from time import sleep  
sleep(3)  
print("Hello World!")
```

指的是在程式編輯窗格中編輯與執行。

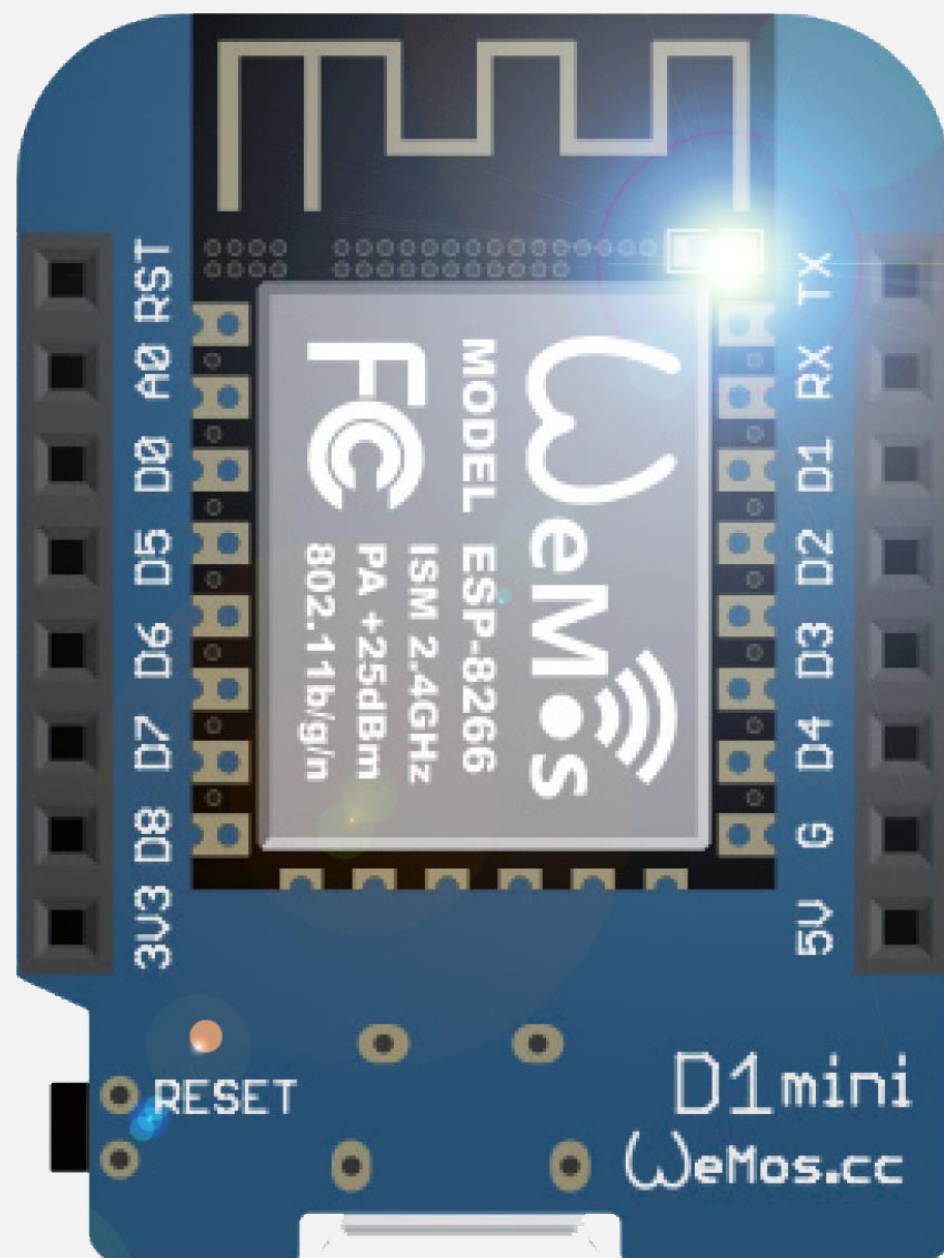
觀念整理

各種程式語言的語法邏輯都差不多，就像人類語言的文法。

1. 程式的每一個東西都是**物件**，有些物件有其特定的操作方法。
2. 基本物件有**資料型別**，型別不同，結果不同。甚至有時會錯誤。
3. **變數**是物件的名牌而已，也方便程式設計師操作。
4. 使用有意義的變數名稱與註解。
5. **內建函式**是經常用的函式，預先寫好的。
6. 適當的**匯入模組**，能精簡效能。

LAB 點亮控制板內建 LED 燈

連接 D1 mini 與電腦,再使用 Thonny 開發環境編寫 Python 程式點亮和熄滅控制板內建 LED



安裝與設定 DI MINI

1 連線 http://www.wch.cn/downloads/CH341SER_EXE.html

WCH 沁恒

产品中心 应用方案 沁恒社区 服务支持 关于沁恒

全部

CH341SER.EXE

适用范围	版本	上传时间	资料大小	
CH340G · CH340T · CH340C · CH340E · CH340B · CH341A · CH341T · CH341B · CH341C · CH341U	3.4	2016-09-28	237KB	下载

CH340/CH341USB转串口WINDOWS驱动程序。支持32/64位 Windows 10/8.1/8/7/VISTA/XP · SERVER 2016/2012/2008/2003 · 2000/ME/98。通过微软数字签名认证。支持USB转3线和9线串口等。用于随产品发行到最终用户。

产品手册

开发资源

2 按此鈕下載

驅動程式安裝步驟

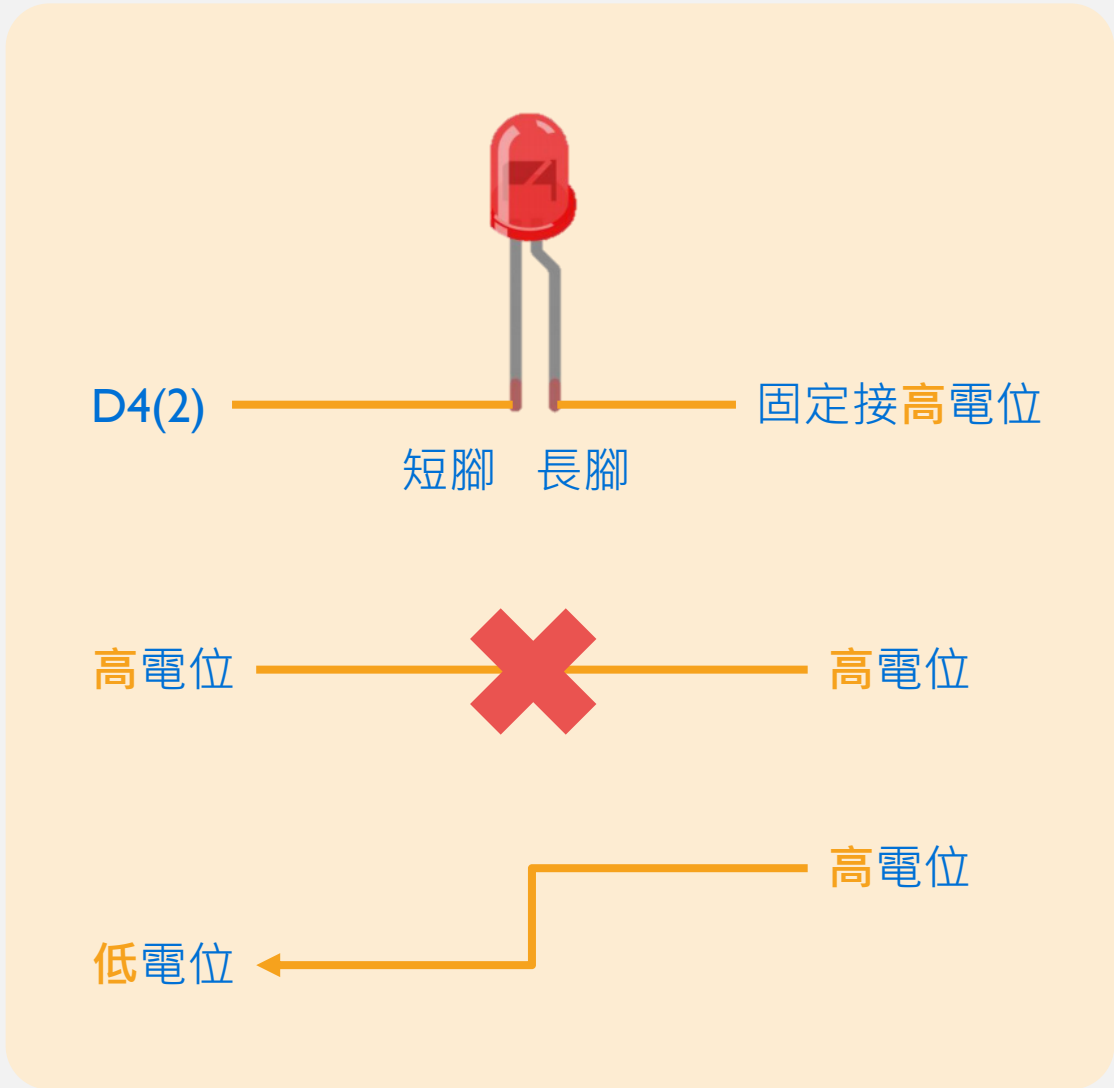
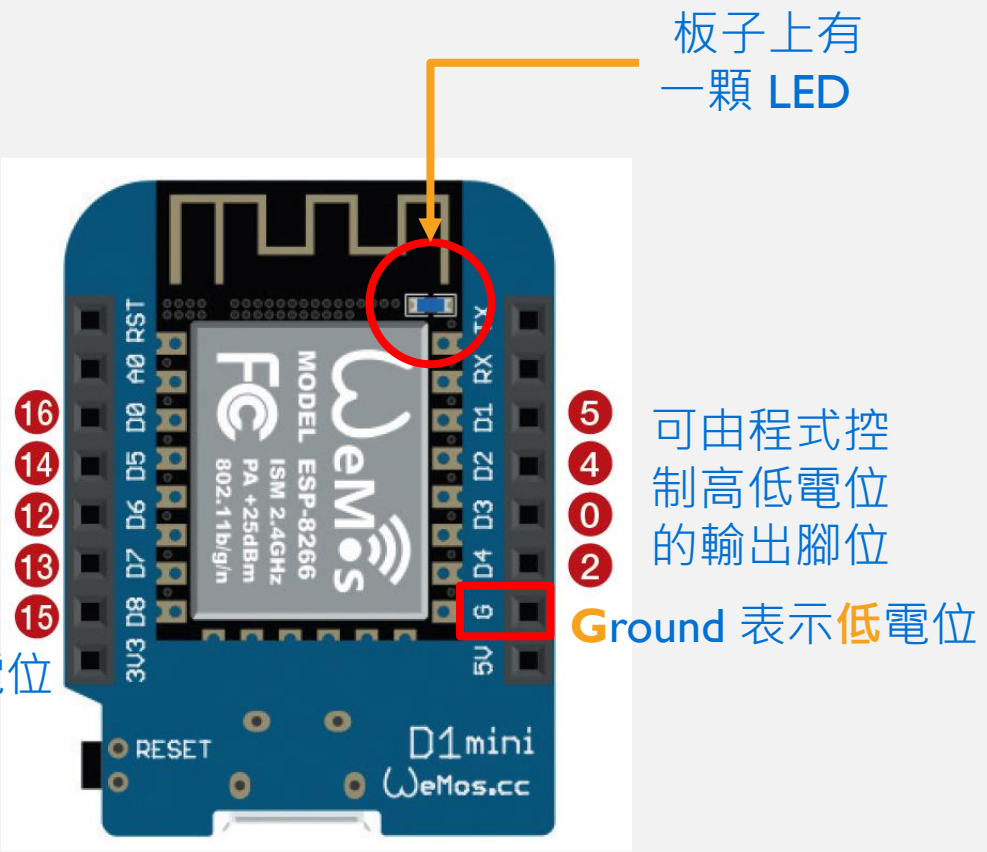
The image shows a three-step process for installing a driver. Step 1: A WinRAR self-extracting archive window displays a list of files being extracted, including CH341S98.SYS, CH341SER.CAT, CH341SER.INF, CH341SER.SYS, CH341SER.VXD, SETUP.EXE, DRVSETUP64\DRVSETUP64.exe, CH341PT.DLL, CH341S64.SYS, and DRVSETUP64. The destination folder is C:\WCH.CN\CH341SER, and an installation progress bar is shown. Step 2: The DriverSetup(X64) window is open, showing 'Device Driver Install / UnInstall' for CH341SER.INF. The 'INSTALL' button is highlighted with an orange box. Step 3: A smaller DriverSetup window displays 'Driver install success!' with a blue information icon, and the '確定' (OK) button is highlighted with an orange box.

②按『INSTALL』開始安裝

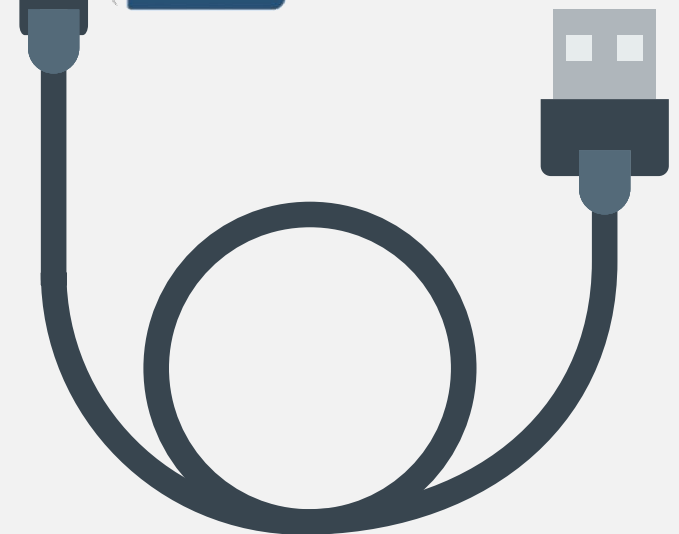
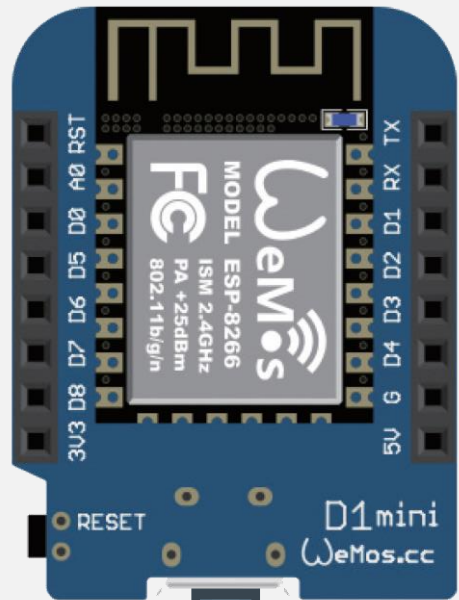
①會先解開驅動程式安裝檔案

③安裝完成

閃爍 LED 燈



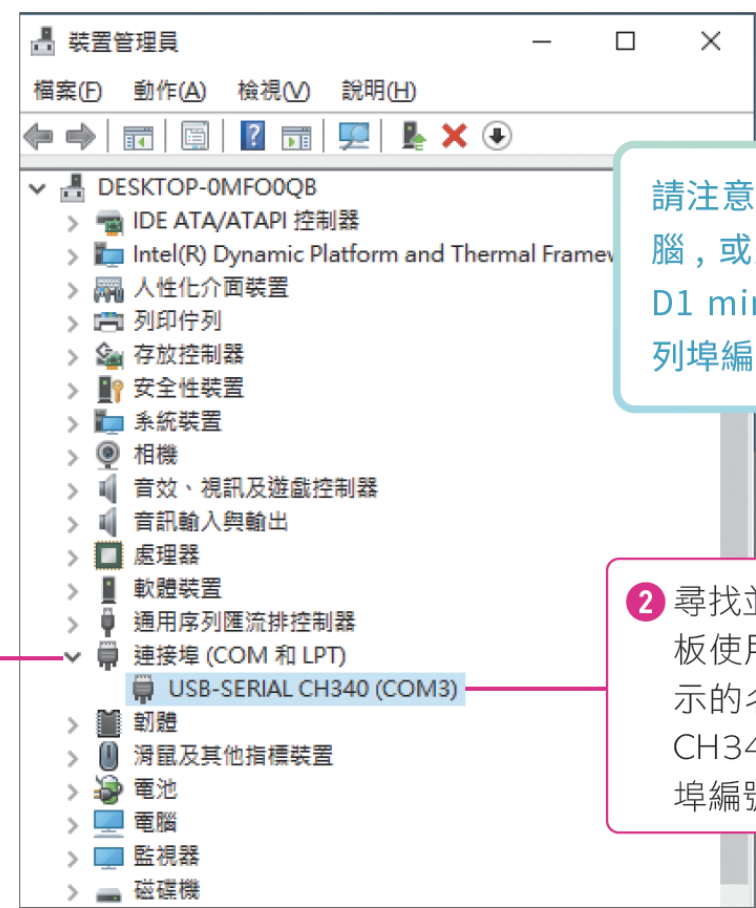
連接 D1 MINI 與電腦



『開始 / 控制台 / 系統及安全性 / 系統 / 裝置管理員』 (Win7) OR 按右鈕執行『裝置管理員』命令 (Win10)



1 展開連接埠項目



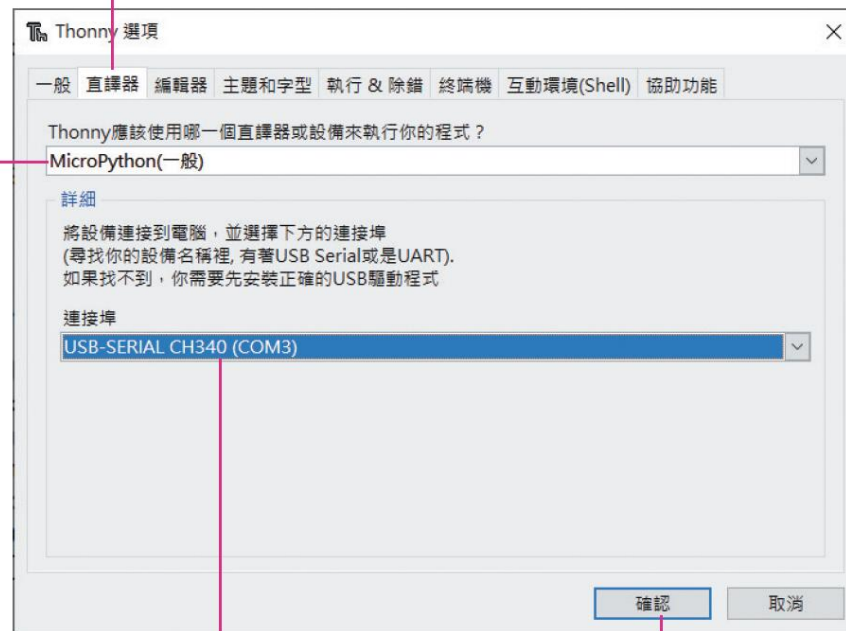
請注意，使用不同的電腦，或是連接到不同的 D1 mini 控制板，其序列埠編號都可能不同

2 尋找並記下 D1 mini 控制板使用的序列埠編號 (顯示的名稱是 USB-SERIAL CH340, COM3 表示序列埠編號為 3)



1 執行選單的『工具 / 選項...』命令，開啟設定視窗

2 切換到直譯器頁面



3 拉下選單選擇

MicroPython(一般) 列埠編號 (Mac 上請選有 "/dev/cu.usbserial-1440" 字樣的項目)

4 拉下選單選擇剛剛記下的序

5 按**確認**鈕儲存設定

控制 D4 腳位的電位高低

請在互動窗格內測試

用 machine 模組的 Pin 功能

```

互動環境 (Shell) x
>>> from machine import Pin
>>>
>>> led = Pin(2, Pin.OUT)
>>>
>>> led.value(0)
>>>

```

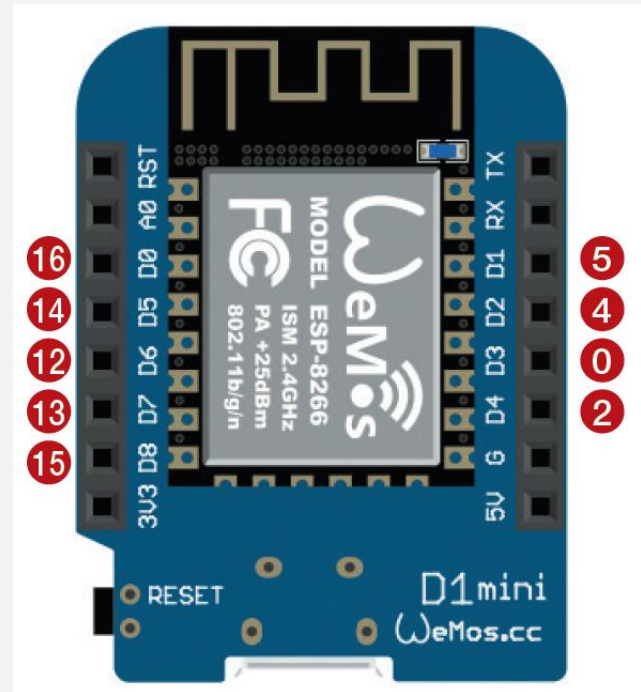
表示要控制的腳位編號

表示要控制高低電位

1 : 高電位
0 : 低電位

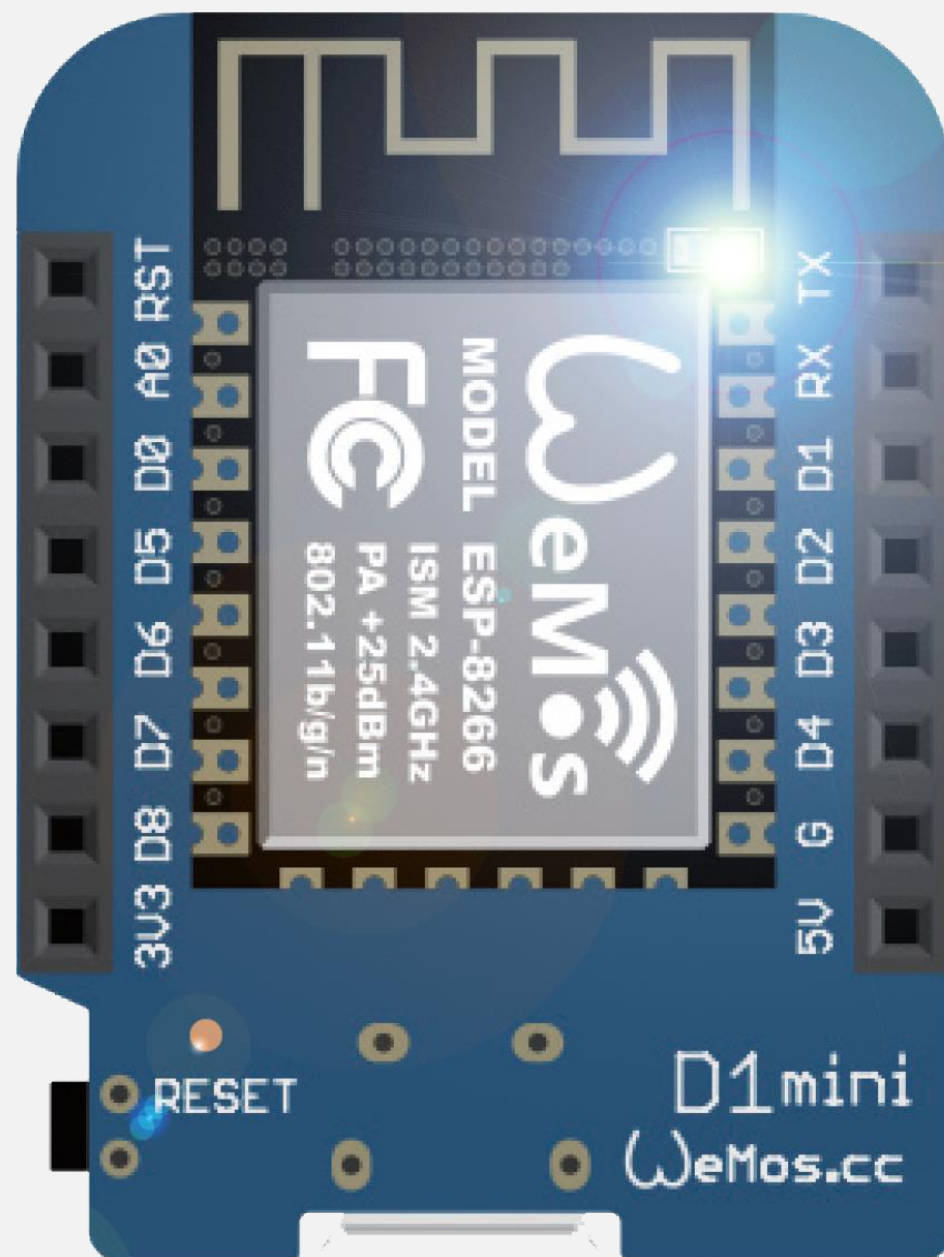
到這裡燈應該就亮起來了
現在考一考自己讓燈熄滅

請注意大小寫



LAB01 不斷閃爍 LED 燈

使用迴圈、程式的執行與終止



```

Thonny - C:\Users\ShinWei\Desktop\tets00.py @ 11:20
檔案 編輯 檢視 執行 工具 說明
[Run Button]
tets00.py x
1 from machine import Pin
2 import time
3
4 # 建立物件 (# 開始是註解)
5 led = Pin(2, Pin.OUT)
6
7 while True:
8     led.value(0)
9     time.sleep(0.5)
10    led.value(1)
11    time.sleep(0.5)

```

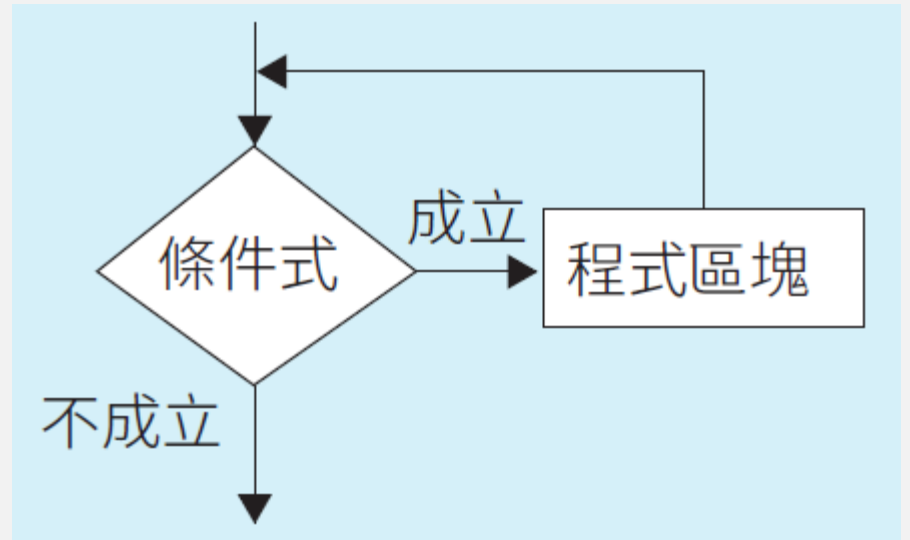
打完後請按執行鈕或 F5 鍵

是寫在上面的編輯區喔！！

表示無條件成立
一定要加上冒號

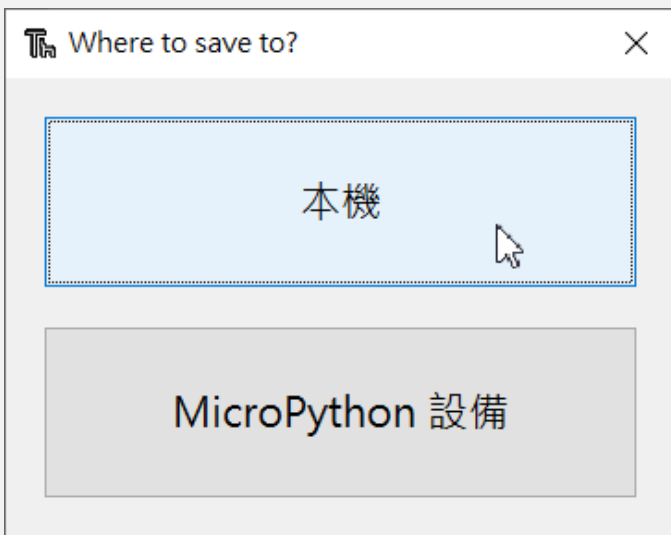
這幾行一定要空相同格數 (Thonny 會自動幫你空官方推薦的 4 格)

while 條件式：
程式區塊

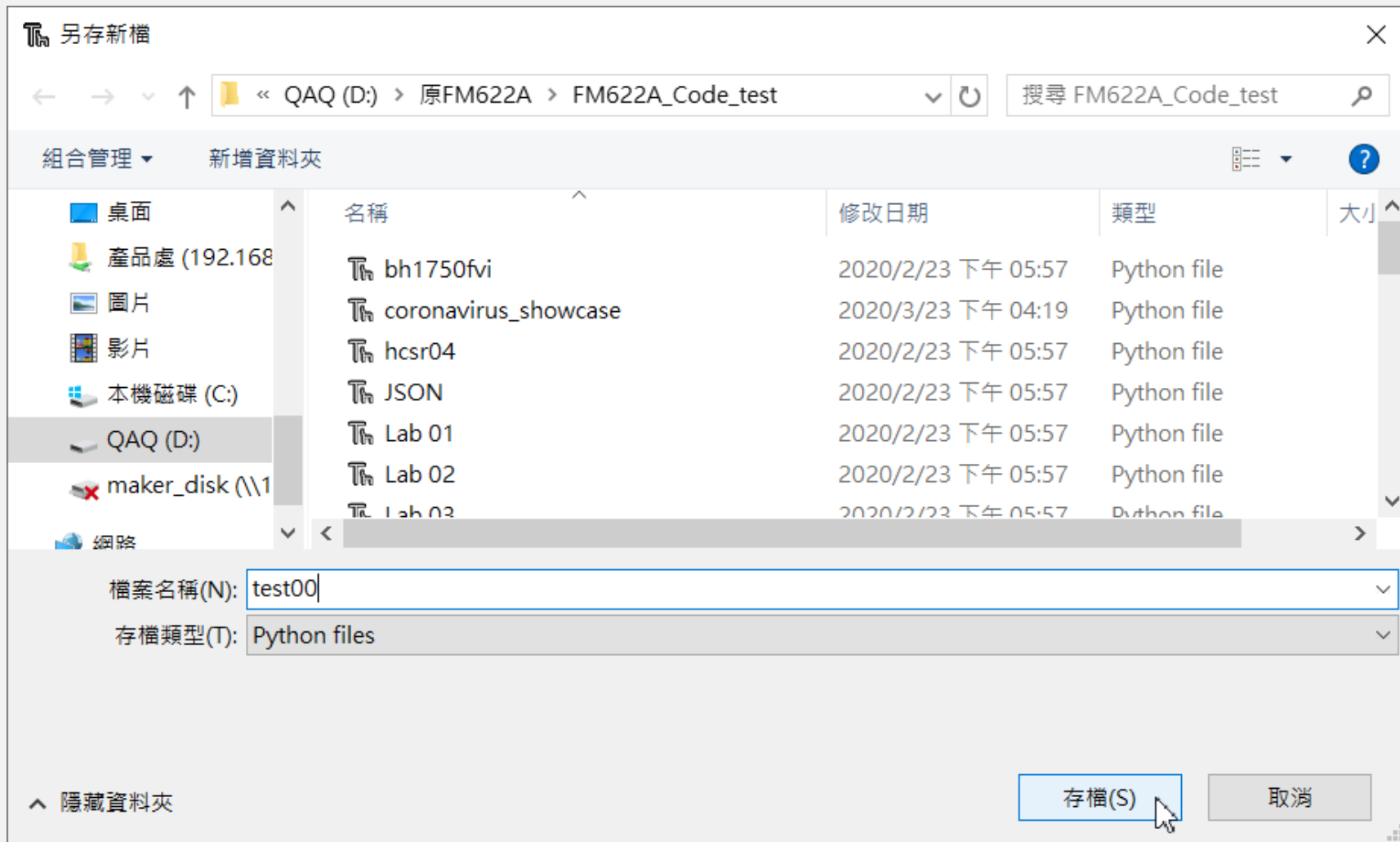


必須要存檔才可以執行

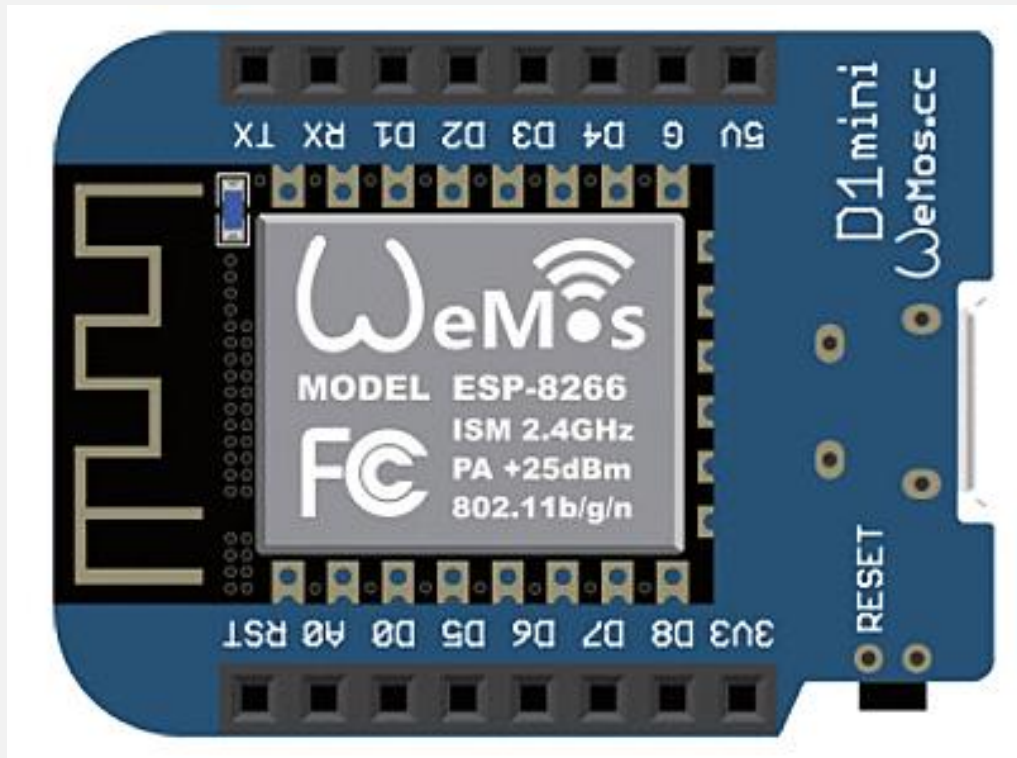
本機就是你的電腦



MicroPython 設備
就是你的控制板

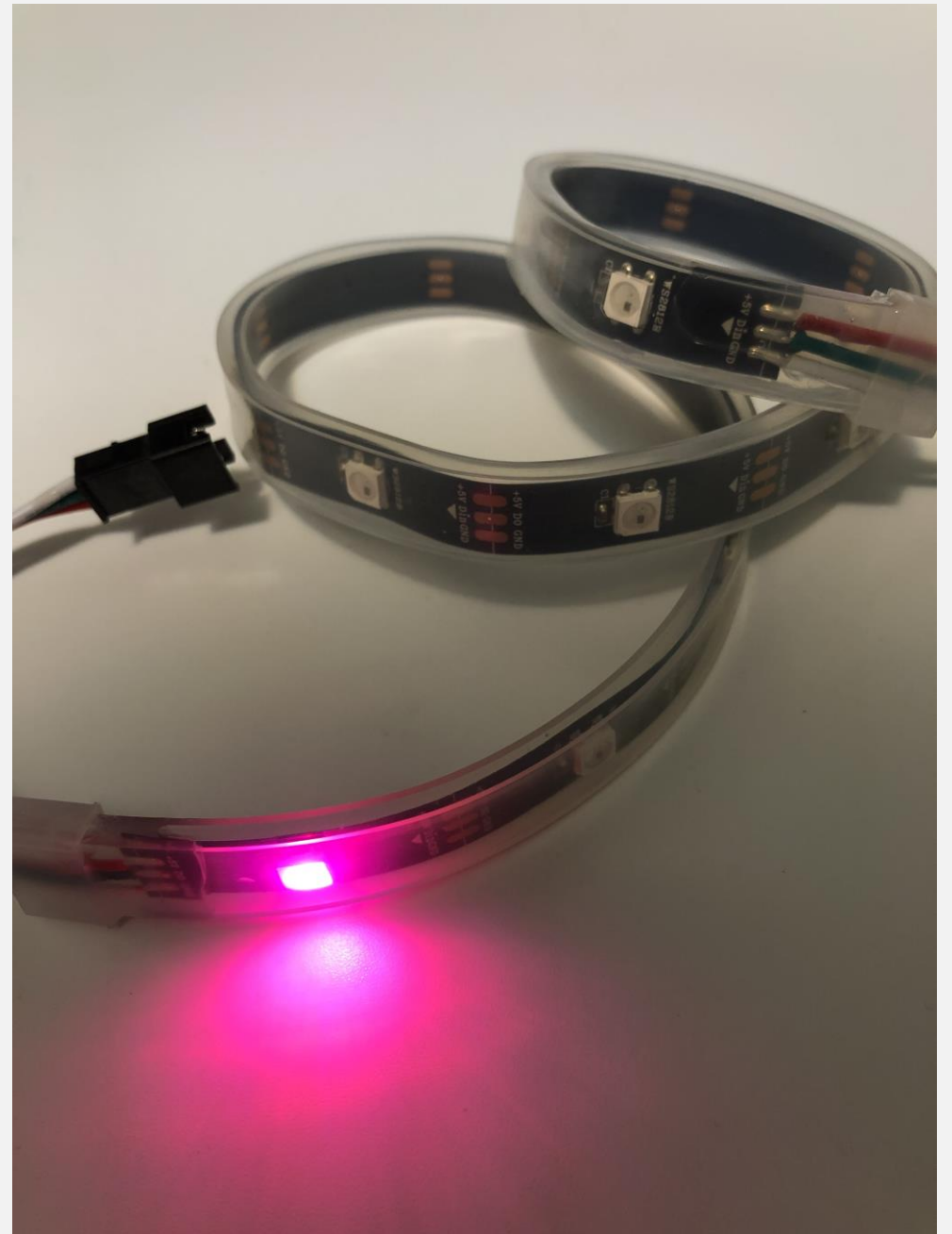


LAB01 點亮控制板內建 LED 燈實驗結果



LAB02 點亮外接 LED 條燈

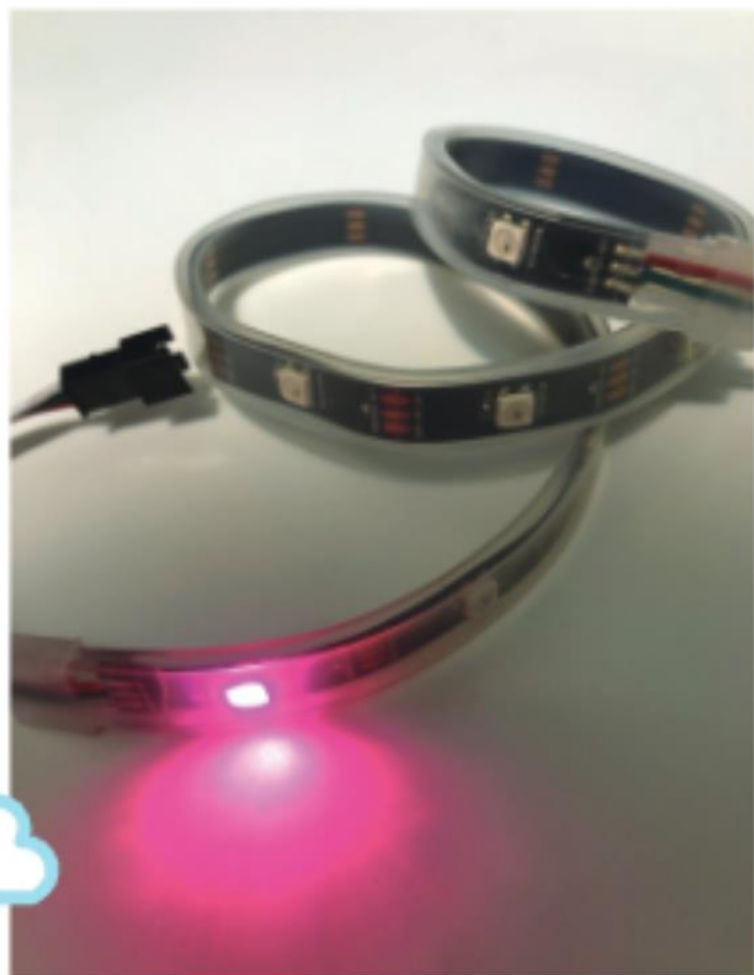
編寫 Python 程式控制外接 LED 燈亮度與顏色



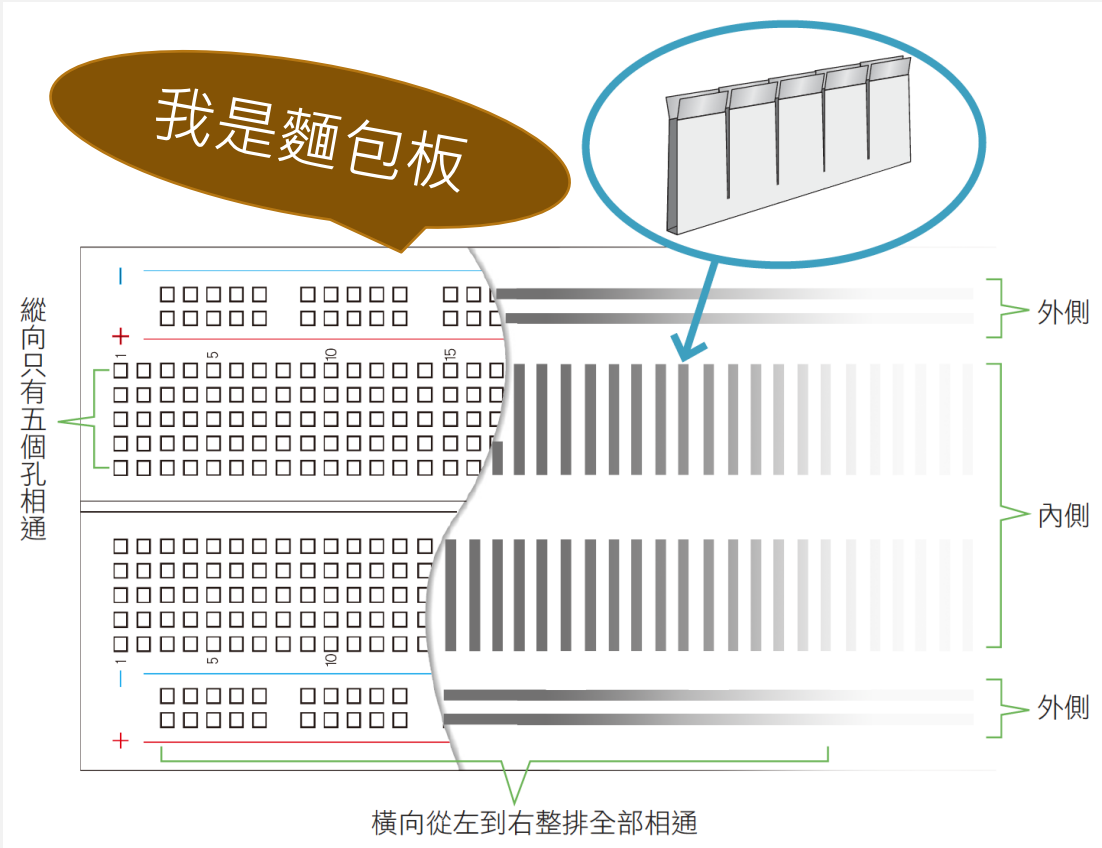
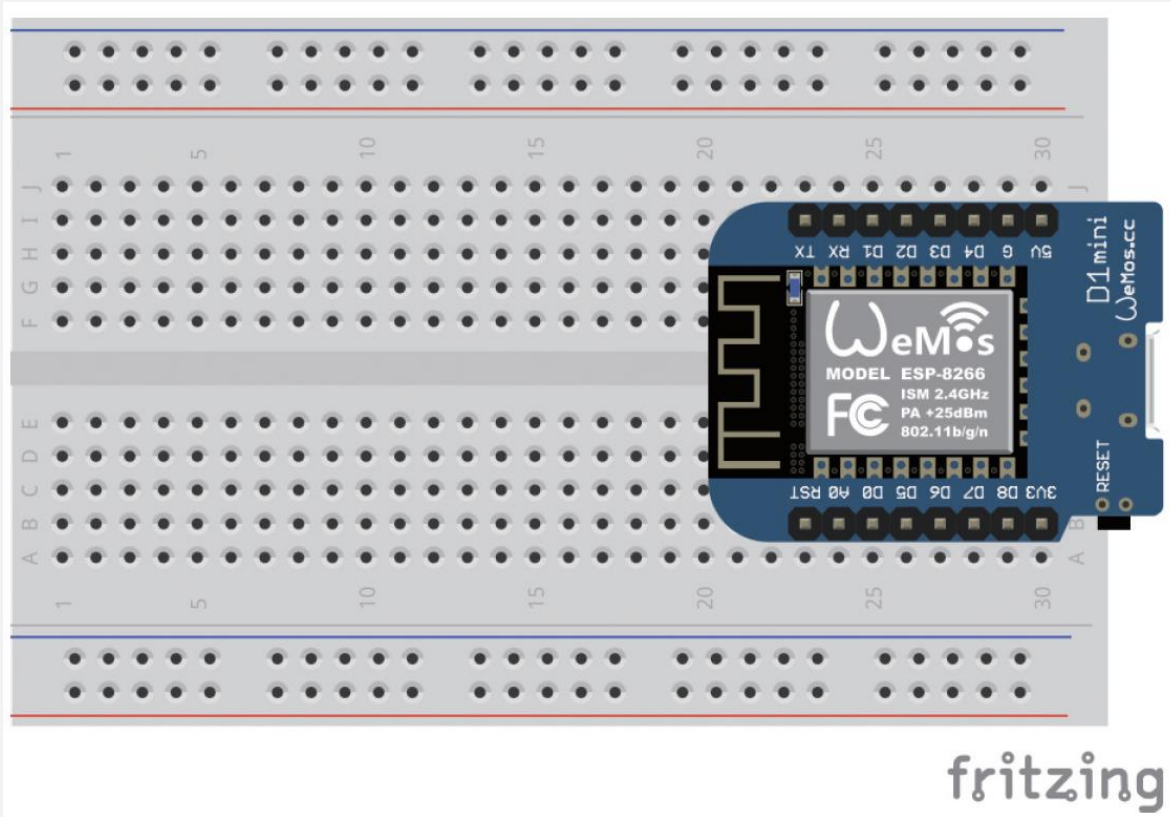
RGB LED 條燈

本套件所附的條燈上分別有 15 顆可程式控制的 RGB 燈珠，可以藉由程式讓燈珠依照想要的亮度以及顏色顯示，而且可以多條串接使用。

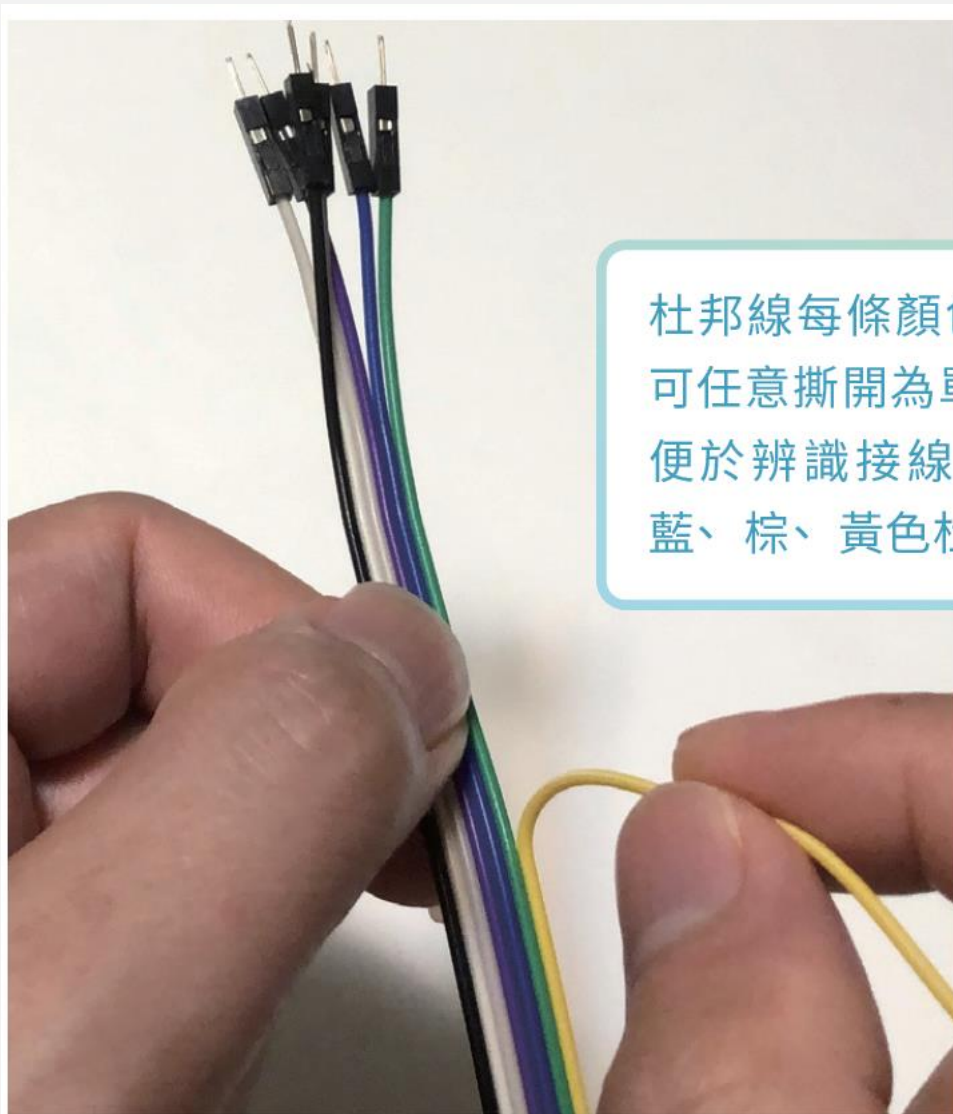
防水 RGB LED 條燈



接線圖

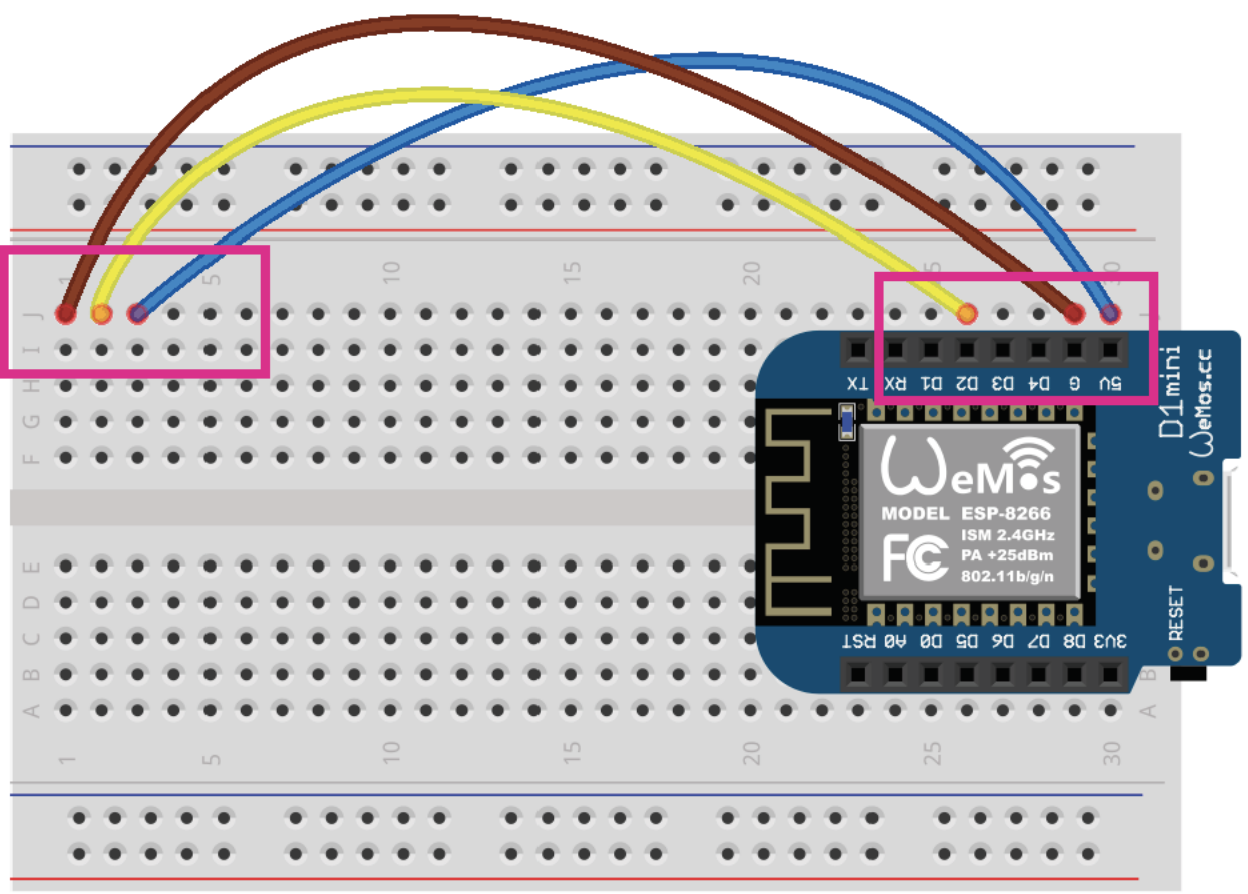


撕下杜邦線：藍、棕、黃色



杜邦線每條顏色不同但功能相同，可任意撕開為單條使用，本實驗為便於辨識接線，請依照指示撕下藍、棕、黃色杜邦線使用

依照圖片插上杜邦線

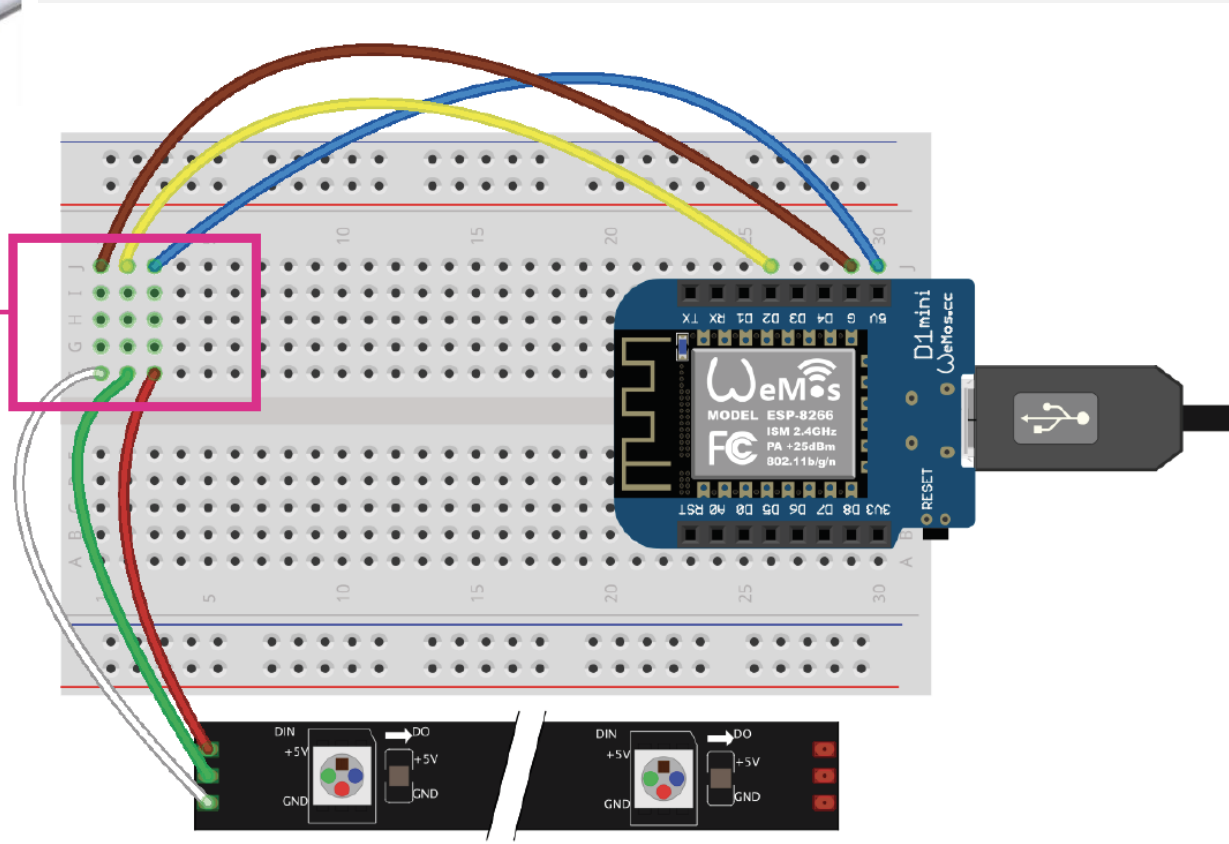


杜邦線顏色	對應腳位
黃色	D2
棕色	G
藍色	5V

取出紅、綠、白色杜邦線



紅綠白分別
對應藍黃棕



範例程式：LAB02.PY

```
# 從 machine 模組匯入 Pin 物件
from machine import Pin

# 從 neopixel 模組匯入控制條燈的 NeoPixel 物件
from neopixel import NeoPixel

# 建立 NeoPixel 物件
led_strip = NeoPixel(Pin(4), 15)

# 設定第 1 顆燈珠數值
led_strip[0] = (0, 0, 255)

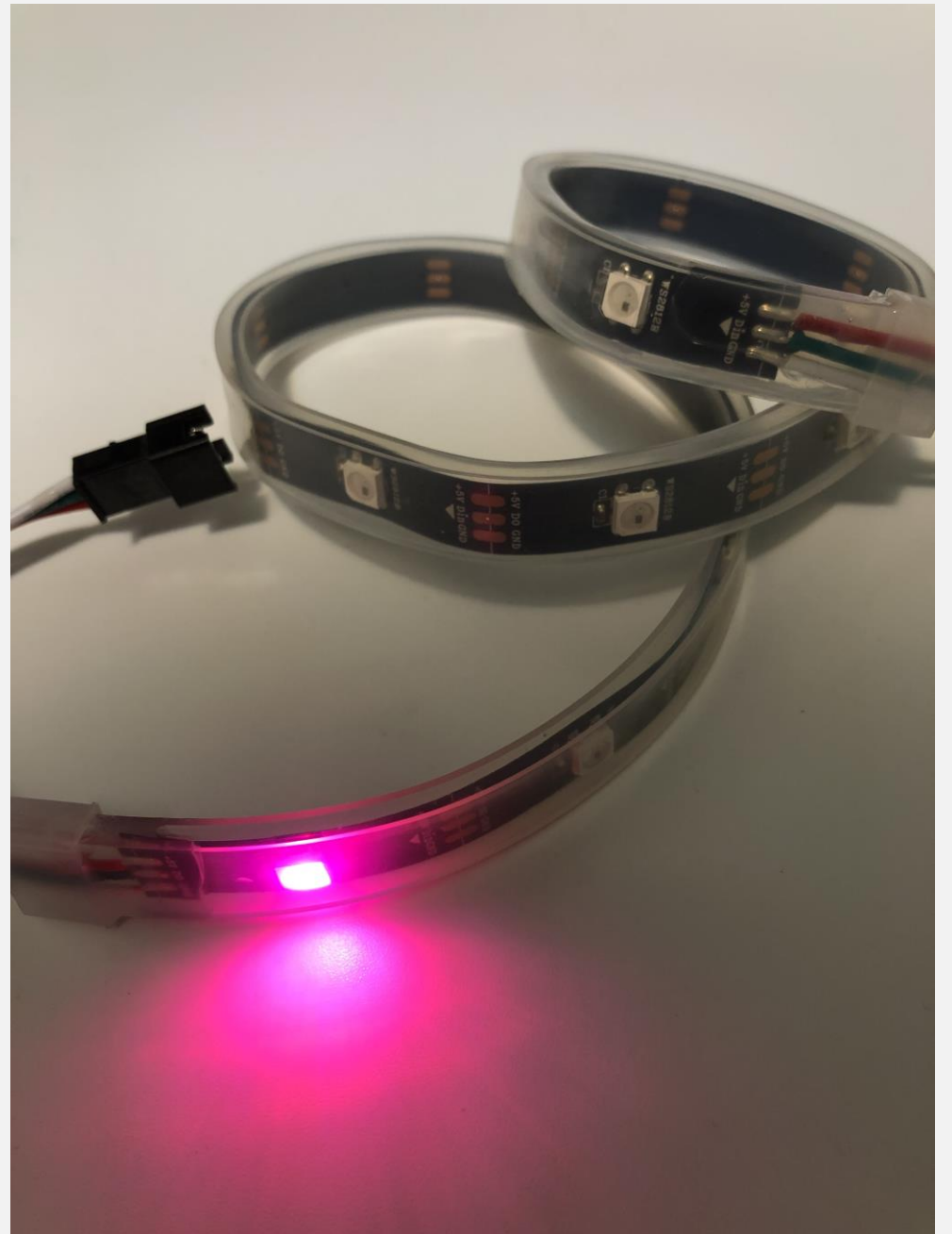
# 依設定顯示
led_strip.write()
```

LAB02 點亮外接 LED 條燈實驗結果



LAB03 LED 特效跑馬燈

for 迴圈控制外接 LED 燈亮度與顏色



控制所有的 LED 燈

- 點亮所有的 LED 燈

```
led_strip[0] = (0, 48, 64)
```

```
led_strip[1] = (0, 48, 64)
```

```
led_strip[2] = (0, 48, 64)
```

```
...
```

```
Led_strip.write()
```

- 熄滅所有的 LED 燈

```
Led_strip[0] = (0, 0, 0)
```

```
Led_strip[1] = (0, 0, 0)
```

```
Led_strip[2] = (0, 0, 0)
```

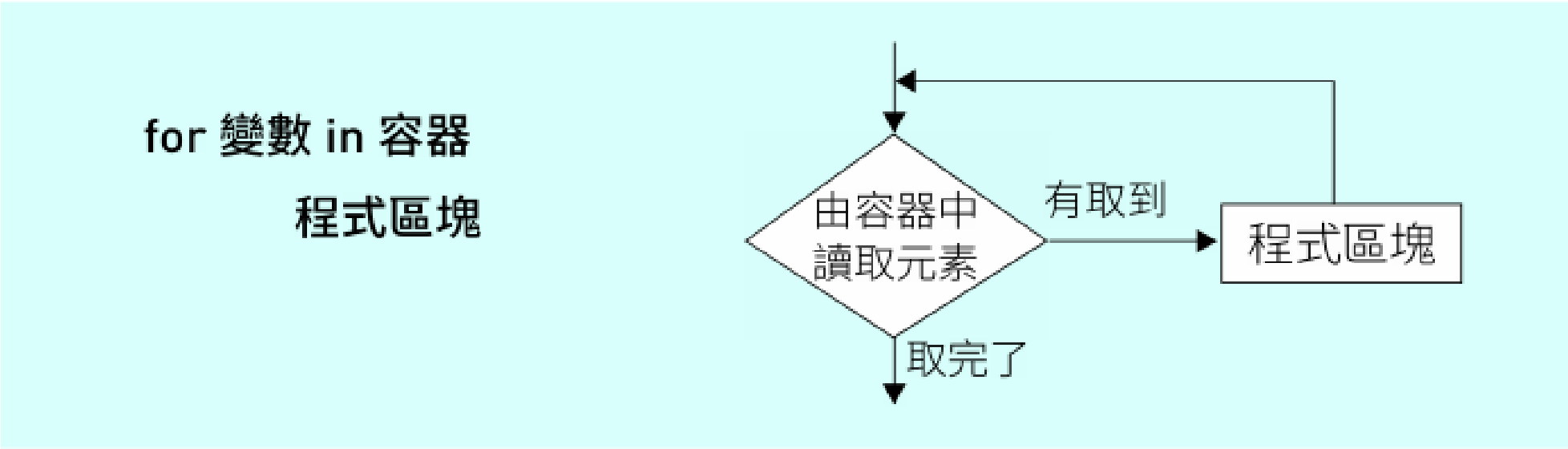
```
...
```

```
led_strip.write()
```

- 若有 15 顆 LED，就要重複輸入 15 次，若有 1000 顆.....暴力法

PYTHON 流程控制 (FOR 迴圈)

為了解決這個問題，Python 提供了 for 迴圈的語法，for 迴圈可將容器中的元素一一讀取出來做處理，其語法如下：



⚠ 與 while 和 if 一樣，for 迴圈的程式區塊也要內縮 4 個空白。

PYTHON 流程控制 (FOR 迴圈)

為了產生一個有 0~1023 數值的容器，我們還可以使用 Python 內建的 range() 來產生一個指定範圍的數列容器，其語法如下：

- range (x) ← 產生「由 0 到 x 但不包含 x」的數列
- range (x, y) ← 產生「由 x 到 y 但不包含 y」的數列
- range (x, y, z) ← 產生「由 x 到 y 但不包含 y, 間隔為 z」的數列

PYTHON 流程控制 (FOR 迴圈)

for 迴圈搭配 range() 的範例如下：

```
>>> for i in range(10):  
    print(i)
```

← 產生 0 到 10 但不包含 10 的數列
← 輸出 0 1 2 3 4 5 6 7 8 9
← Thonny 會幫你自動內縮

0
1
2
3
4
5
6
7
8
9

多按一次空行才
會結束 for 執行

PYTHON 流程控制 (FOR 迴圈)

```
>>> for i in range(1, 11):  
    print(i)
```

← 產生 1 到 11 但不包含 11 的數列
← 輸出 1 2 3 4 5 6 7 8 9 10

1
2
3
4
5
6
7
8
9
10

⚠ 可以用「有頭無尾」的口訣來記憶 range() 會產生的數列！

LAB03 LED 特效跑馬燈

```
from machine import Pin
from neopixel import NeoPixel
import time

led_strip = NeoPixel(Pin(4), 15)

while True:
    for i in range(led_strip.n):
        led_strip[i] = (0, 48, 64)
        led_strip.write()
        time.sleep_ms(100)
        led_strip[i] = (0, 0, 0)
```

課程大綱

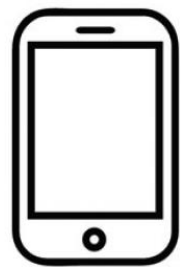
- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



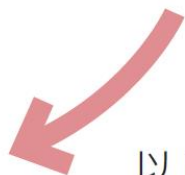
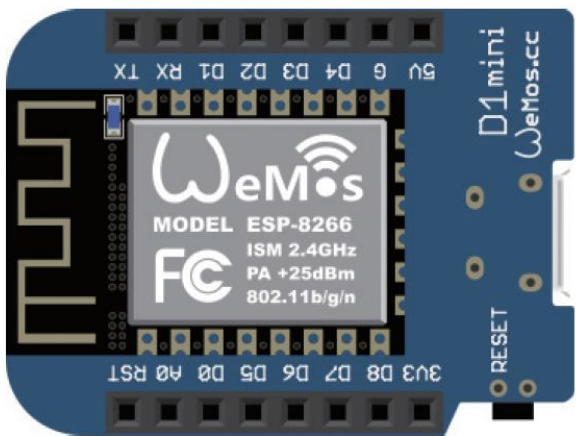
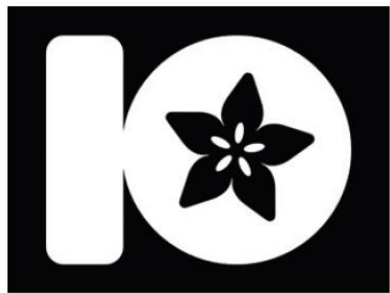
03 影像辨識 IOT 物聯網應用

MQTT 通訊與應用

ADAFRUIT IO 雲端資料庫

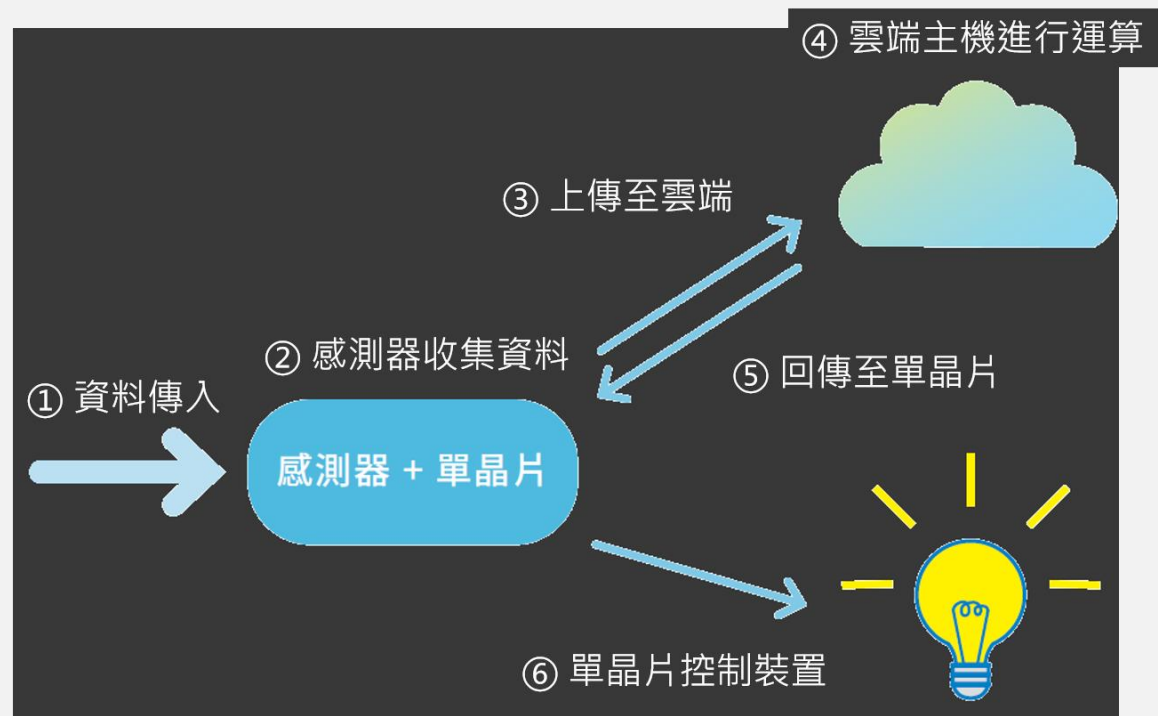


網頁端將辨識結果
傳至雲端資料庫 AIO



以 MQTT 通訊
取得辨識結果

▲ 建立網頁伺服器 & 訂閱 AIO



https://io.adafruit.com

The screenshot shows the 'accounts.adafruit.com/users/sign_up' page. It features a 'SIGN UP' form with the following fields: FIRST NAME, LAST NAME, EMAIL, USERNAME, and PASSWORD. Below the form are two buttons: 'CREATE ACCOUNT' and 'SIGN IN'. Annotations with numbered circles point to the name and email fields (1), the username and password fields (2), and the 'CREATE ACCOUNT' button (3).

1 依序輸入您的名字、姓氏與 Email

2 輸入您想要使用的帳號名稱與密碼

3 按此鈕建立帳號



The screenshot shows the user profile settings page. It includes a success message: 'Welcome! You have signed up successfully.' and a notification: 'We think you are in the Beijing time zone, if this isn't correct, please choose the correct time zone below.' The settings form contains: GRAVATAR EMAIL, FAVORITE COLOR (set to green), and TIME ZONE (set to '(GMT+08:00) Taipei'). A 'SAVE SETTINGS' button is at the bottom.

請確認此欄位的時區是否正確，若錯誤請自行選擇正確時區

如果修改時區的話，請按此鈕儲存

Shop Learn Blog Forums LIVE! AdaBox IO

adafruit Profile **Feeds** Dashboards Triggers Services My Key

Feeds | **view all** Learn Guides | view all

face

1 按 Feeds 連結

2 再點 View All

Adafruit IO Basics: Feeds

3 點擊 +New Feed 按鈕

+ New Feed + New Group

Default

Feed Name

Loaded in 0.75 seconds.

Create a new Feed X

Name

mood

Maximum length: 128 characters. Used: 4

4 輸入 mood 再按下 Create

+ New Feed + New Group Search

Default Disable Feed Delete Feed

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> mood	mood		less than a minut...
<input checked="" type="checkbox"/> Welcome Feed	welcome-feed		2 minutes ago

5 勾選不會用到的 Feed

mood 建立成功

6 按此刪除

7 點擊 My Key

adafruit Profile Feeds Dashboards Triggers Services My Key

The Adafruit IO SSL/TLS Certificate was updated. More details can be found here.

The user interface has been updated. If you have any feedback or suggestions please post on the forums here.

YOUR ADAFRUIT IO KEY

Your Adafruit IO Key should be kept in a safe place and treated the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

Username

colombo0718

你的 Adafruit IO 帳號名稱

Active Key

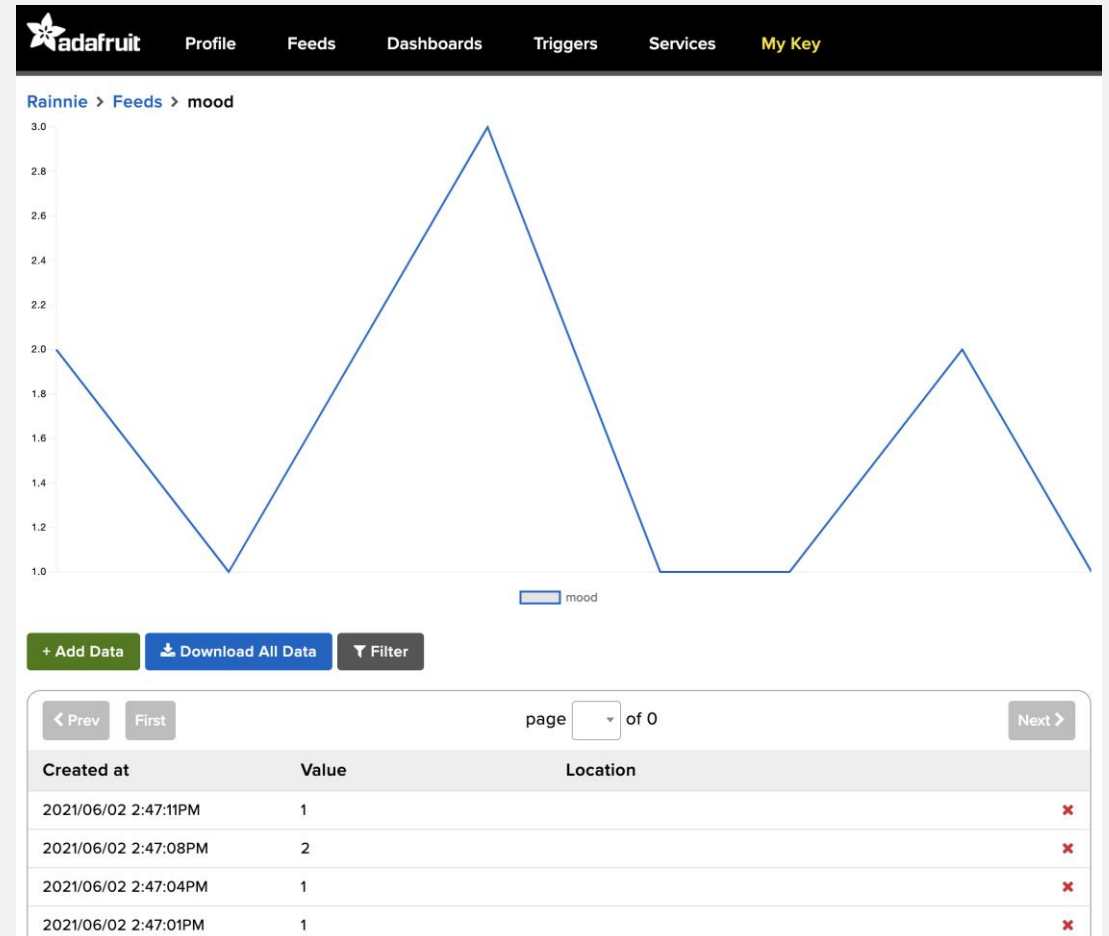
aio_gege27MYy9hnTx6Wg7YSE7tpjdRA

你的 Adafruit IO 金鑰

LAB04 心情日記

將表情辨識結果傳至 MQTT 伺服器

有些人可能曾經興起想寫日記的念頭,但很多時候連續一陣子都沒發生什麼特別的事,寫著寫著就覺得像是流水帳,索性就不寫了,這樣實在有點可惜,或許需要的只是每天的心情紀錄,一個笑臉、哭臉都是一個紀錄,這時候就可以利用 AIO 資料庫搭配臉部表情分類來幫我們記下任何想紀錄的時候。



https://teachablemachine.withgoogle.com/train/image

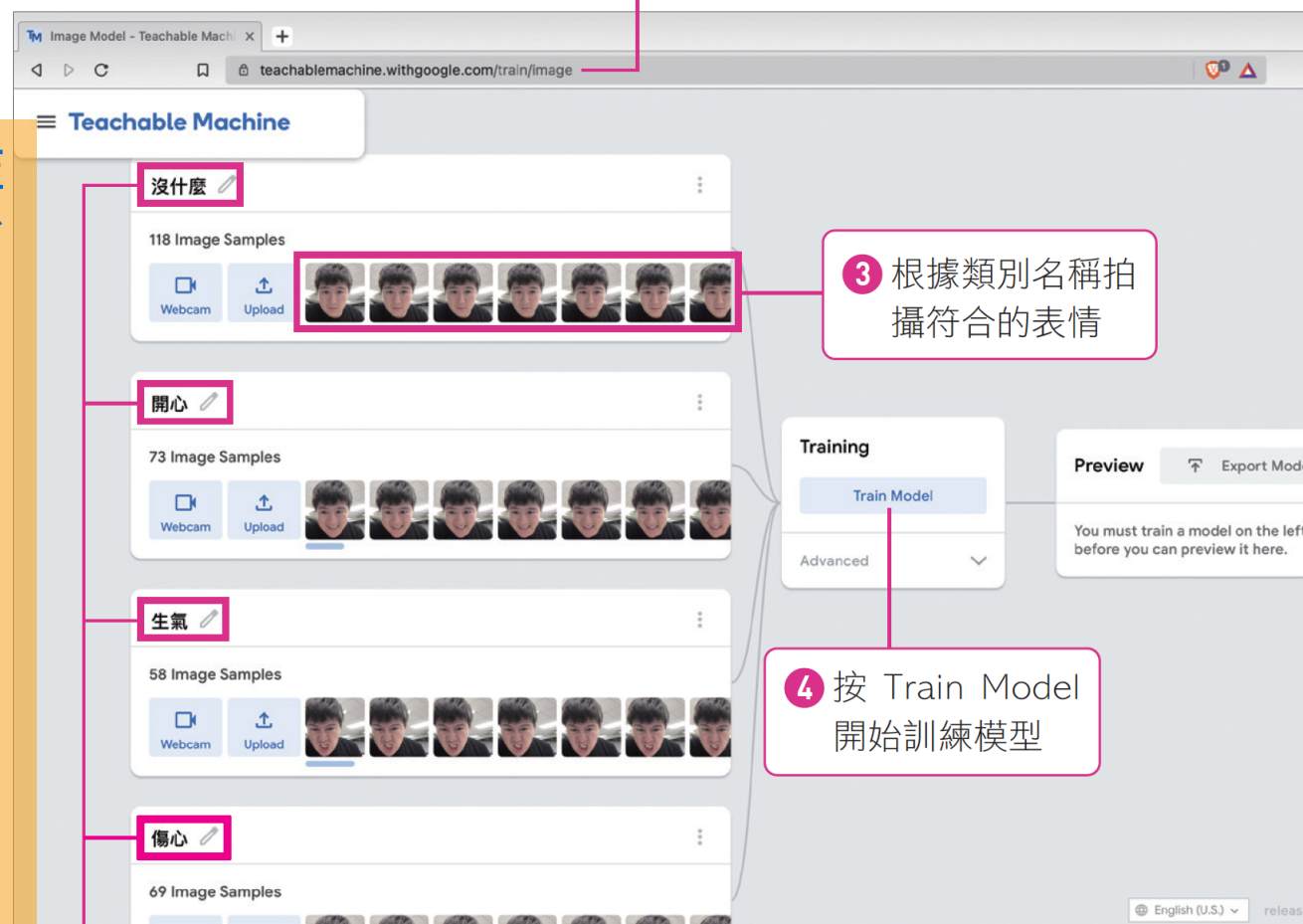
沒什麼

開心

生氣

傷心

1 連線 <https://teachablemachine.withgoogle.com/train/image>



3 根據類別名稱拍攝符合的表情


4 按 Train Model 開始訓練模型

2 新增並修改類別名稱為：沒什麼、開心、生氣、傷心

5 按此輸出模型

Preview **Export Model**

Input ON Webcam



Output

- 沒什麼
- 開心
- 生氣 100%
- 傷心

Training

Model Trained

Advanced



Export your model to use it in projects.

Tensorflow.js | Tensorflow | Tensorflow Lite

Export your model:

Upload (shareable link) Download **Upload my model**

6 按此上傳模型至雲端

Your sharable link:

[https://teachablemachine.withgoogle.com/models/\[...\]](https://teachablemachine.withgoogle.com/models/[...])

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

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>
```

Copy



Export your model to use it in projects.

Tensorflow.js | Tensorflow | Tensorflow Lite

Export your model:

Upload (shareable link) Download **Update my cloud model**

Your sharable link:

<https://teachablemachine.withgoogle.com/models/r-t4uSsXj/> **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.

確認上傳完成

按此可複製網址

https://flagtech.github.io/FM633A/lab04.html

LAB04 心情日記

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

填入各項
資訊

按 **Start**

LAB04 心情日記

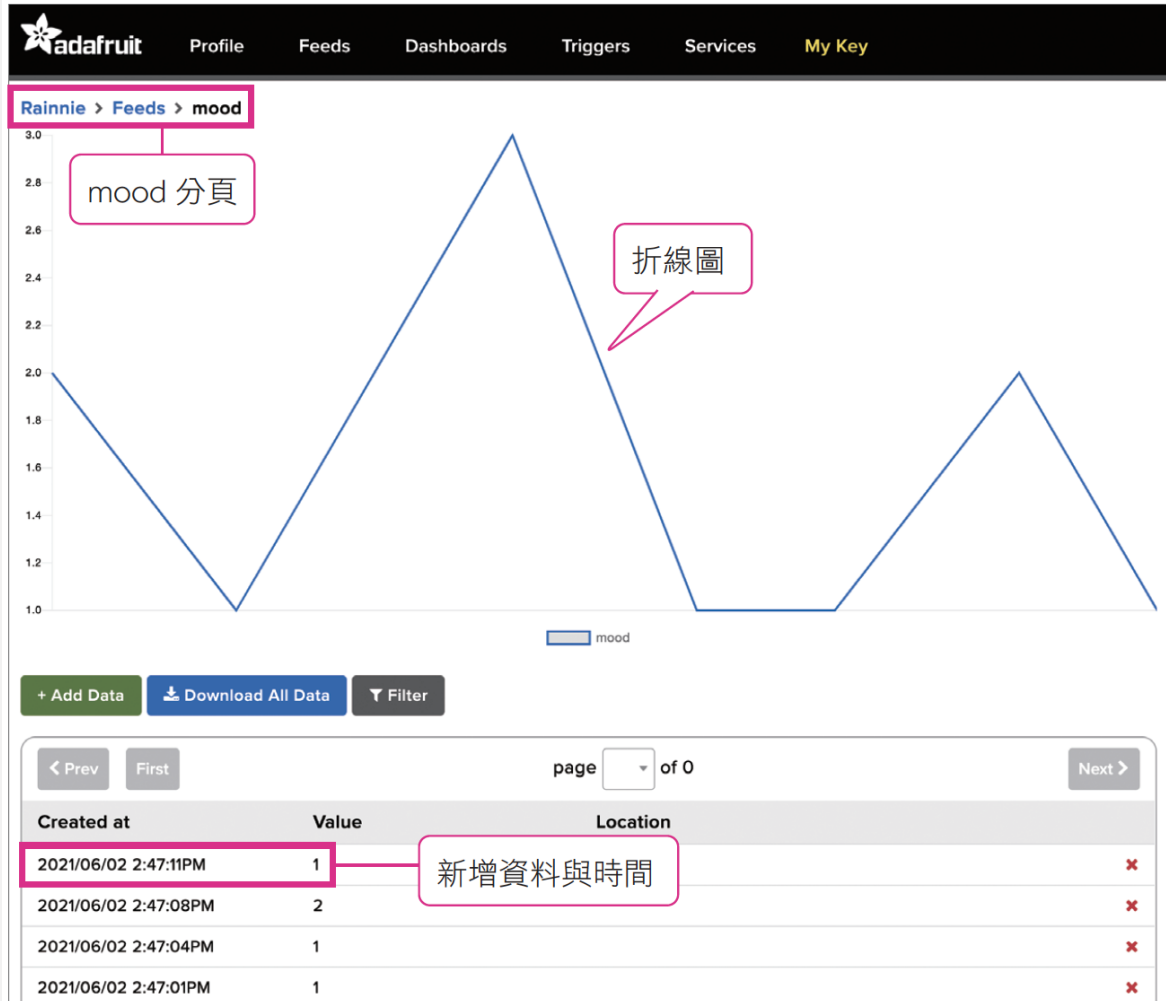
您看起來有77%心情是 生氣



分享心情

按此分享心情

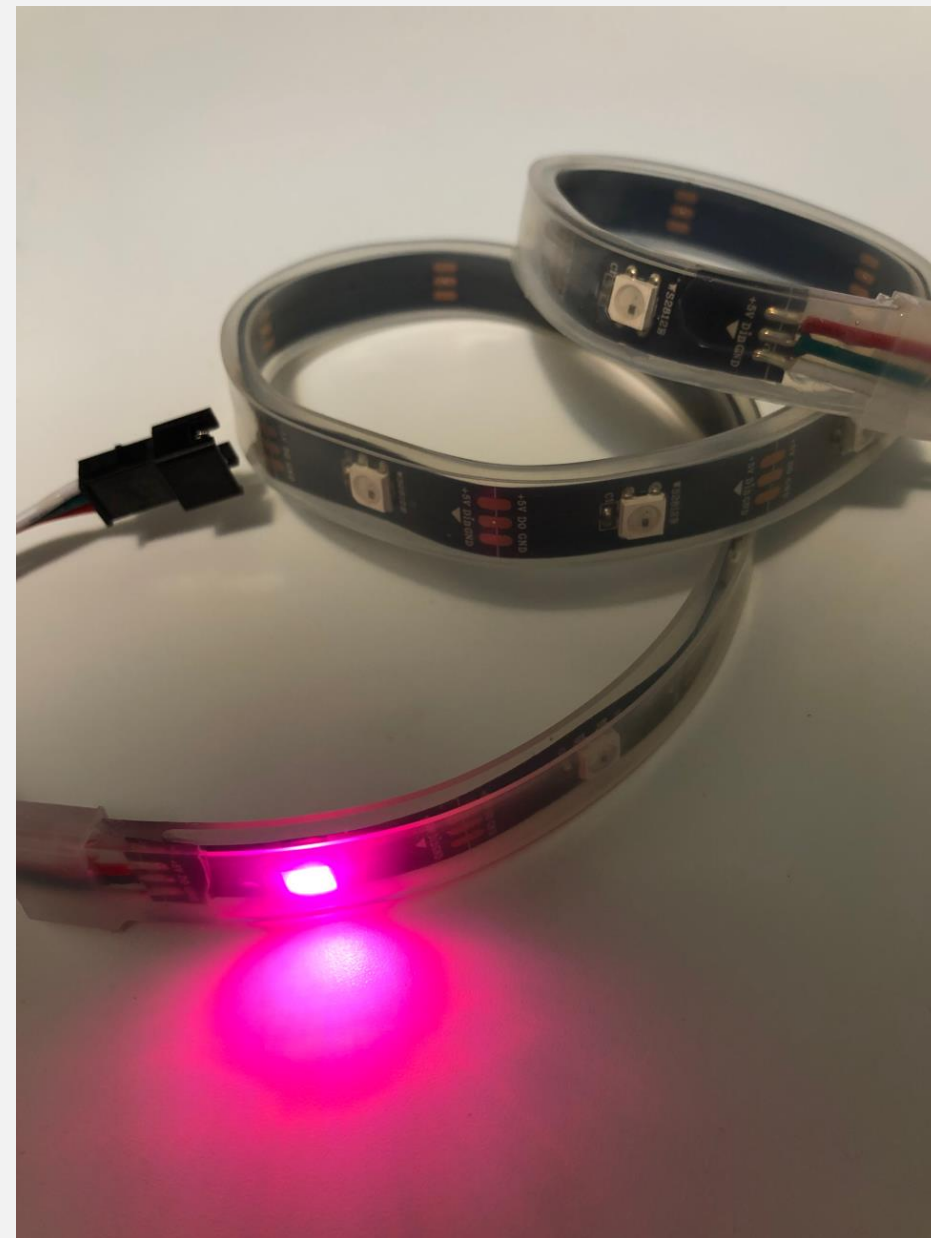
LAB04 心情日記實驗結果



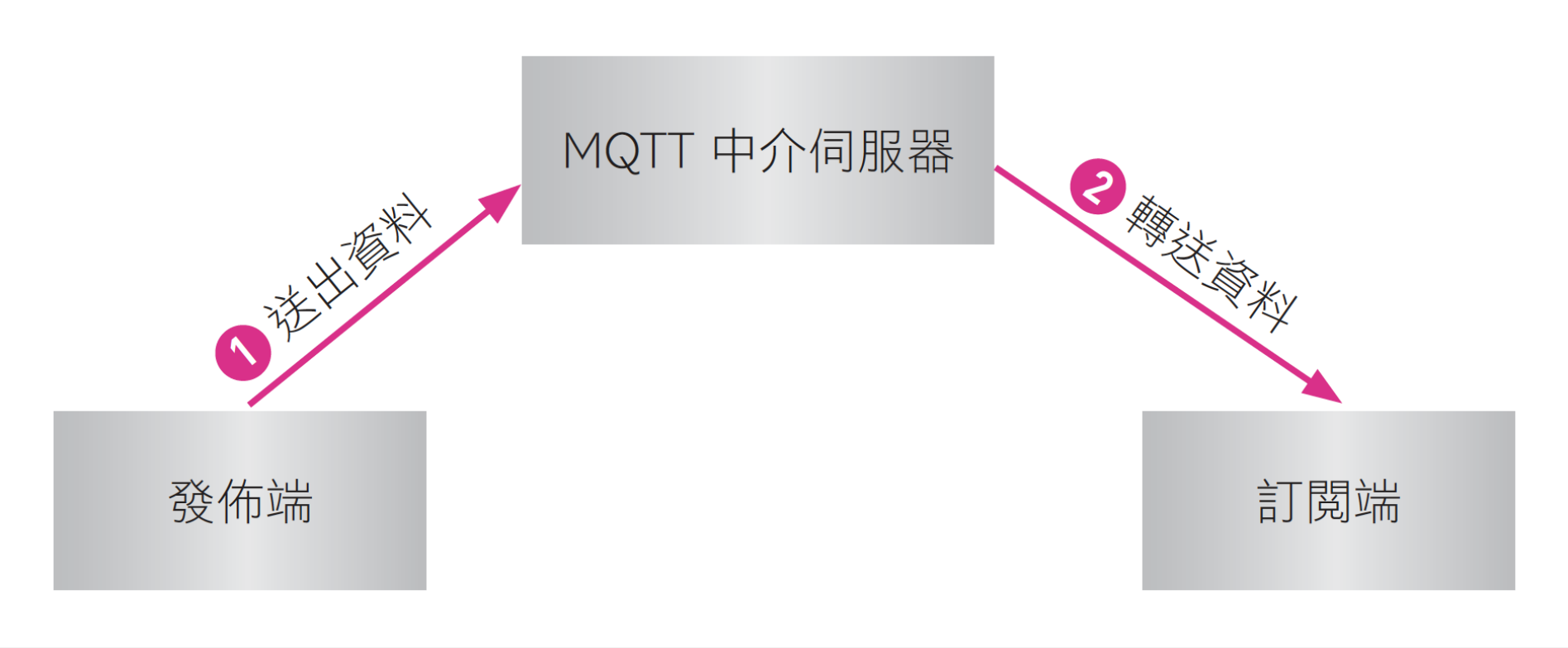
LAB05 心情分享氣氛燈

使用 MQTT 通訊讓控制板可以收到辨識的結果,再對應到不同的 LED 條燈特效

當你想要跟你的朋友或家人分享你的心情時,但沒有很迫切需要對方知道,就可以在親人或伴侶那邊放置一個心情氣氛分享燈,只要你連線到網頁露出表情並按下分享按鈕,心情分享氣氛燈就會將你的心情傳達給對方,透過 LED 燈的特效,讓朋友或家人即時感受到相同的氣氛。



MQTT 訊息協定



WIFI 連線

要使用網路，首先必須匯入 **network** 模組，利用其中的 **WLAN** 類別建立控制無線網路的物件：

```
>>> import network
>>> sta_if = network.WLAN(network.STA_IF)
```

網路介面	說明
network.STA_IF	工作站 (station) 介面，專供連上現有的 Wi-Fi 無線網路基地台，以便連上網際網路
network.AP_IF	熱點 (access point) 介面，可以讓 D1 mini 變成無線基地台，建立區域網路

啟用並連接 WI-FI 無線網路

由於我們需要讓 D1 mini 連上網際網路擷取資訊，所以必須使用**工作站介面**。取得無線網路物件後，要先啟用網路介面：

```
>>> sta_if.active(True)
```

參數 True 表示要啟用網路介面；如果傳入 False 則會停用此介面。接著，就可以嘗試連上無線網路：

```
>>> sta_if.connect('無線網路名稱', '無線網路密碼')
```

確認是不是有連上網路：

```
# 檢查連線狀態，還沒連上就繼續跑迴圈  
while not wifi.isconnected():  
    pass
```

什麼都不做的動作
單純為了合乎語法

Python 的資料結構 (容器)

Python 的資料結構

- 字串容器：由字元組成。例如：`string = "52python"`。
- tuple 容器：由資料物件組成。例如：`tuple = (1, '2', 3)`。
- 串列容器：由資料物件組成。例如：`list = [1, '2', 3]`。
- 集合容器：由資料物件組成。例如：`set = {1, '2', 3}`。
- 字典容器：由資料物件組成，以鍵：值表示。例如：`dick = {'A':1, 'B':'2', 'C':3}`。

UMQTT 模組

在 MicroPython 中內建有 umqtt 模組，提供 MQTT 用戶端的功能，使用時必須先匯入其中的 MQTTClient 類別：

```
from umqtt.robust import MQTTClient
```

接著建立物件：

```
client = MQTTClient(  
    client_id="D1mini",  
    server="io.adafruit.com",  
    user="AIO 帳號",  
    password="AIO 金鑰",  
    ssl=False)
```

其中 client_id 是用戶端識別名稱，使用同一帳號連上伺服器的個別裝置都要指定不一樣的名稱。

UMQTT 模組

umqtt 模組提供用戶端向 MQTT 中介伺服器訂閱頻道的功能，首先要準備一個收到訂閱資料時會被自動呼叫的函式，例如：

```
def get_cmd(topic, msg):  
    print(topic,msg)
```

定義好函式後，還要將該函式註冊為收到訂閱資料時的處理函式：

```
client.set_callback(get_cmd)
```

接著就可以向伺服器訂閱頻道，AIO 的頻道格式固定為 "帳號 /feeds/feed 名稱"，在本例中 feed 名稱為 mood：

```
client.user.encode() + b"/feeds/mood"
```

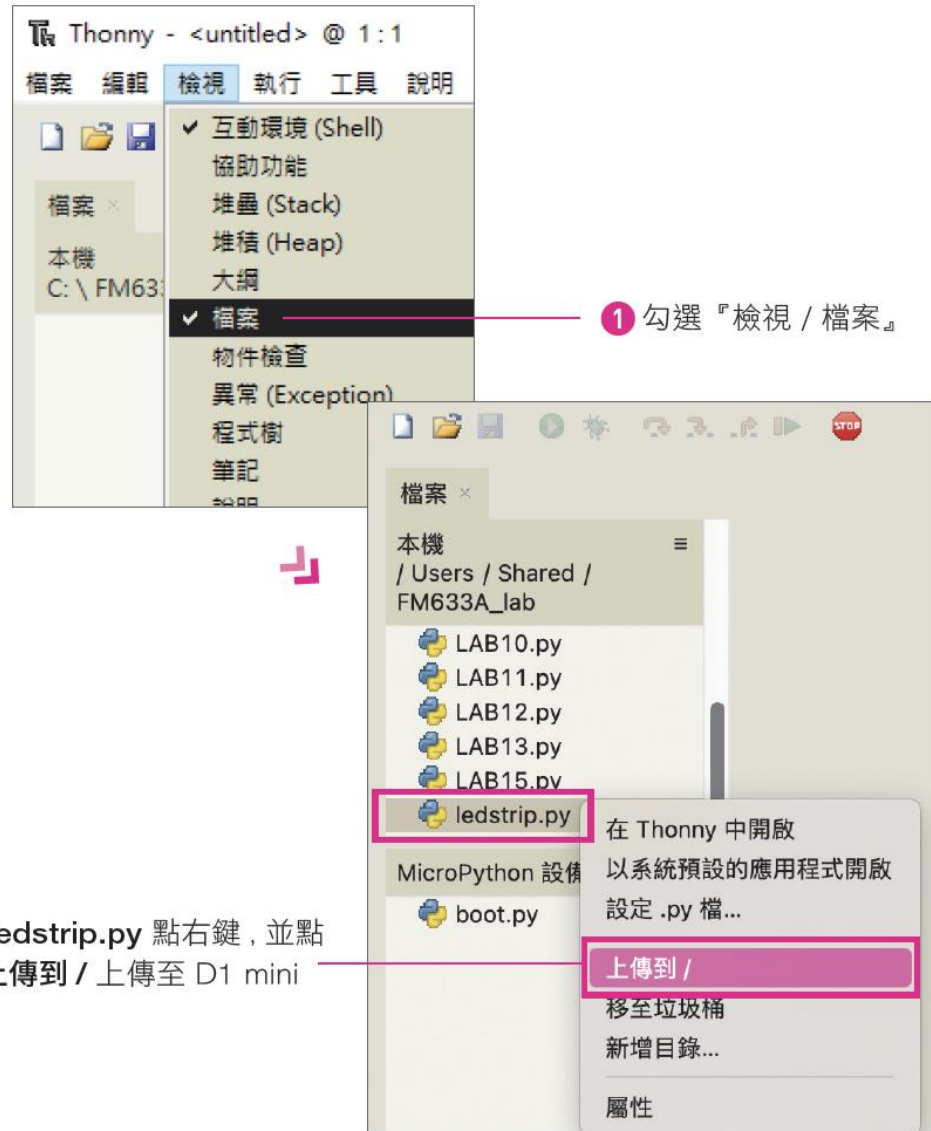


UMQTT 模組

最後還有一個關鍵的步驟，就是要不斷檢查是否有新的資料：

```
while True:  
    client.check_msg()
```

上傳 LED 特效模組到 D1 MINI



LEDSTRIP 模組

- **rainbow_cycle (ms)**

所有燈珠以漸層彩虹顏色流動顯示，而參數 **ms** 表示每種顏色變換的間隔毫秒時間。

- **cycle (r, g, b, ms)**

單顆燈珠以指定顏色跑馬燈效果，從第一顆亮到最後一顆停止，參數 **rgb** 為紅綠藍三色分別從 **0** 至 **255** 的輸出數值，值越大亮度也越高，參數 **ms** 表示跑馬燈過程中每顆燈與燈亮滅的間隔時間。

- **bounce(r, g, b, ms)**

與 **cycle** 特效相似，不同在於跑馬燈跑到最後一顆燈珠後，會折返跑向第一顆後才停止。

範例程式：LAB05.PY

SSID : wifi 帳號
Password: wifi 密碼

lab05.py ×

```

9
10 # 連線到無線網路
11 sta_if = network.WLAN(network.STA_IF)
12 sta_if.active(True)
13 sta_if.connect("無線網路基地台", "無線網路密碼")
14
15 # 循環測試網路直到網路連線成功
16 while not sta_if.isconnected():
17     pass
18 print("控制板已連線")
19
20 # 建立 MQTT 客戶端物件
21 client = MQTTClient(
22     client_id="D1mini",
23     server="io.adafruit.com",
24     user="AIO 帳號",
25     password="AIO 金鑰",
26     ssl=False)
27
28 # 註冊收到訂閱資料時的處理函式

```



或 F5

互動環境 (Shell) ×

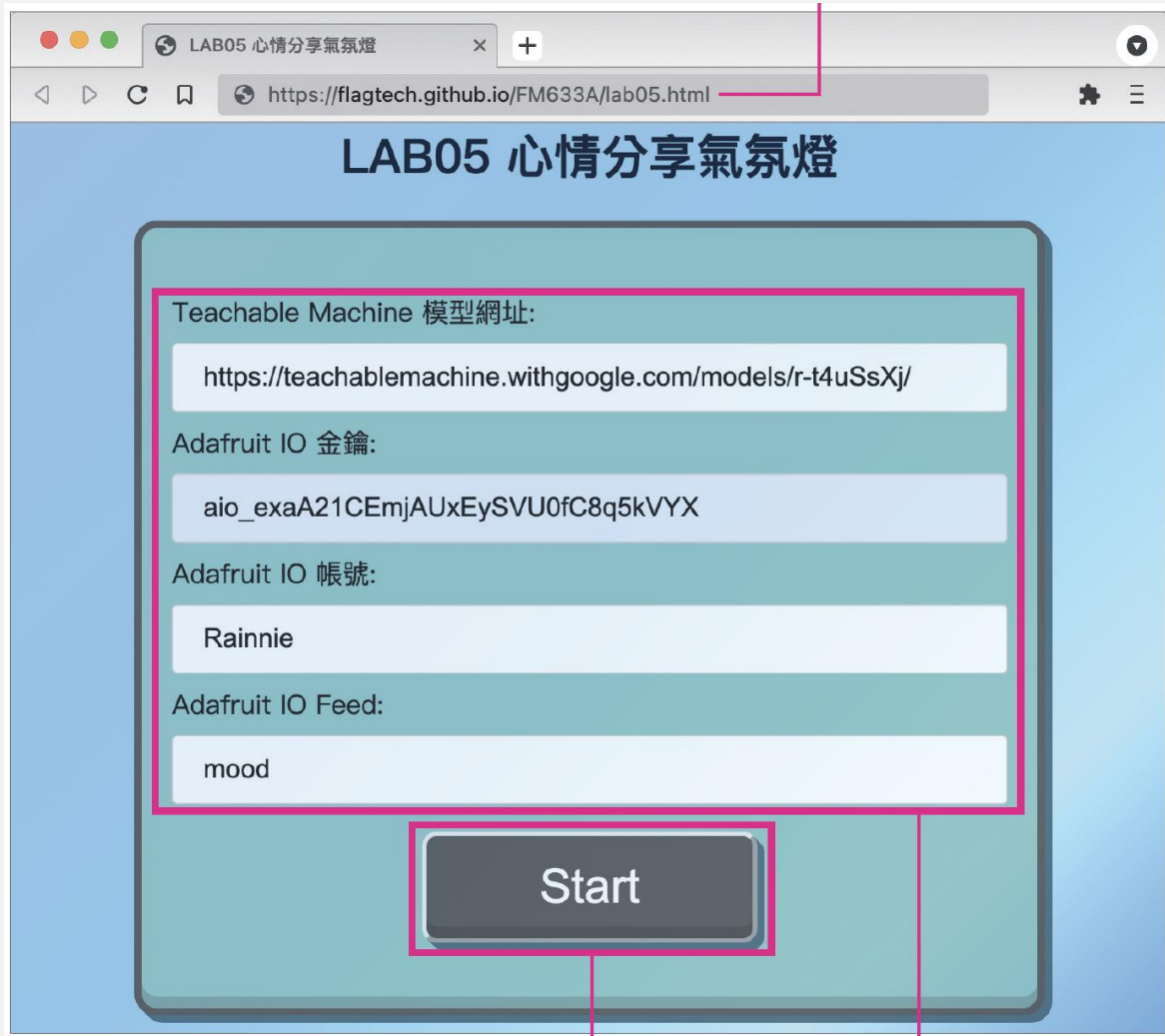
```

MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

```

控制板已連線

https://flagtech.github.io/FM633A/lab05.html



按 Start

填入各項資訊

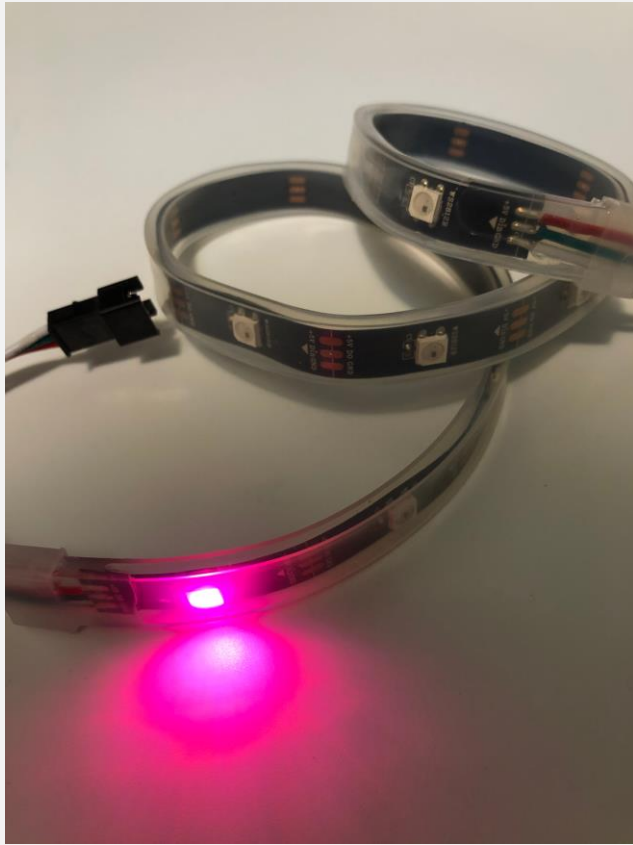
LAB05 心情日記實驗結果

您看起來有77%心情是 生氣



分享心情

按此分享心情



LAB05.PY 程式解說

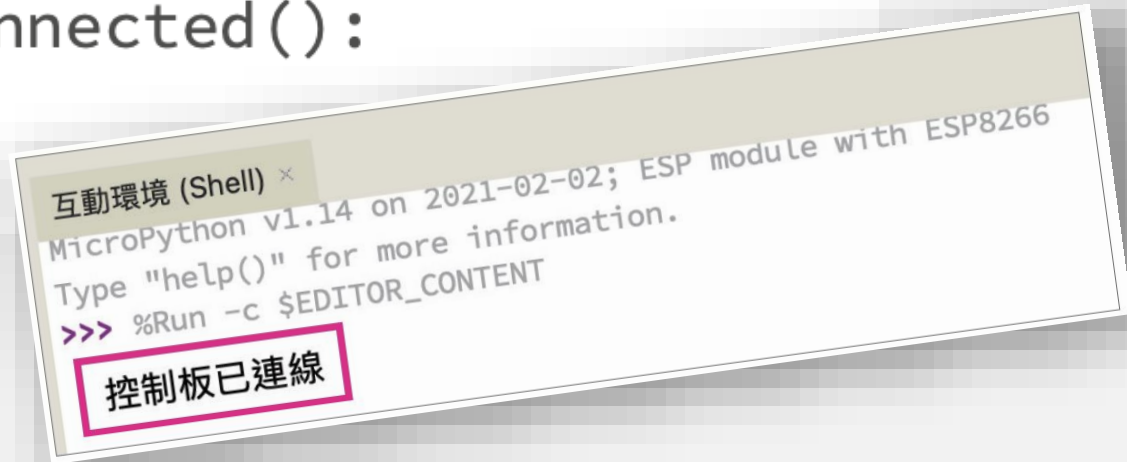
lab05.py ×

```
1 import network
2 import ledstrip
3 from umqtt.robust import MQTTClient
4
5 #建立變數來存放條燈特效類別
6 led_strip_effect = b'0'
7 #設定條燈腳位為 4，燈珠數量為 15
8 ledstrip.setup(4,15)
9
```

我們希望程式接收到下一筆資料前，都重複目前的特效，所以需要建立一個變數來存放代表特效的編號（其實就是辨識結果）：

LAB05.PY 程式解說

```
10 # 連線到無線網路
11 sta_if = network.WLAN(network.STA_IF)
12 sta_if.active(True)
13 sta_if.connect("無線網路基地台", "無線網路密碼")
14
15 # 循環測試網路直到網路連線成功
16 while not sta_if.isconnected():
17     pass
18 print("控制板已連線")
```



LAB05.PY 程式解說

```
20 # 建立 MQTT 客戶端物件
21 client = MQTTClient(
22     client_id="D1mini",
23     server="io.adafruit.com",
24     user="AIO 帳號",
25     password="AIO 金鑰",
26     ssl=False)
27
28 # 註冊收到訂閱資料時的處理函式
29 def get_cmd(topic, msg):
30     # 宣告使用全域變數 led_strip_effect
31     global led_strip_effect
32     print(topic,msg)
33     # 設定條燈特效項目
34     led_strip_effect = msg
```

5 #建立變數來存放條燈特效類別
6 led_strip_effect = b'0'

LAB05.PY 程式解說

```
40 # 不斷重複執行
41 while True:
42     client.check_msg()
43     if led_strip_effect == b'0':
44         ledstrip.clear()
45     elif led_strip_effect == b'1':
46         #rainbow_cycle(間隔毫秒時間)
47         ledstrip.rainbow_cycle(5)
48     elif led_strip_effect == b'2':
49         #cycle(r, g, b, 間隔毫秒時間)
50         ledstrip.cycle(123, 0, 20, 100)
51     elif led_strip_effect == b'3':
52         #bounce(r, g, b, 間隔毫秒時間)
53         ledstrip.bounce(23, 20, 128, 50)
```

課程大綱

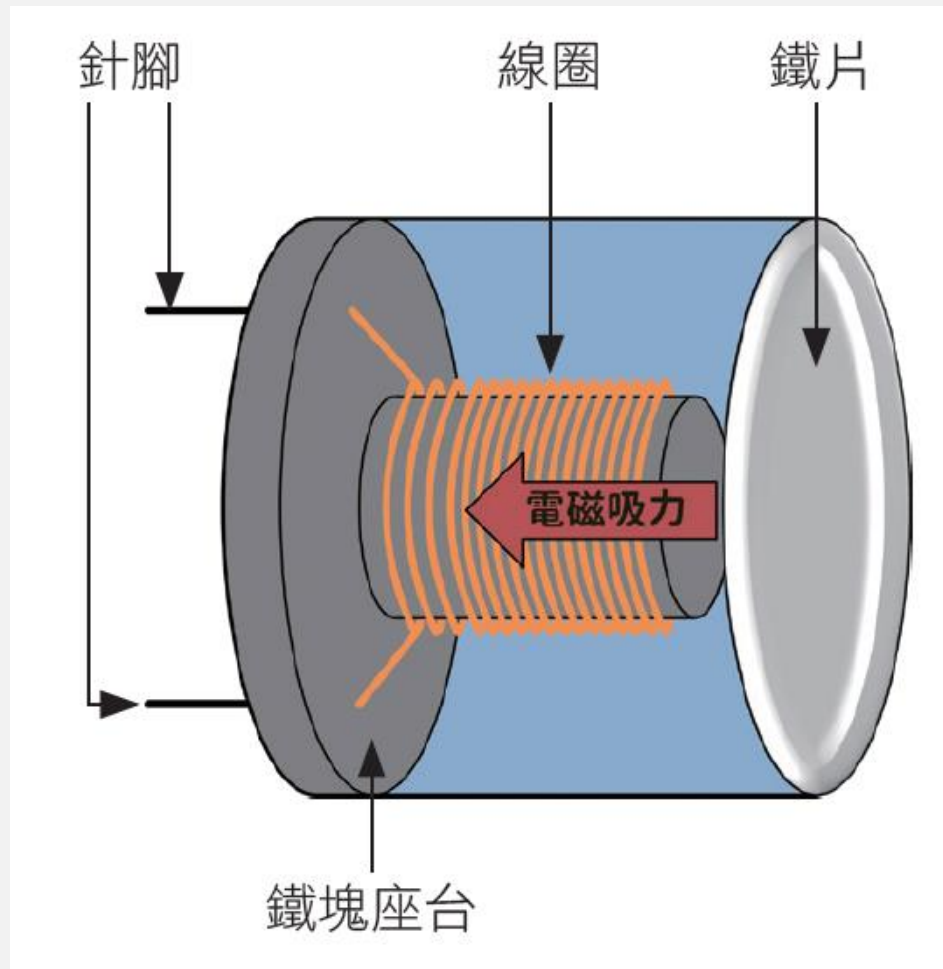
- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



04 影像辨識分類大應用

從建立模型、上傳辨識結果再讓控制板根據結果輸出不同聲光效果

無源蜂鳴器



控制吸放鐵片頻率即可鼓動空氣發出聲音
控制吸放的方式就跟控制 LED 亮滅一樣

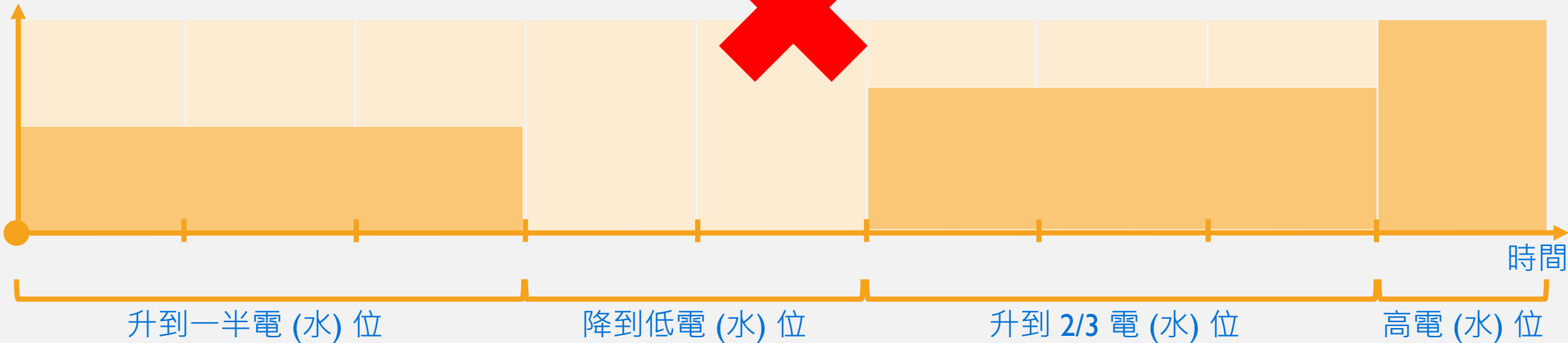
數位輸出不能控制大小

數位輸出只有高電位與低電位
沒有高、低電位間的中間大小

```
>>> from machine import Pin
>>> led = Pin(2, Pin.OUT)
>>> led.value(1)
>>> led.value(0)
>>> led.value(0.2) ← 猜猜會怎樣？
>>> led.value(30)
>>> led.value("hello")
>>> led.value("")
```

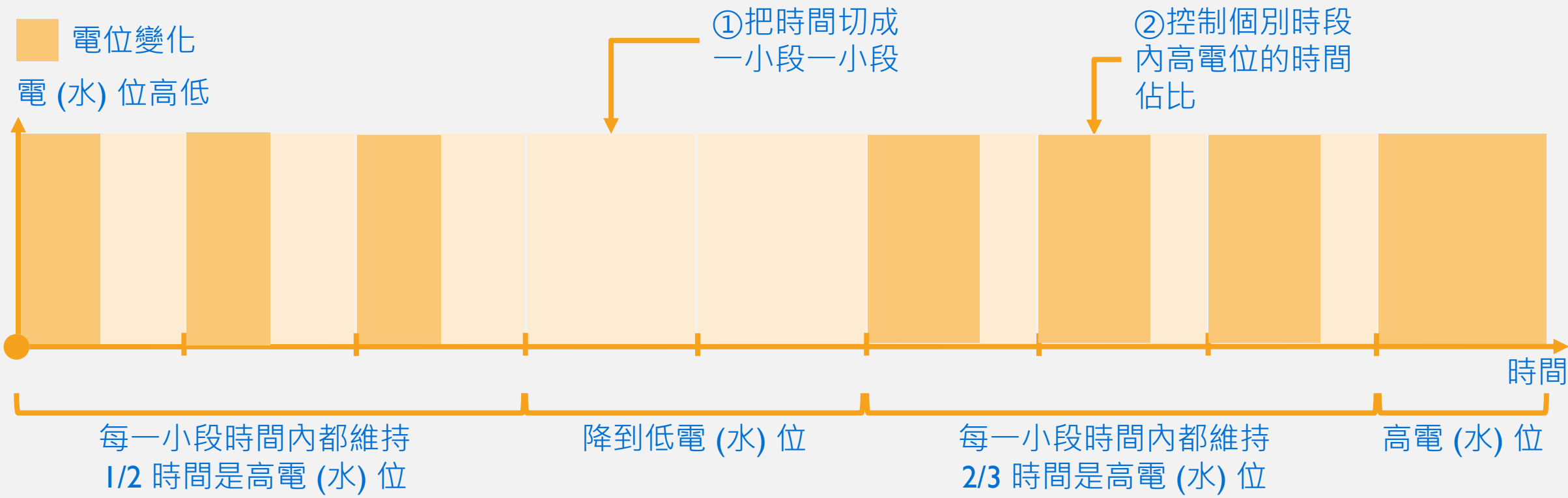
其實這裡只能給 True (是) / False (否)
Python 把 0、空字串 "、"" 當成 False、其餘為 True

電位變化
電 (水) 位高低



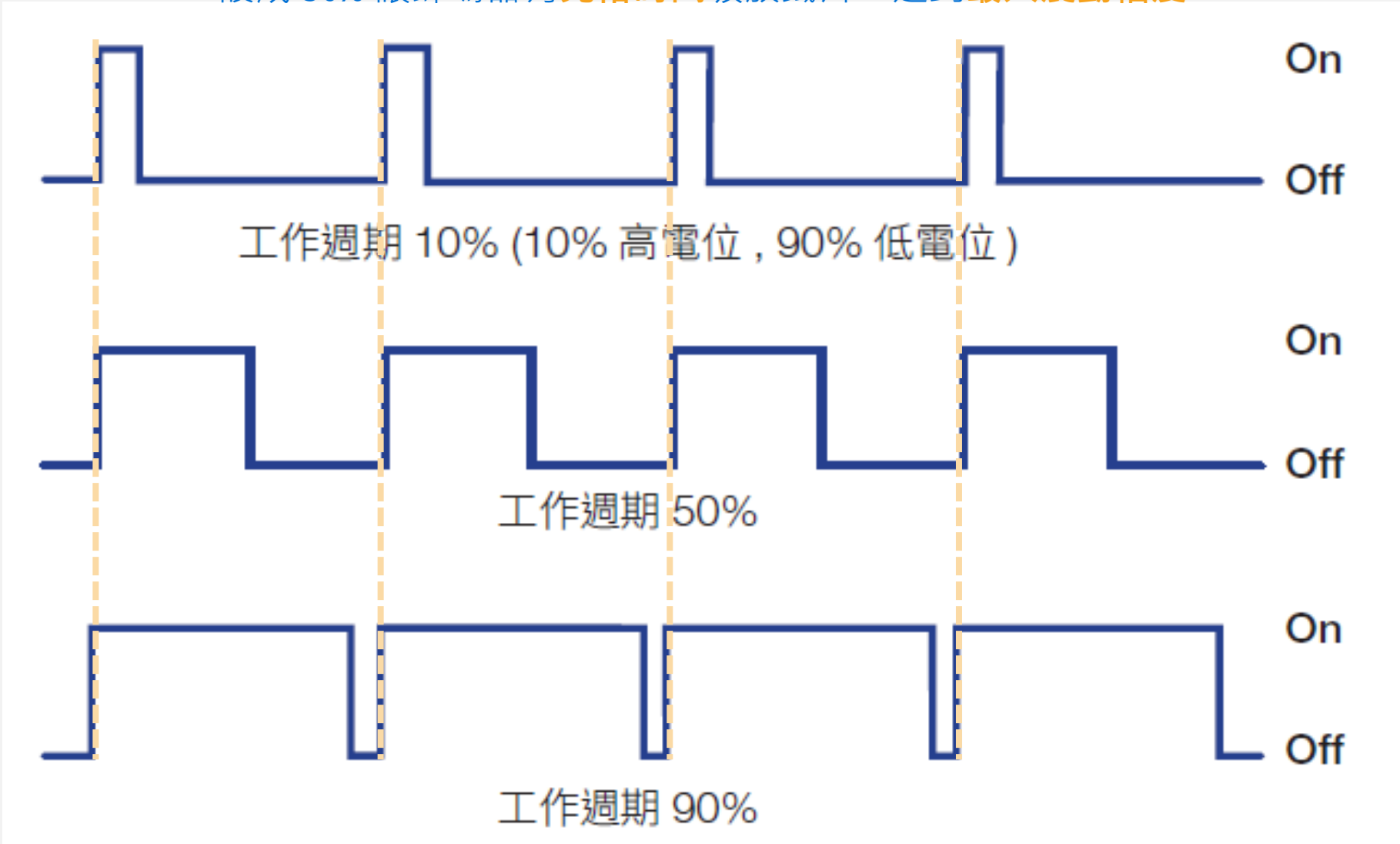
用時間控制大小變化的 PWM

原本→想直接控制高低電位間變化
改成→控制高、低電位的持續時間
就像→用踩腳踏車板時間控制速度
稱 **PWM (Pulse Width Modulation)**



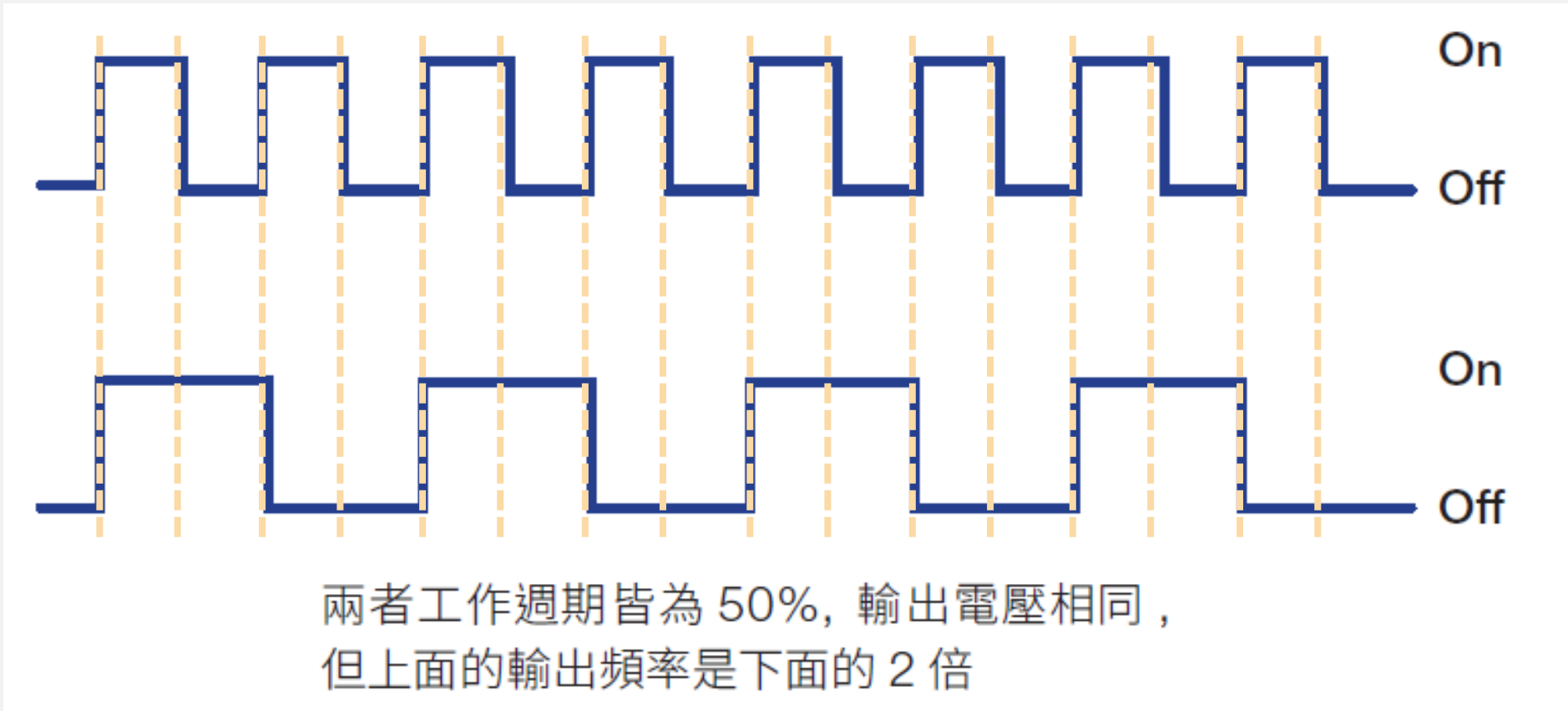
DUTY CYCLE 控制 PWM 產生的大小變化

工作週期 (duty cycle) 是單一時間切片內高電位佔的時間比例
設成 50% 讓蜂鳴器有充裕時間吸放鐵片、達到最大震動幅度



FREQUENCY (頻率) 控制 PWM 的變換速度

頻率 (frequency) 控制要將時間切成多細
對蜂鳴器來說就可以控制發出的聲音高低



兩者工作週期皆為 50%，輸出電壓相同，
但上面的輸出頻率是下面的 2 倍

MICROPYTHON 中的 PWM

```
from machine import Pin, PWM # 匯入 Pin 和 PWM 子函式庫
```

```
buzzer = PWM(Pin(15, Pin.OUT))
```

```
buzzer.freq(0)
```

頻率 1~1000Hz

```
buzzer.duty(512)
```

工作週期 0~1023 -> 0%~100%

上面這三行程式也能合併成一行：

```
buzzer = PWM(Pin(15, Pin.OUT), freq=0, duty=512)
```

無源蜂鳴器聲音輸出

音名	中音C	D	E	F	G	A	B
唱名	Do	Re	Mi	Fa	Sol	La	Ti
頻率 (Hz)	261	294	330	349	392	440	494

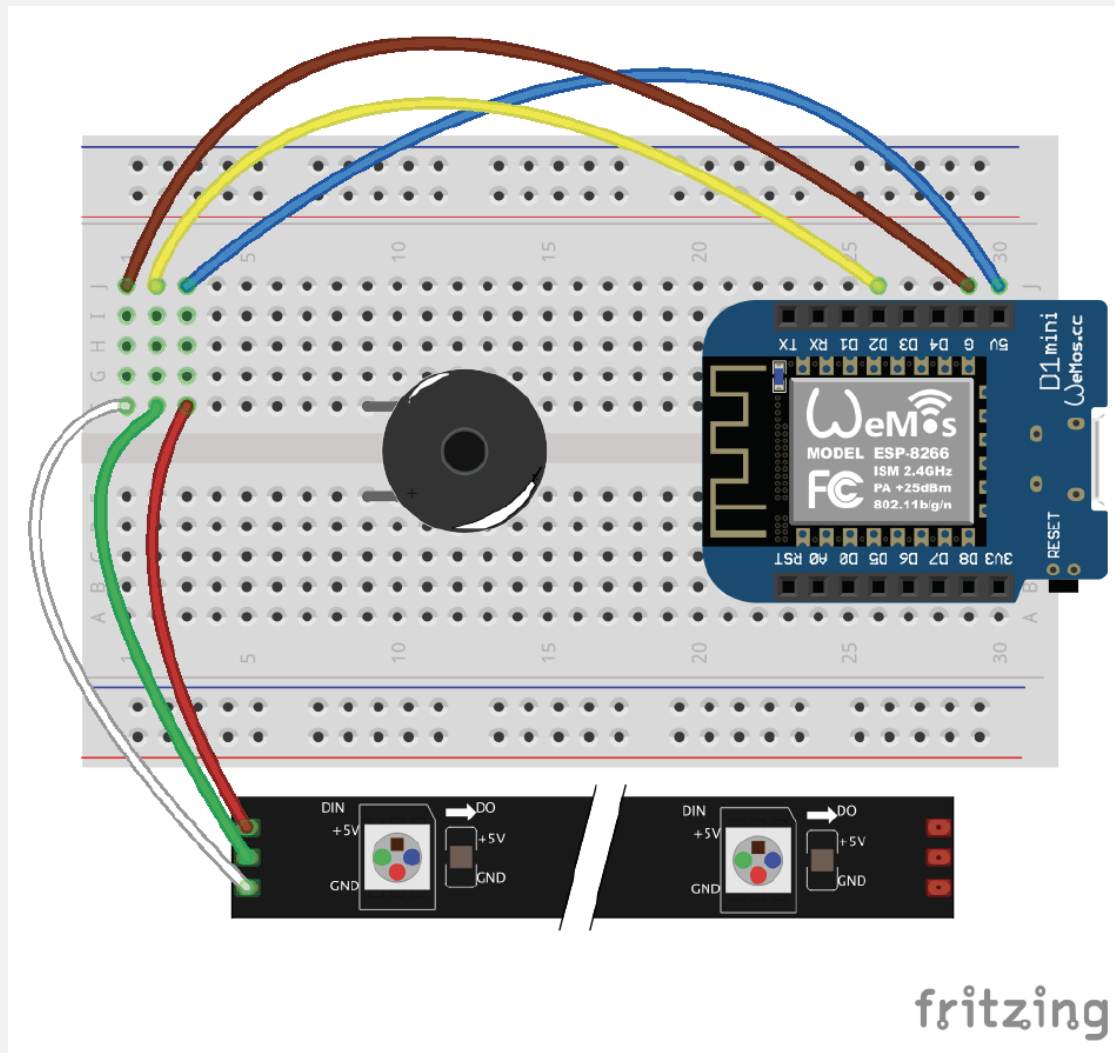
LAB06 歡樂音樂盒

讓蜂鳴器發出特定頻率, 組合出旋律播放小蜜蜂

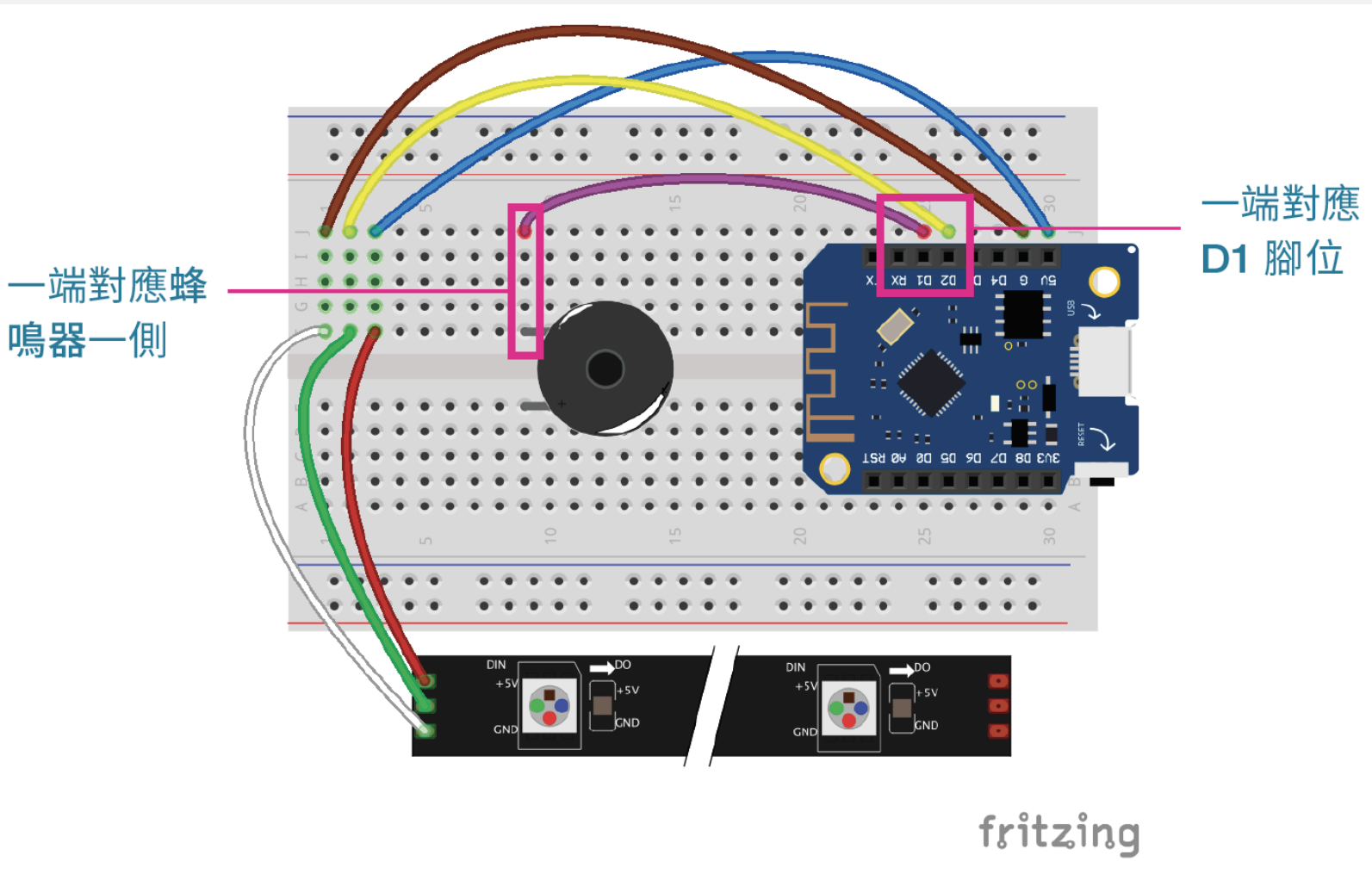
音樂能夠陶冶性情, 接著就來做個電子音樂盒讓自己的心靈沉澱一下吧!



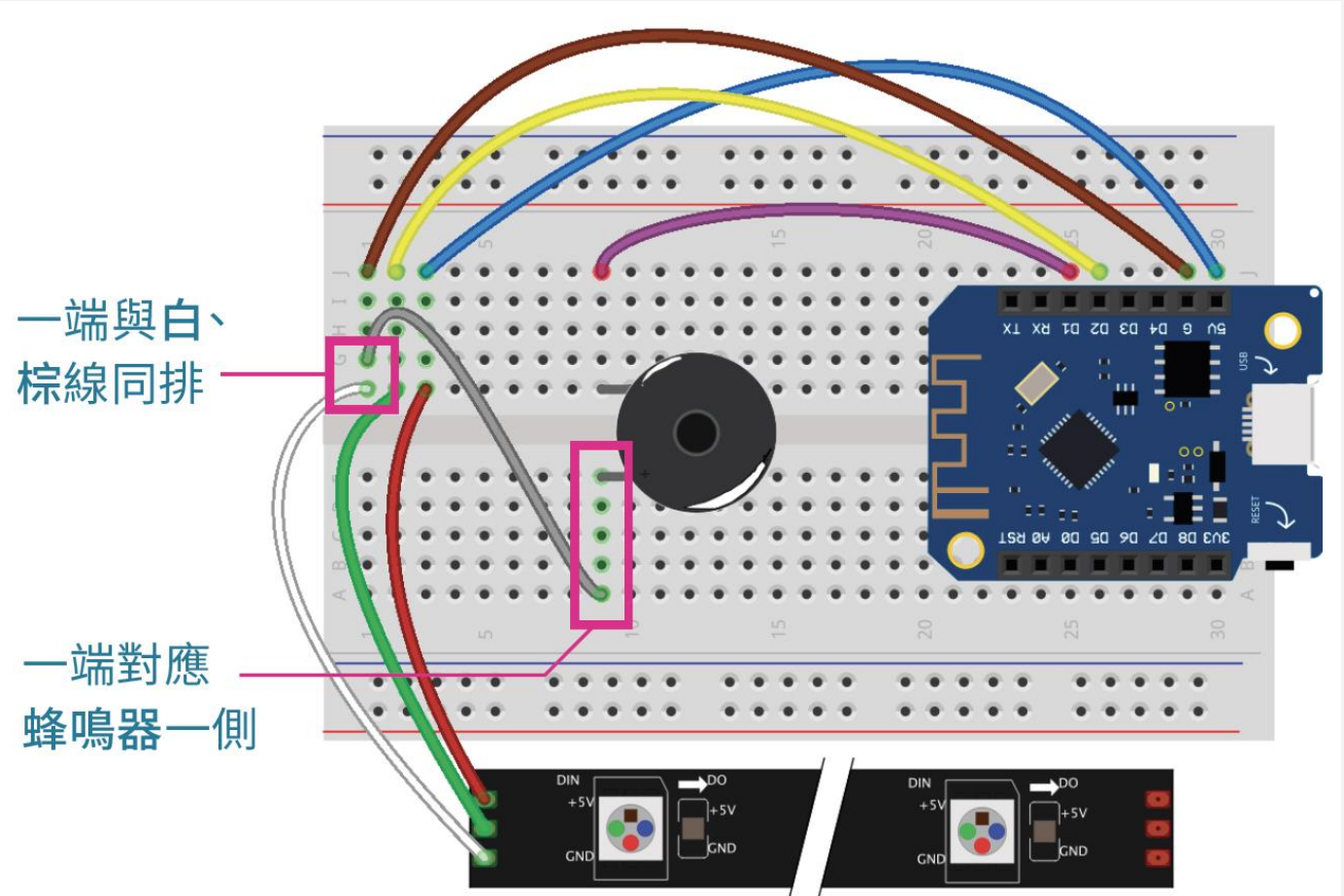
插上蜂鳴器：插在中間、一腳一邊



使用紫色杜邦線：一端蜂鳴器一端 D1



使用灰色杜邦線



範例程式：LAB06.PY

```
lab06.py ×  
1 from machine import Pin, PWM  
2 import time  
3  
4 buzzer = PWM(Pin(5))  
5 buzzer.duty(512)  
6  
7 buzzer.freq(392)  
8 time.sleep(.5)  
9 buzzer.freq(330)  
10 time.sleep(.5)  
11 buzzer.freq(330)  
12 time.sleep(1.)  
13 buzzer.freq(349)  
14 time.sleep(.5)  
15 buzzer.freq(294)
```



或 F5

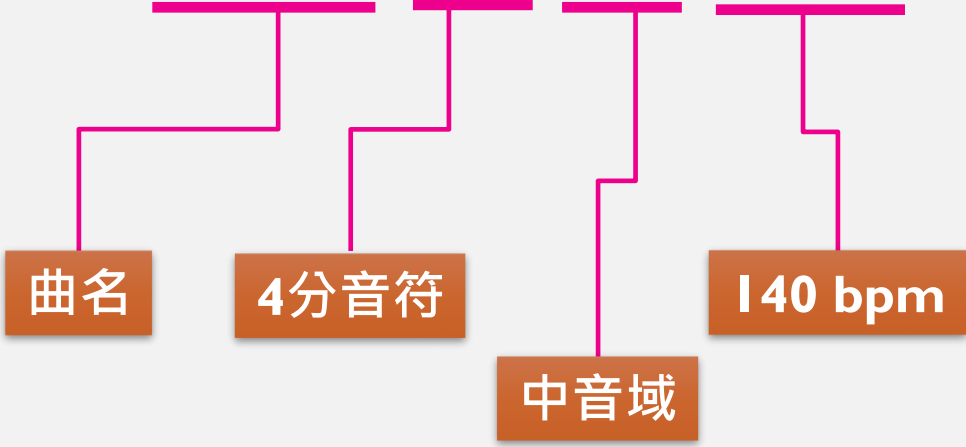
一直輸入頻率跟間隔時間也太辛苦

RTTTL (Ring Tone Text Transfer Language)



RTTTL

```
tune = RTTTL("winner:d=4,o=5,b=140:l6e6,l6e6,32p,8e6,l6c6, ...")
```



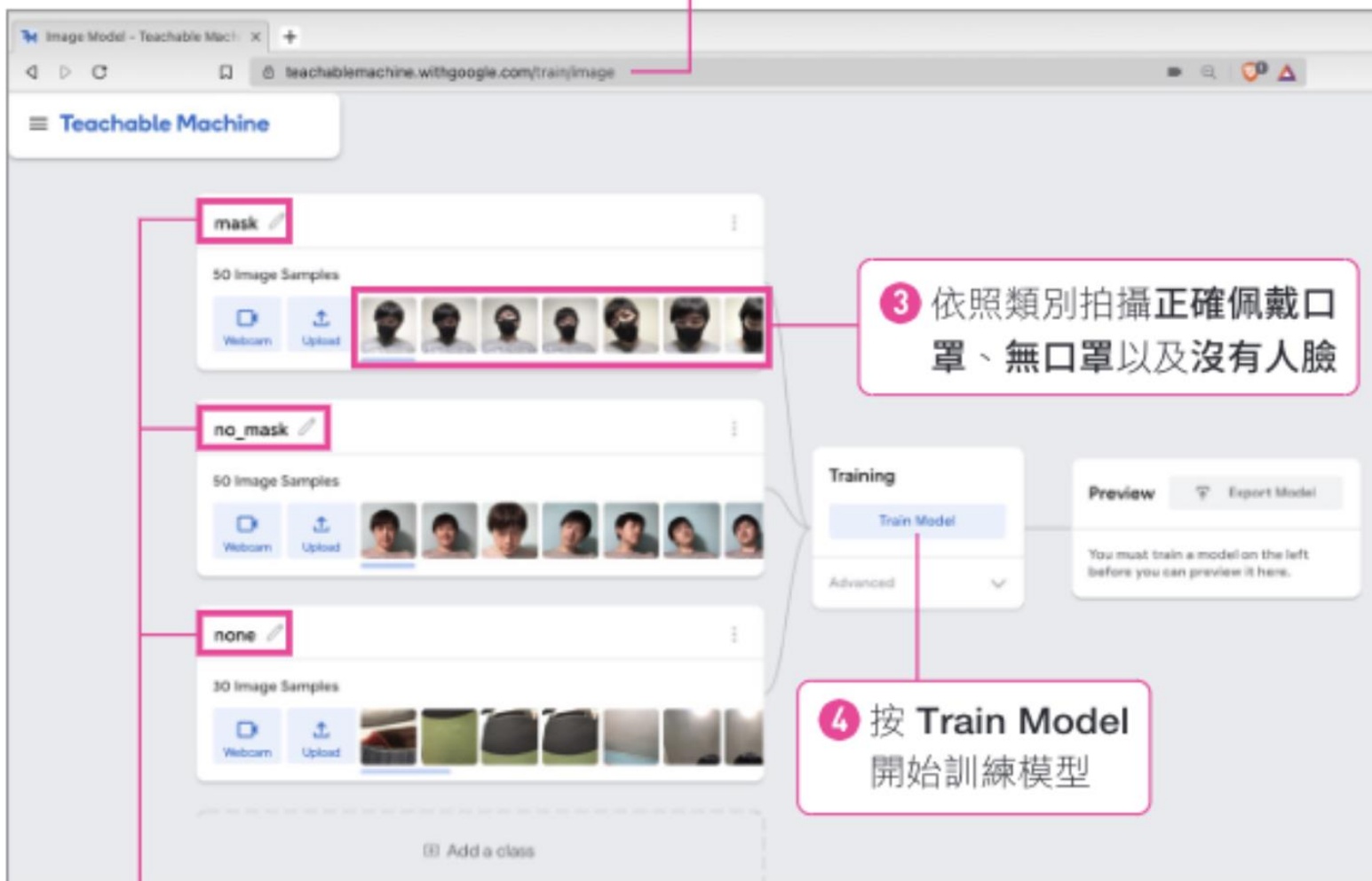
LAB07 口罩偵測警報系統

根據辨識結果判斷是否需要發出警報, 若需要發出警報時, 傳送特定數值100 至 AIO 資料庫, 再由控制板讓蜂鳴器發

當傳染疾病盛行的時期, 各公共場合都會要求民眾佩戴口罩, 若是有台機器可以幫我們辨識進出的民眾有無配戴口罩, 就可以省下許多人力資源了。



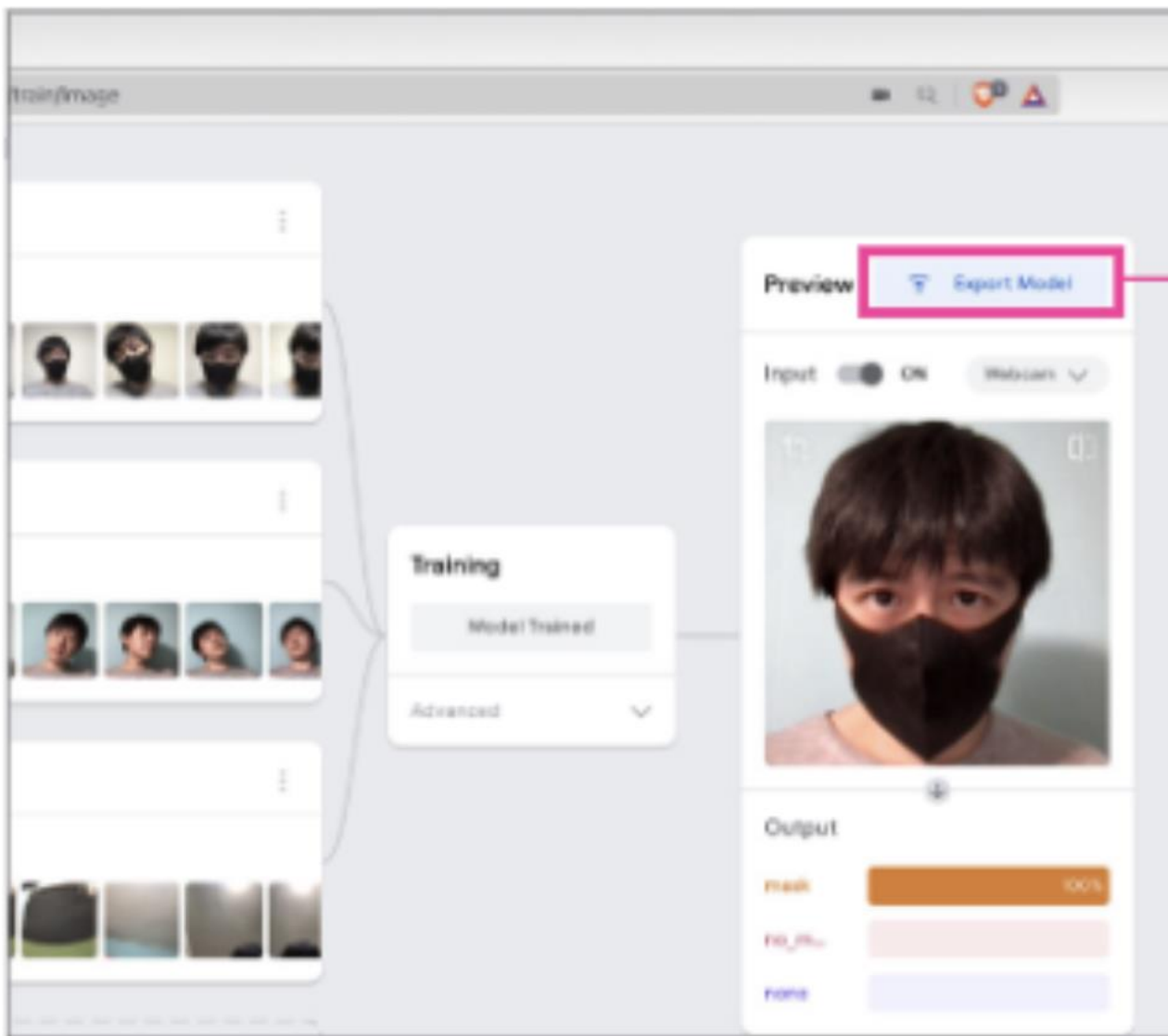
1 連線 <https://teachablemachine.withgoogle.com/train/image>



3 依照類別拍攝正確佩戴口罩、無口罩以及沒有人臉

4 按 Train Model 開始訓練模型

2 新增並修改類別名稱為：mask、no_mask、none (注意字母皆為小寫)



5 按此輸出模型

Export your model to use it in projects. ✕

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow Lite ⓘ

Export your model:

Upload (shareable link) Download Upload my model

6 按此上傳模型至雲端

Your sharable link:

`https://teachablemachine.withgoogle.com/models/[...]`

When you upload your model, Teachable Machine hosts it at this link.

Code snippets to use your model:

Javascript p5.js

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

Export your model to use it in projects. ✕

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow Lite ⓘ

Export your model:

Upload (shareable link) Download Update my cloud model

Your sharable link:

`https://teachablemachine.withgoogle.com/models/Jpt5XnbqF/` 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.

確認上傳完成

按此可複製網址

https://io.adafruit.com

The screenshot shows the Adafruit IO website interface. At the top, there is a navigation bar with links for Shop, Learn, Blog, Forums, LIVE!, AdaBox, and IO. Below this is a secondary navigation bar with the Adafruit logo and links for Profile, Feeds, Dashboards, Triggers, Services, and My Key. The 'Feeds' link is highlighted with a pink box. Below the navigation, there is a 'Feeds' section with a 'view all' link also highlighted with a pink box. A pink callout box with the number 1 and the text '按 Feeds 連結' points to the 'Feeds' link. Another pink callout box with the number 2 and the text '再點 View All' points to the 'view all' link. On the right side, there is a 'Learn Guides | view all' section with a thumbnail image of a graph and the text 'Adafruit IO Basics: Feeds'.

3 點擊 +New Feed 按鈕

The screenshot shows the 'Create a new Feed' dialog box. At the top left, there are two buttons: '+ New Feed' and '+ New Group'. The '+ New Feed' button is highlighted with a pink box. Below the buttons, there is a 'Default' section with a 'Feed Name' input field. The 'Create a new Feed' dialog box is open, showing a 'Name' input field with the text 'mask' entered. A pink callout box with the number 3 and the text '輸入 mask 再按 Create' points to the 'Name' input field. Below the 'Name' field, it says 'Maximum length: 128 characters. Used: 4'. There is a 'Description' input field below that. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'Create'. The 'Create' button is highlighted with a pink box.

另外再建立後面會用到的所有 feed :

cardgame
effect
music
timecapsule
timer
tw
voicemood
yoga

+ New Feed + New Group

Default

Feed Name	Key
<input type="checkbox"/> cardgame	cardgame
<input type="checkbox"/> effect	effect
<input type="checkbox"/> mask	mask
<input type="checkbox"/> mood	mood
<input type="checkbox"/> music	music
<input type="checkbox"/> timecapsule	timecapsule
<input type="checkbox"/> timer	timer
<input type="checkbox"/> tw	tw
<input type="checkbox"/> voicemood	voicemood
<input type="checkbox"/> yoga	yoga

LAB07.PY 程式解說

```

1  import network
2  from machine import Pin,PWM
3  from time import sleep_ms
4  from umqtt.robust import MQTTClient
5
6  buzzer = PWM(Pin(5))
7  # 設定蜂鳴器發聲頻率
8  buzzer.freq(400)
9  # 建立變數作為警報狀態
10 buzzer_on = 0
11
12 # 連線到無線網路
13 sta_if = network.WLAN(network.STA_IF)
14 sta_if.active(True)
15 sta_if.connect("無線網路基地台", "無線網路密碼")
16
17 # 循環測試網路直到網路連線成功
18 while not sta_if.isconnected():
19     pass
20 print("控制板已連線")
21
22 # 建立 MQTT 客戶端物件
23 client = MQTTClient(
24     client_id="",
25     server="io.adafruit.com",
26     user="AIO 帳號",
27     password="AIO 金鑰",
28     ssl=False)

```

LAB07.PY 程式解說

```

30 # 註冊收到訂閱資料時的處理函式
31 def get_cmd(topic, msg):
32     global buzzer_on
33     print(topic,msg)
34     if msg == b"start":
35         buzzer_on = 1
36         print('NO MASK!')
37     else:
38         buzzer_on = 0
39         print('CLEAR.')
40
41 # 蜂鳴器發聲函式
42 def buzz():
43     buzzer.duty(512)
44     sleep_ms(200)
45     buzzer.duty(0)
46     sleep_ms(200)
47
48 client.connect()
49 client.set_callback(get_cmd)
50 client.subscribe(client.user.encode() + b"/feeds/mask");
51
52 while True:
53     client.check_msg()
54     # 判斷狀態是否需要發聲
55     if buzzer_on :
56         buzz()

```

控制板一樣使用與 Lab05 的 MQTT 訊息協定相同方式與 AIO 資料庫收取訊息, 再根據訊息內容是 100 或是 0 判斷發出或是停止警報

連線 <https://flagtech.github.io/FM633A/lab07.html>

The screenshot shows a web browser window with the following content:

- Browser title: LAB07 口罩偵測警報系統
- Address bar: <https://flagtech.github.io/FM633A/lab07.html>
- Page title: LAB07 口罩偵測警報系統
- Form fields:
 - Teachable Machine 模型網址:
 - Adafruit IO 金鑰:
 - Adafruit IO 帳號:
 - Adafruit IO Feed:
- Start button:

填入各項資訊

按 Start

LAB07 口罩偵測警報系統

100

注意!!! 請佩戴口罩!

此為沒有佩戴口罩 (no_mask) 類別信心數值, 若超過 70% 則發出警報



LAB08 英文圖卡練習機

若辨識結果與題目相同, 就會傳送特定數值到 AIO 資料庫,
再讓控制板以蜂鳴器發出音樂

想學英文又不想死背單字? 讓英文圖卡練習機當你的英文小老師, 隨機出題考考你, 答對還會幫你放音樂。



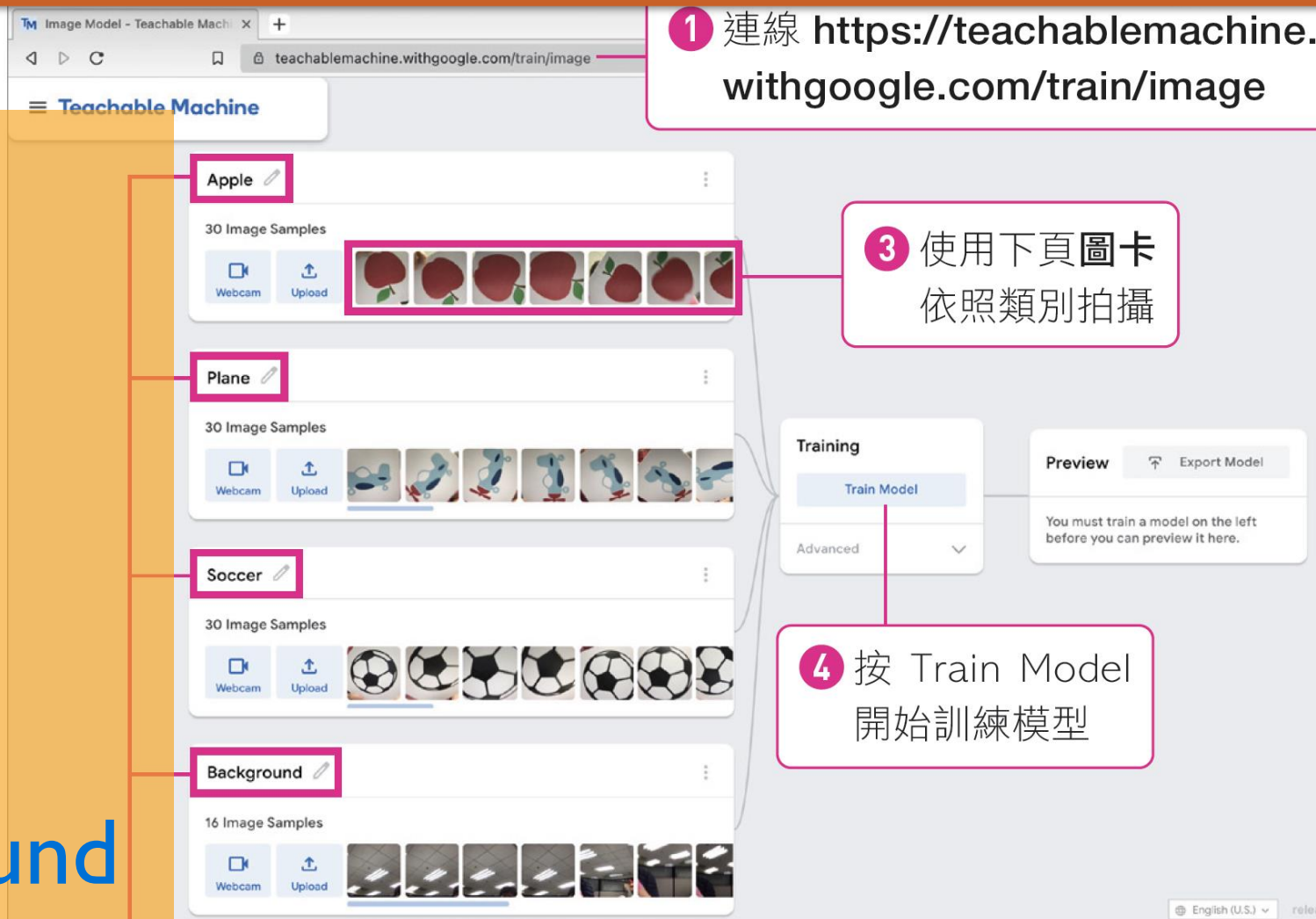
https://teachablemachine.withgoogle.com/train/image

Apple

Plane

Soccer

Background



1 連線 <https://teachablemachine.withgoogle.com/train/image>

3 使用下頁圖卡
依照類別拍攝

4 按 Train Model
開始訓練模型

2 新增並修改類別名稱為：Apple、Plane、Soccer、Background (注意字首需大寫)

Apple

Plane

Soccer

Background



圖卡在手冊 p.47, p48

上傳 RTTTL 到 DI MINI



範例程式：LAB08.PY

```
lab08.py x
18 # 連線到無線網路
19 sta_if = network.WLAN(network.STA_IF)
20 sta_if.active(True)
21 sta_if.connect("無線網路基地台", "無線網路密碼")
22
23
24 # 循環測試網路直到網路連線成功
25 while not sta_if.isconnected():
26     pass
27 print("控制板已連線")
28
29 # 建立 MQTT 客戶端物件
30 client = MQTTClient(
31     client_id="D1mini",
32     server="io.adafruit.com",
33     user="AIO 帳號",
34     password="AIO 金鑰",
35     ssl=False)
36
```



```
互動環境 (Shell) x
MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
控制板已連線
```

https://flagtech.github.io/FM633A/lab08.html

LAB08 英文圖卡練習機

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

填入各項資訊

按 Start

LAB08 英文圖卡練習機實驗結果



LAB08.PY 程式解說

```
37 # 註冊收到訂閱資料時的處理函式
38 def get_cmd(topic, msg):
39     print(topic,msg)
40     if msg == b"100":
41         print('play music')
42         tune = RTTTL("winner:d=4,o=5,b=140:16e6,16e6,32p,8e6")
43         for freq, msec in tune.notes():
44             play_tone(freq, msec)
45     buzzer.duty(0)
```

```
10 def play_tone(freq, msec):
11     if freq > 0:
12         buzzer.freq(freq)
13         buzzer.duty(512)
14         time.sleep(msec*0.001)
15         buzzer.duty(0)
16     time.sleep(0.05)
```

IFTTT 物聯網

IFTTT 是英文 "IF This, Then That" 的縮寫, 其服務的精神就是『如果 A 然後就 B』。我們希望如果資料庫收到資料就發一個 Line 通知給使用者, 這樣的規則稱為一個小程式 (Applet) :



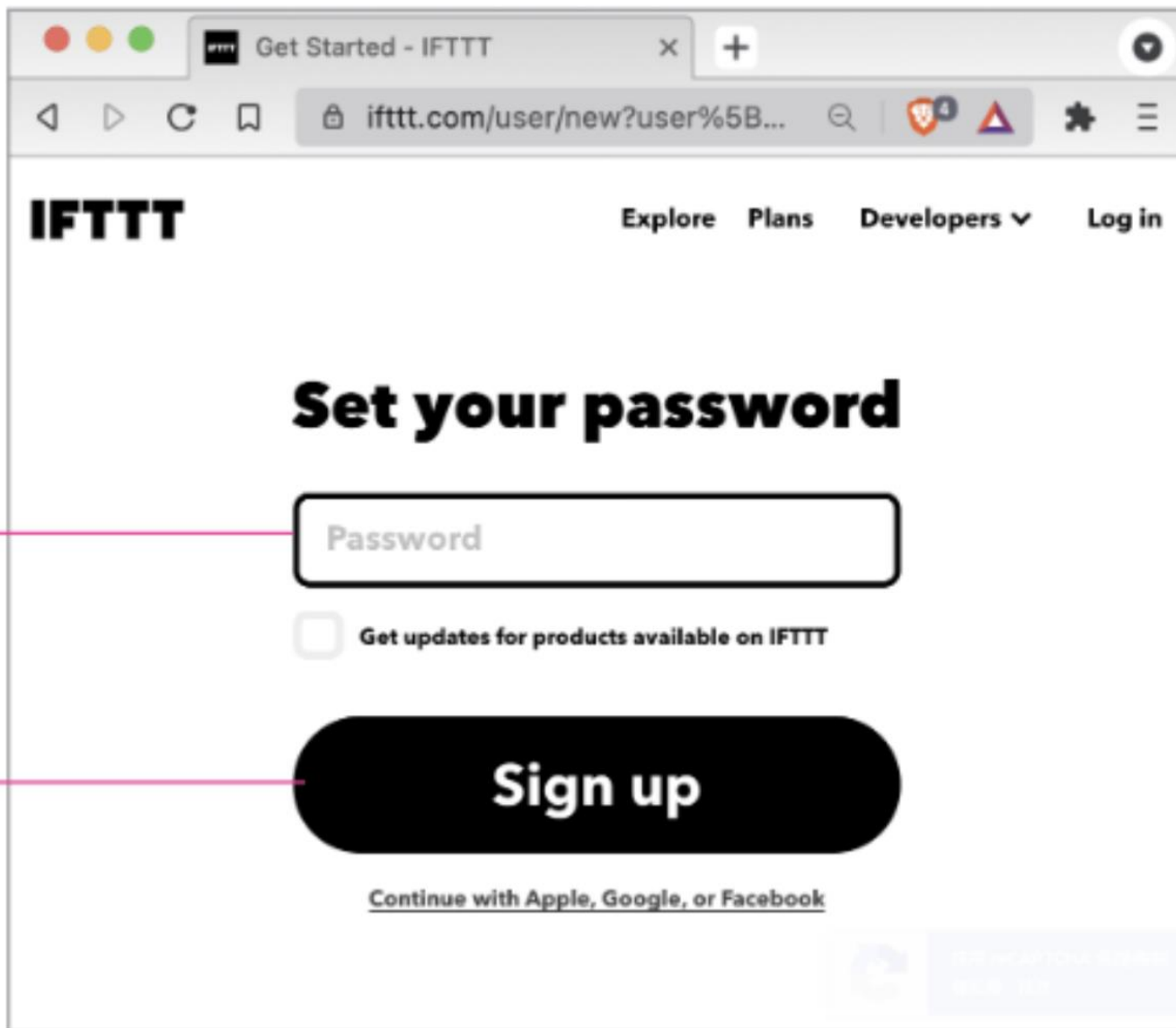
註冊 IFTTT 會員

<https://ifttt.com/>

The screenshot shows the IFTTT website's registration page. At the top, there is a navigation bar with the IFTTT logo, a 'New to IFTTT' button, and links for 'Explore', 'Plans', 'Developers', 'Log in', and 'Try Pro free'. The main content area features the headline 'Do more with the things you love' and a registration form. The form consists of an 'Enter your email' input field and a 'Get started' button. Below the form are buttons for 'Apple', 'Google', and 'Facebook'. The page is decorated with various social media icons in circular frames. Two pink callout boxes with numbered steps are overlaid on the page: '1 輸入 Email 信箱作為會員帳號' points to the email input field, and '2 按 Get started' points to the 'Get started' button.

1 輸入 Email 信箱作為會員帳號

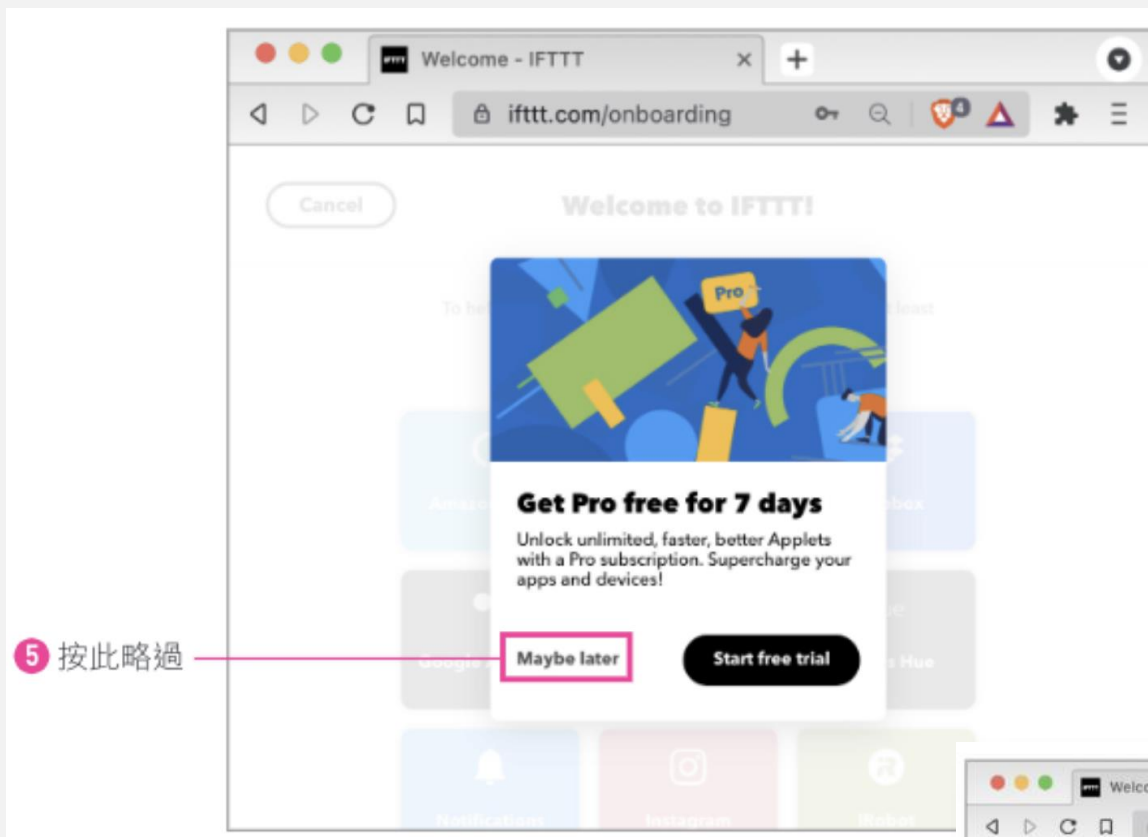
2 按 Get started



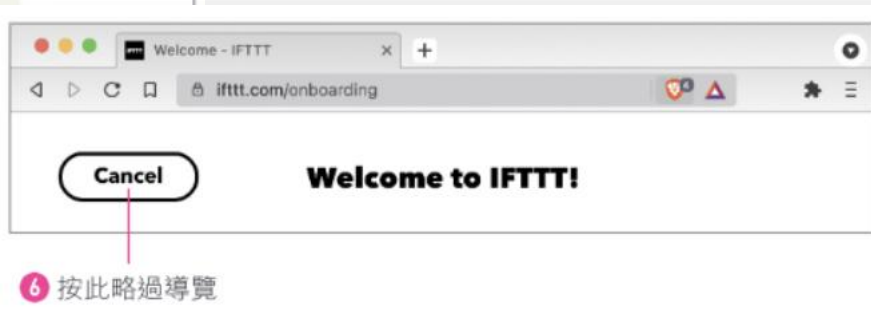
3 設定會員
密碼

4 按 Sign up
完成註冊



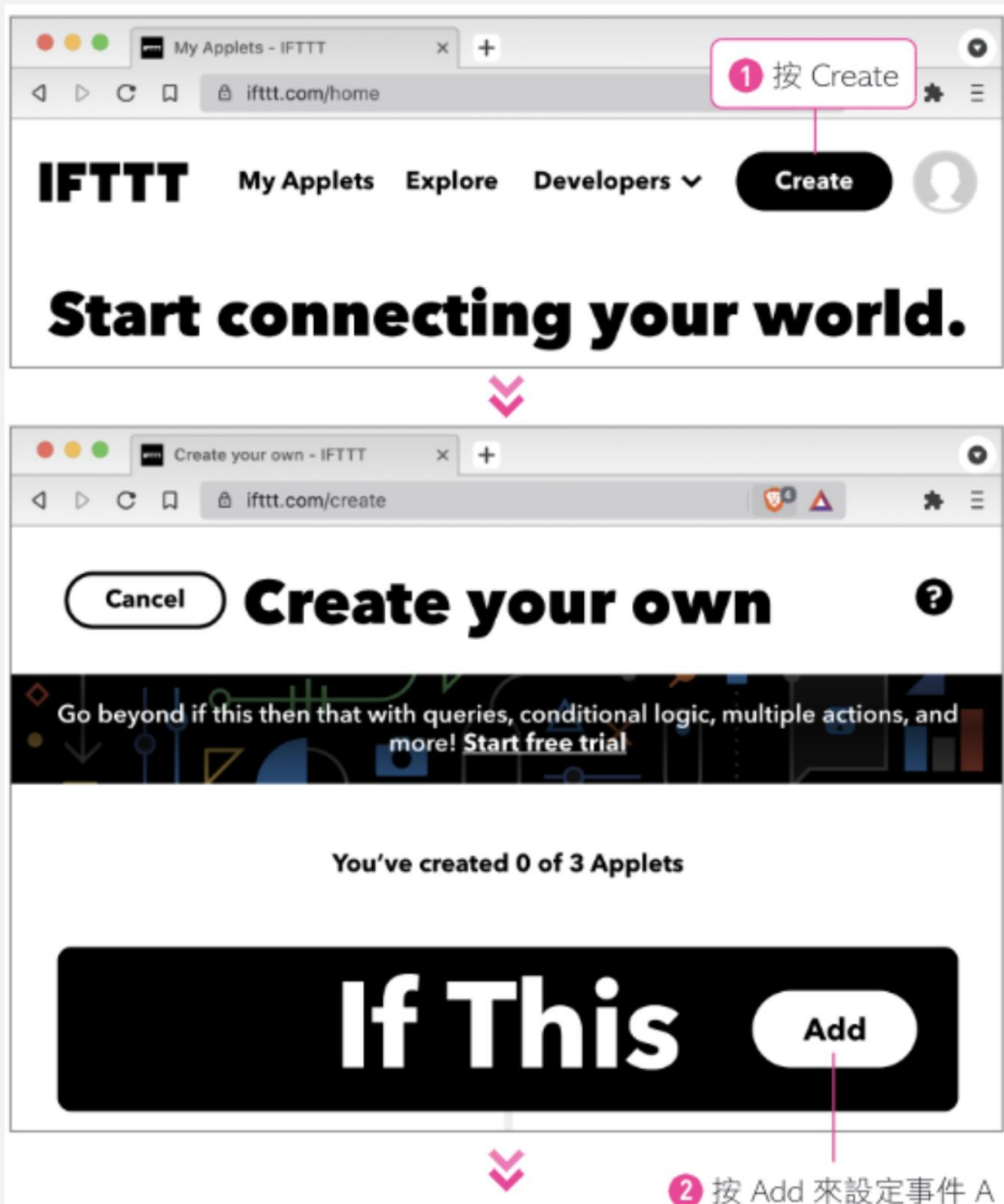


5 按此略過



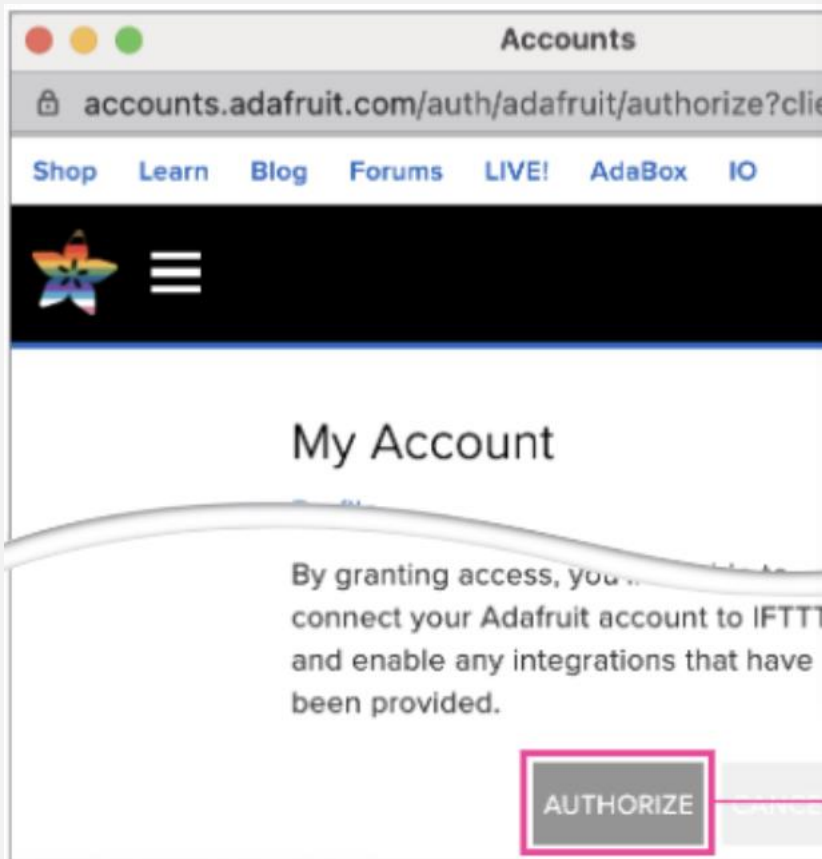
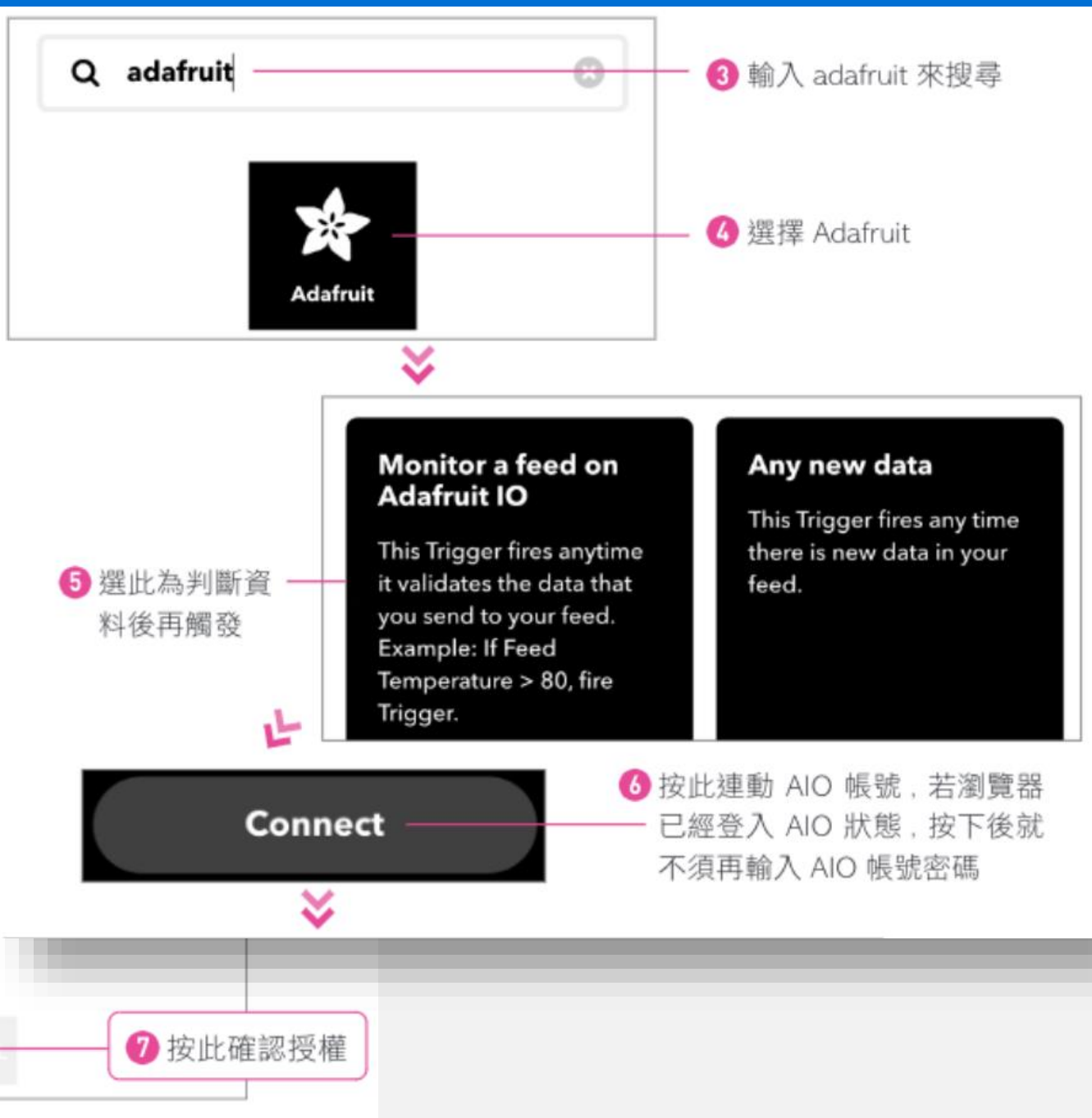
6 按此略過導覽

設定事件 A



2 按 Add 來設定事件 A

設定事件 A



設定事件 A

8 選擇已經在 AIO 建立的 feed :
timecapsule

9 選 equal to

10 輸入 100

11 按 Create trigger

Feed
timecapsule
The name of the feed to check.

Relationship
equal to
Relationship between two values.

Value
100
The value to compare against.

Create trigger

設定動作 B

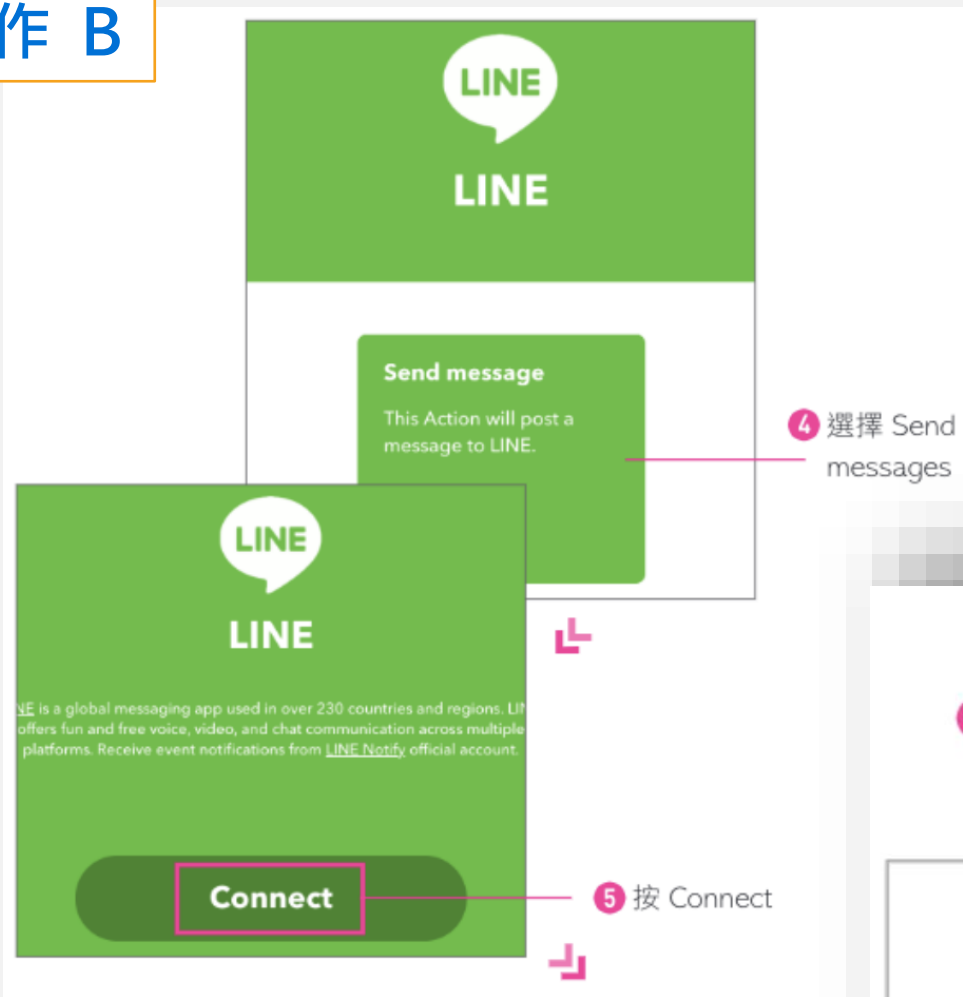
The image shows a workflow builder interface. At the top, there is a black box labeled "If" with a star icon and the text "Monitor a feed on Adafruit IO". Below it is a plus sign. The main part of the interface is a black box labeled "Then That" with a white "Add" button highlighted by a pink rectangle. A pink line connects this button to the text "1 按 Add". Below the "Then That" box is a search bar containing the text "line". A pink line connects the search bar to the text "2 輸入 line 搜尋". Below the search bar is a green square icon with a white speech bubble containing the word "LINE" and the word "LINE" below it. A pink line connects this icon to the text "3 選擇 LINE".

1 按 Add

2 輸入 line 搜尋

3 選擇 LINE

設定動作 B



設定動作 B

LINE

Send message

This Action will post a message to LINE.

Recipient

透過1對1聊天接收LINE Notify的通! ▼

Message destination

Message

The FeedValue is Operator TriggerValue at CreatedAt !

Add ingredient

Photo URL

Add ingredient

Create action

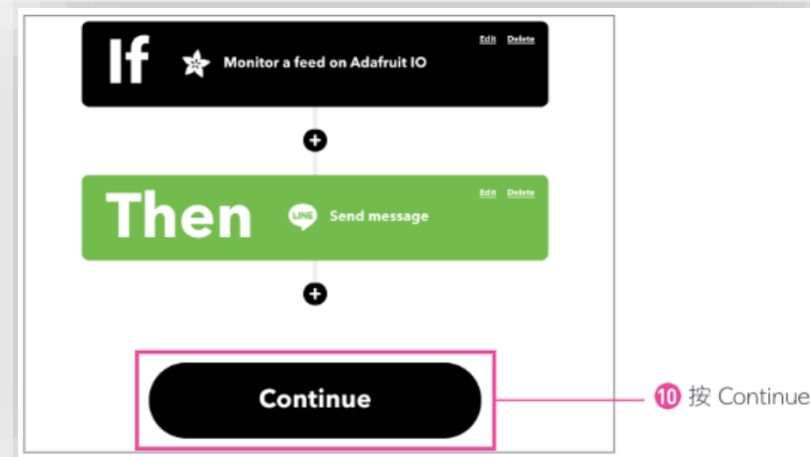
設定 LINE 訊息的內容畫面

8 刪除此處所有文字，設定訊息內容如下圖：

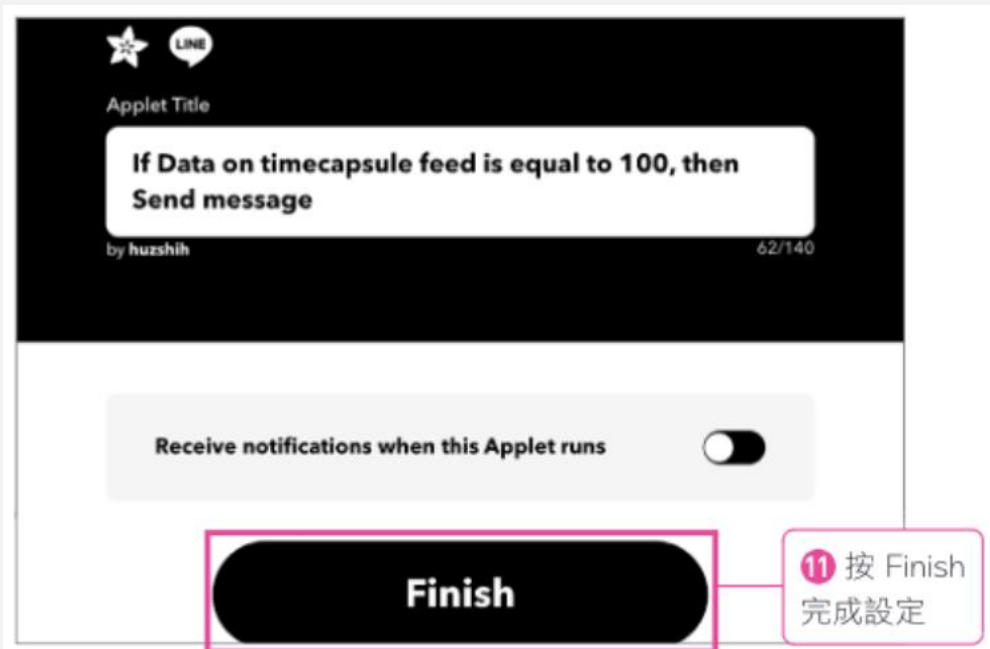
Message

想要我的財寶嗎？想要的話可以全部給你，自己去找吧！我把所有財寶都放在那裡～

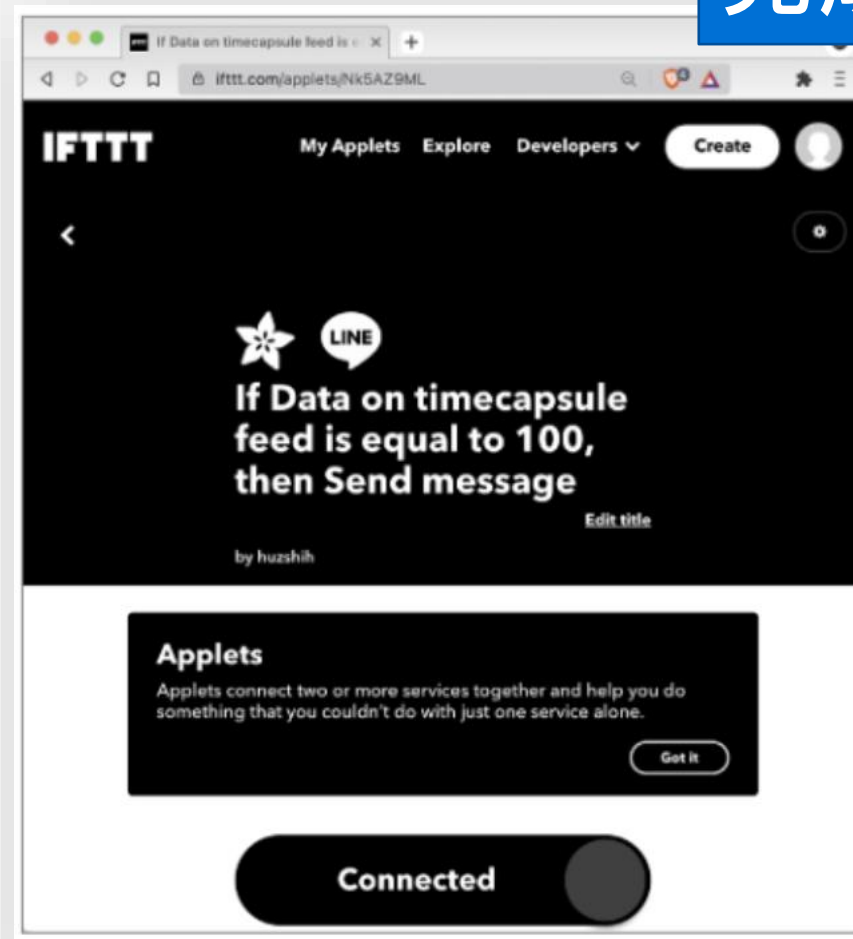
9 修改完成按 Create action

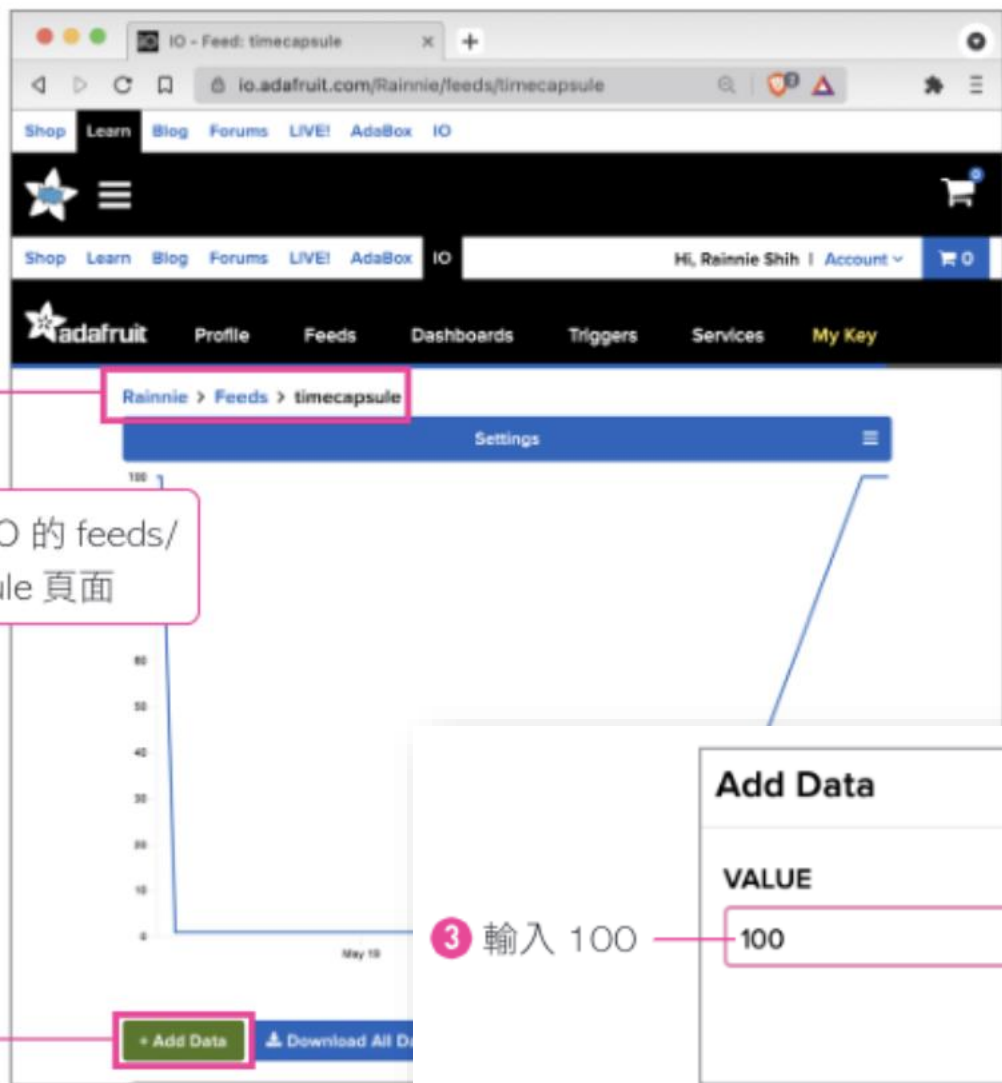


10 按 Continue



完成





1 連線到 AIO 的 feeds/
timecapsule 頁面

2 按此手動
增加資料

3 輸入 100

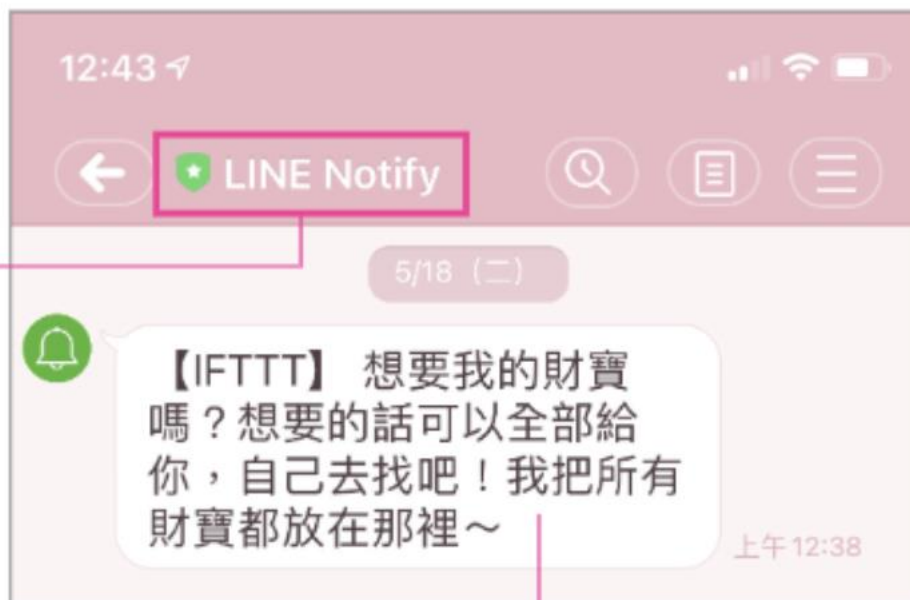
Add Data ✕

VALUE

Cancel **Create**

4 按 Create

1 打開手機的 LINE
選擇 LINE Notify

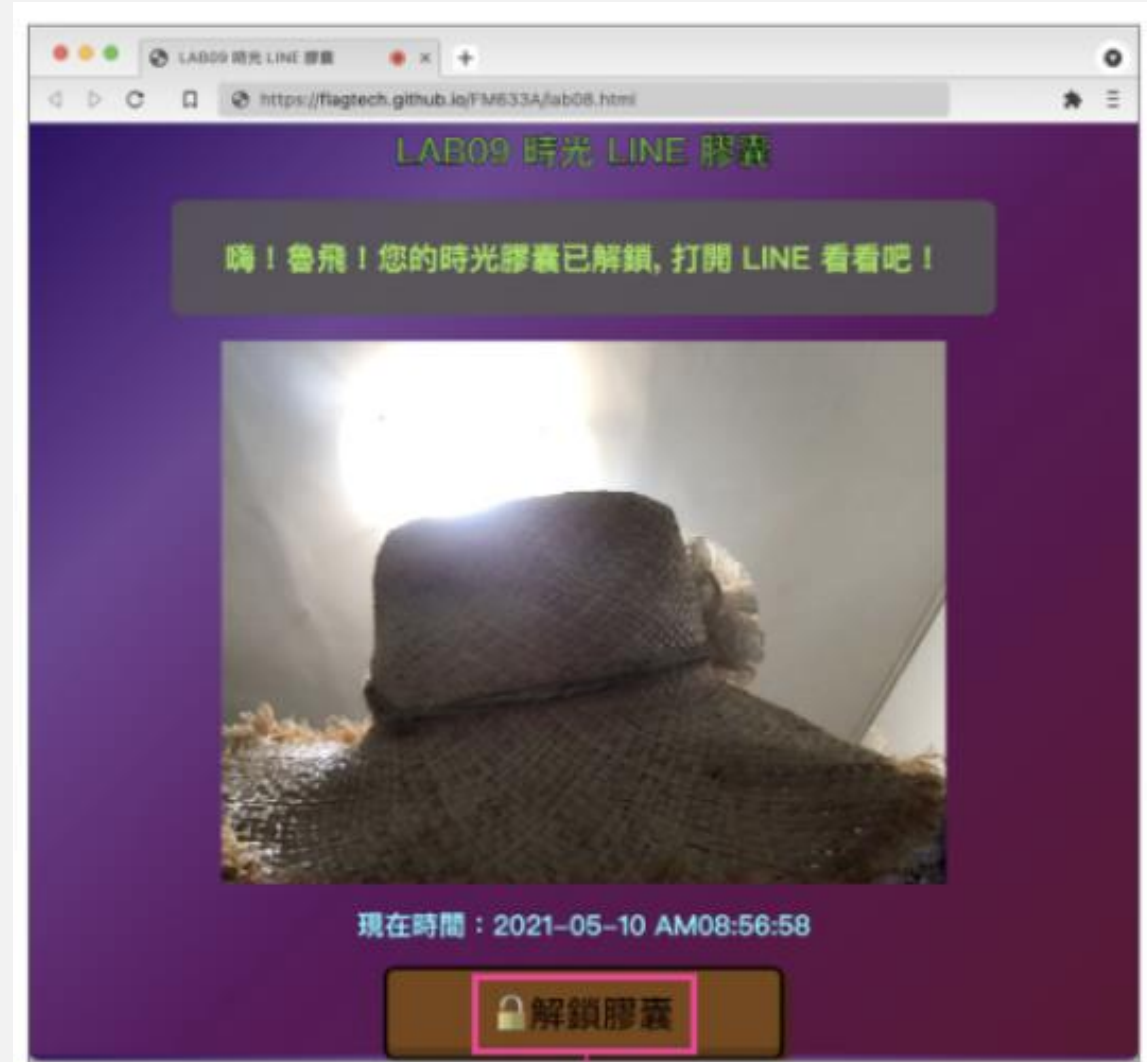


2 由 IFTTT 傳來的訊息

LAB09 時光 LINE 膠囊

將辨識結果傳至 AIO 資料庫,再經由 IFTTT 根據資料庫收到的訊息判斷發送LINE 訊息與否。

時光膠囊即將訊息傳達給未來的方式,可能是物品也可能是訊息,通常並不會設定開啟膠囊的時間,而是將其埋在地下,等待未來的某天會被開啟;白鬍子大叔想要留下 LINE 訊息給海賊群組裡面的所有人,但是必須在他退隱之後,為了確保這個訊息也可以被魯飛讀取,解鎖的條件只讓魯飛一人知道,也就是大叔留下的那頂



1 連線 <https://teachablemachine.withgoogle.com/train/image>

沒東西

30 Image Samples

Webcam Upload

魯飛

30 Image Samples

Webcam Upload

飛飛

30 Image Samples

Webcam Upload

Training

Train Model

Advanced

Preview

Export Model

You must train a model on the left before you can preview it here.

Add a class

3 依照類別拍攝不同對象
(您可以拍攝棒球帽或其他物品替代草帽)

4 按 Train Model
開始訓練模型

2 新增並修改類別名稱為：沒東西、魯飛、飛飛

Machine

沒東西

30 Image Samples

Webcam Upload

魯飛

30 Image Samples

Webcam Upload

飛飛

30 Image Samples

Webcam Upload

5 按此輸出模型

Export Model

Export your model to use it in projects.

Tensorflow.js Tensorflow Tensorflow Lite

Export your model:

Upload (shareable link) Download Upload my model

6 按此上傳模型至雲端

Your shareable link:

[https://teachablemachine.withgoogle.com/models/\[...\]](https://teachablemachine.withgoogle.com/models/[...])

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

Code snippets to use your model:

Javascript p5.js Contribute on Github

Learn more about how to use the code snippet on github.

Copy

7 按此可複製網址

Copy

8 確認上傳完成

LAB09 時光 LINE 膠囊

連線 <https://flagtech.github.io/FM633A/lab09.html>

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

膠囊解鎖時間:

填入各項資訊

按 Start

LAB09 時光 LINE 膠囊

嗨！魯飛！您的時光膠囊已解鎖，打開 LINE 看看吧！

現在時間：2021-05-10 AM08:56:58

12:43

LINE Notify

5/18 (二)

【IFTTT】想要我的財寶嗎？想要的话可以全部給你，自己去找吧！我把所有財寶都放在那裡～

由 IFTTT 傳來膠囊解鎖的訊息

課程大綱

- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



05 聲音辨識分類大應用

Teachable Machine 目前除了可以製作影像分類辨識專案外，還提供了聲音分類辨識與姿態分類辨識，本章將以聲音分類辨識完成應用。

聲音辨識

The screenshot shows the Teachable Machine interface for configuring a 'Background Noise' class. The interface includes the following elements:

- Teachable Machine** header.
- Background Noise** class name with a help icon.
- Add Audio Samples (20 minimum)** label, where '20 minimum' is highlighted with a red box.
- Mic** and **Upload** buttons.
- Class 2** label with an edit icon.
- Add Audio Samples (8 minimum):** label.
- Mic** and **Upload** buttons for the second class.
- Add a class** button at the bottom.

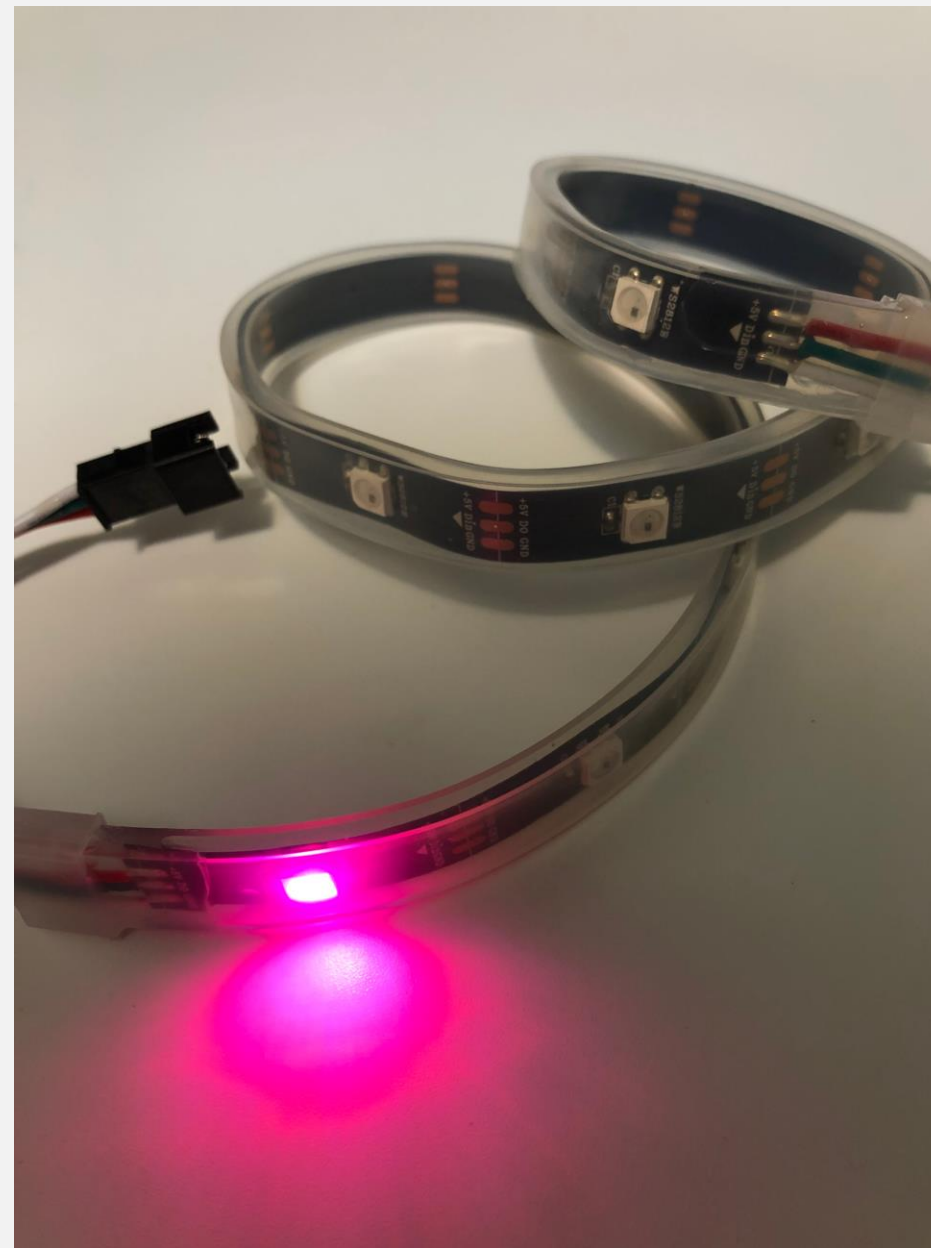
Annotations in the image:

- A red box highlights the **Background Noise** class name.
- A red box highlights the **20 minimum** text in the 'Add Audio Samples' label.
- A red box highlights the **Background Noise** class name with the text 'Background Noise 分類無法刪除' (Background Noise classification cannot be deleted).
- A red box highlights the **20 minimum** text with the text '至少需要 20 段樣本' (At least 20 samples are needed).

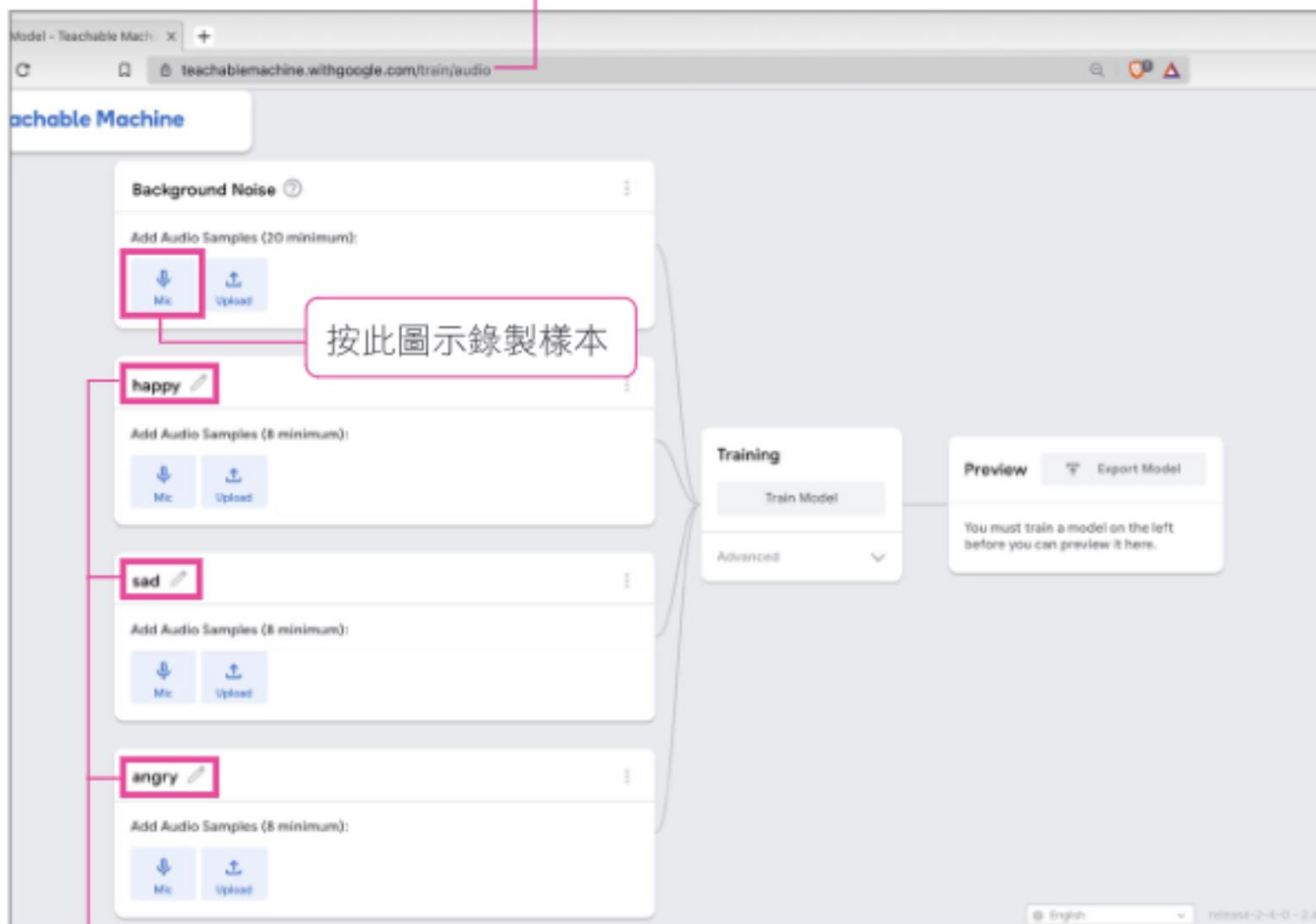
LAB10 聲控心情氣氛燈

訓練聲音辨識模型後,再透過網頁將辨識結果傳至 AIO 資料庫,控制板再根據資料庫的訊息讓條燈做出不同特效。

下班下課回到家一身疲憊,可能連開燈都有點懶,回想起一整天發生的事情,心裡總有些聲音呢喃著,這時候就來一個聲控心情氣氛燈吧!說出你的心情,讓特效燈光伴隨你回味時光。



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



新增並修改類別名稱為：happy、sad、angry

Background Noise ?

Microphone ×

Switch Microphone ▼

Add Audio Samples (20 minimum):

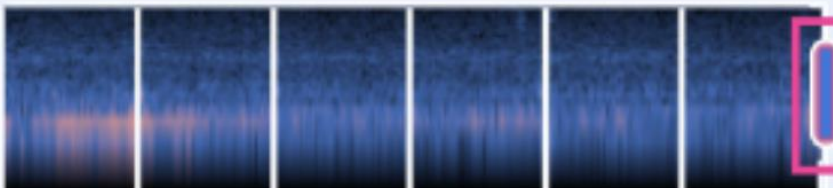
Record 20 Seconds ⚙️

按此開始
錄製

Background Noise ?

Microphone ×

Switch Microphone ∨



音訊暫存區 ▸ 00:00 / 00:20 Extract Sample

Record 20 Seconds ⚙️

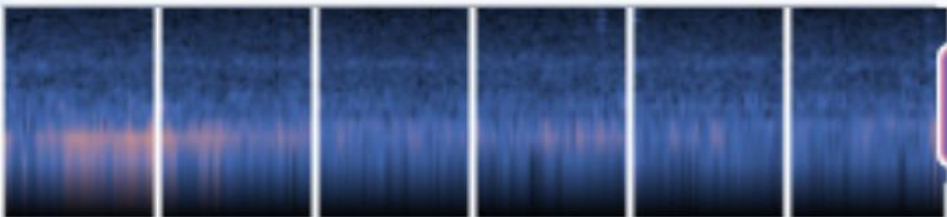
拉動調整結束點

若要重新錄製音訊只要再次按下 Record

Background Noise ?

Microphone ×

Switch Microphone ∨



00:00 / 00:20 Extract Sample

按此確認音訊作為樣本

Background Noise ?

Microphone X

Switch Microphone v

20 Audio Samples / 20 minimum

確認已符合最低樣本數量

00:00 / 00:20 Extract Sample

Record 20 Seconds

happy

Add Audio Samples (8 minimum):

Mic Upload

按此錄製 happy 分類

happy

Microphone X

Switch Microphone v

Record 15 Seconds

按此設定錄製秒數與起始延遲時間

happy

Settings

Delay: 2 seconds

Duration: 15 seconds

錄製秒數

Cancel Save Settings

儲存設定

按下錄製按鈕後延遲開始時間

按此錄製 happy 分類

happy

Microphone

Switch Microphone

00:00 / 00:07

Extract Sample

Record 15 Seconds

39 Audio Samples / 8 minimum

若要錄製新音訊只要再次按下 Record

確認音訊作為樣本

Background Noise 20 minimum
33 Audio Samples / 20 minimum

happy 8 minimum
39 Audio Samples / 8 minimum

sad 8 minimum
37 Audio Samples / 8 minimum

angry 8 minimum
34 Audio Samples / 8 minimum

開始訓練

Training
Train Model

Advanced

Preview **Export Model** 按此輸出模型

Input ON

Switch Microphone

Overlap Factor: 0.5

Output

Backg... Noise	97%
happy	
sad	
angry	

Export your model to use it in projects.

Tensorflow.js Tensorflow Lite

Export your model:
 Upload (shareable link) Download **Update my cloud model** 上傳模型

Your sharable link:
<https://teachablemachine.withgoogle.com/models/1NTjLrAdb/> **Copy** 按此可複製網址

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.

LAB10.PY 程式解說

```

1  import network
2  import ledstrip
3  from umqtt.robust import MQTTClient
4
5  #建立變數來存放條燈特效類別
6  led_strip_effect = 0
7  #設定條燈腳位為 4，燈珠數量為 15
8  ledstrip.setup(4,15)
9
10 # 連線到無線網路
11 sta_if = network.WLAN(network.STA_IF)
12 sta_if.active(True)
13 sta_if.connect("無線網路基地台", "無線網路密碼")

```

```

15 # 循環測試網路直到網路連線成功
16 while not sta_if.isconnected():
17     pass
18 print("控制板已連線")
19
20 # 建立 MQTT 客戶端物件
21 client = MQTTClient(
22     client_id="",
23     server="io.adafruit.com",
24     user="AIO 帳號",
25     password="AIO 金鑰",
26     ssl=False)

```

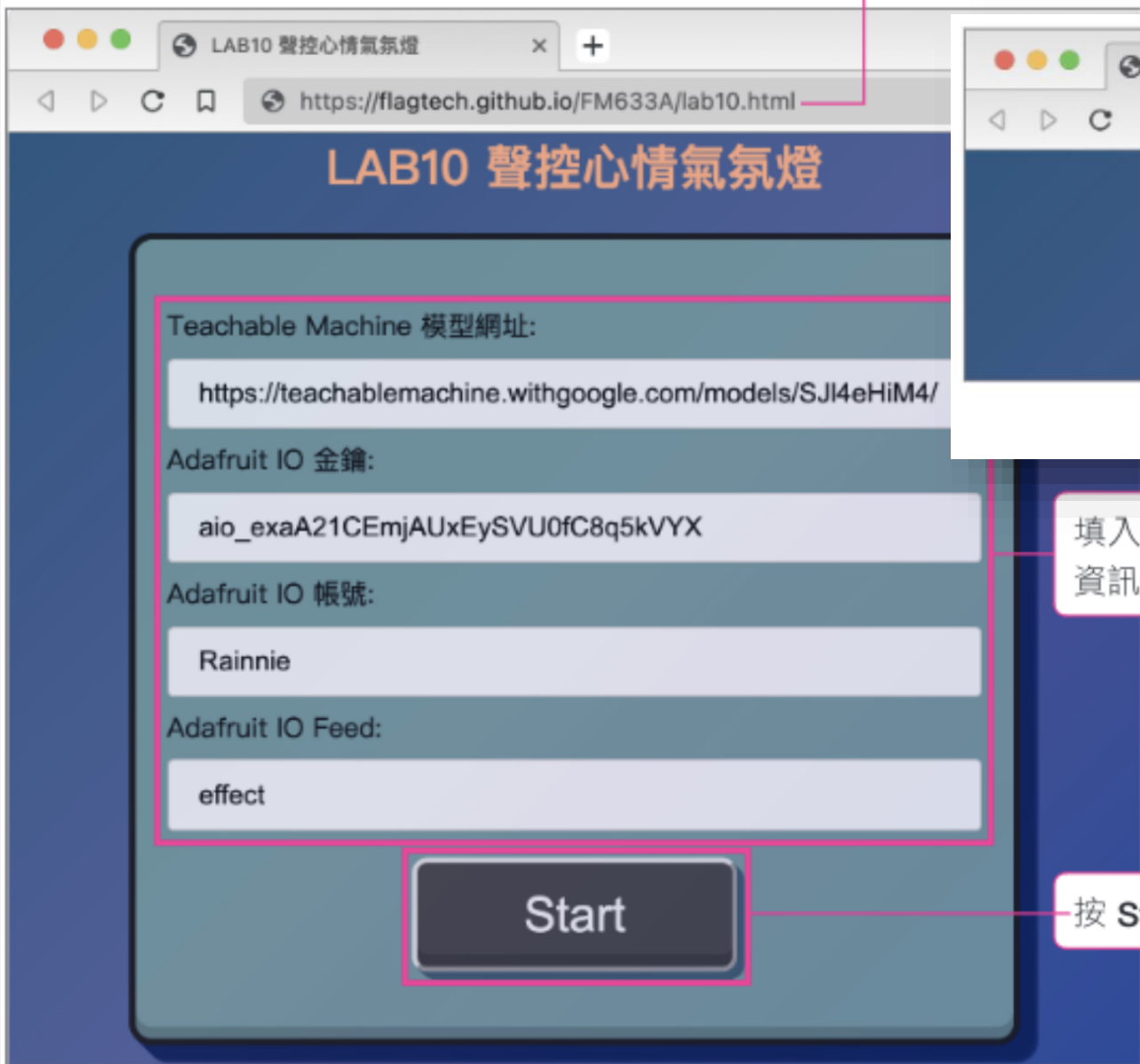
LAB10.PY 程式解說

```
28 # 註冊收到訂閱資料時的處理函式
29 def get_cmd(topic, msg):
30     # 宣告使用全域變數 led_strip_effect
31     global led_strip_effect
32     print(topic,msg)
33     # 設定條燈特效項目
34     led_strip_effect = msg
35
36 client.connect()
37 client.set_callback(get_cmd)
38 client.subscribe(client.user.encode() + b"/feeds/effect");
```

LAB10.PY 程式解說

```
40 # 不斷重複執行
41 while True:
42     client.check_msg()
43     if led_strip_effect == b'0':
44         ledstrip.clear()
45     elif led_strip_effect == b'1':
46         #rainbow_cycle(間隔毫秒時間)
47         ledstrip.rainbow_cycle(5)
48     elif led_strip_effect == b'2':
49         #cycle(r, g, b, 間隔毫秒時間)
50         ledstrip.cycle(123, 0, 20, 100)
51     elif led_strip_effect == b'3':
52         #bounce(r, g, b, 間隔毫秒時間)
53         ledstrip.bounce(23, 20, 128, 50)
```

連線 <https://flagtech.github.io/FM633A/lab10.html>



填入各項
資訊

按 **Start**



顯示辨識結果

LAB II 語音心情點播音樂盒

將語音辨識分類的結果透過 AIO 資料庫, 讓控制板播出不同音樂。

今晚...我想來點開心的音樂。隨時想要來點符合自己心情的音樂, 只要透過語音就可以馬上點歌。



LAB11.PY 程式解說

```
1 import network, time, neopixel
2 from machine import Pin, PWM
3 from umqtt.robust import MQTTClient
4
5 from rtttl import RTTTL
6
7 buzzer = PWM(Pin(5))
8 buzzer.duty(0)
9
10 def play_tone(freq, msec):
11     if freq > 0:
12         buzzer.freq(freq) # Set frequency
13         buzzer.duty(512)
14         time.sleep(msec*0.001) # Play for a number of msec
15     buzzer.duty(0) # Stop playing
16     time.sleep(0.05)
```

LAB11.PY 程式解說

```
18 sta_if = network.WLAN(network.STA_IF)
19 sta_if.active(True)
20 sta_if.connect("無線網路基地台", "無線網路密碼")
21
22 # Wait for connecting to Wi-Fi
23 while not sta_if.isconnected():
24     pass
25 print("控制板已連線")
26
27 # 建立 MQTT 客戶端物件
28 client = MQTTClient(
29     client_id="",
30     server="io.adafruit.com",
31     user="AIO 帳號",
32     password="AIO 金鑰",
33     ssl=False)
```

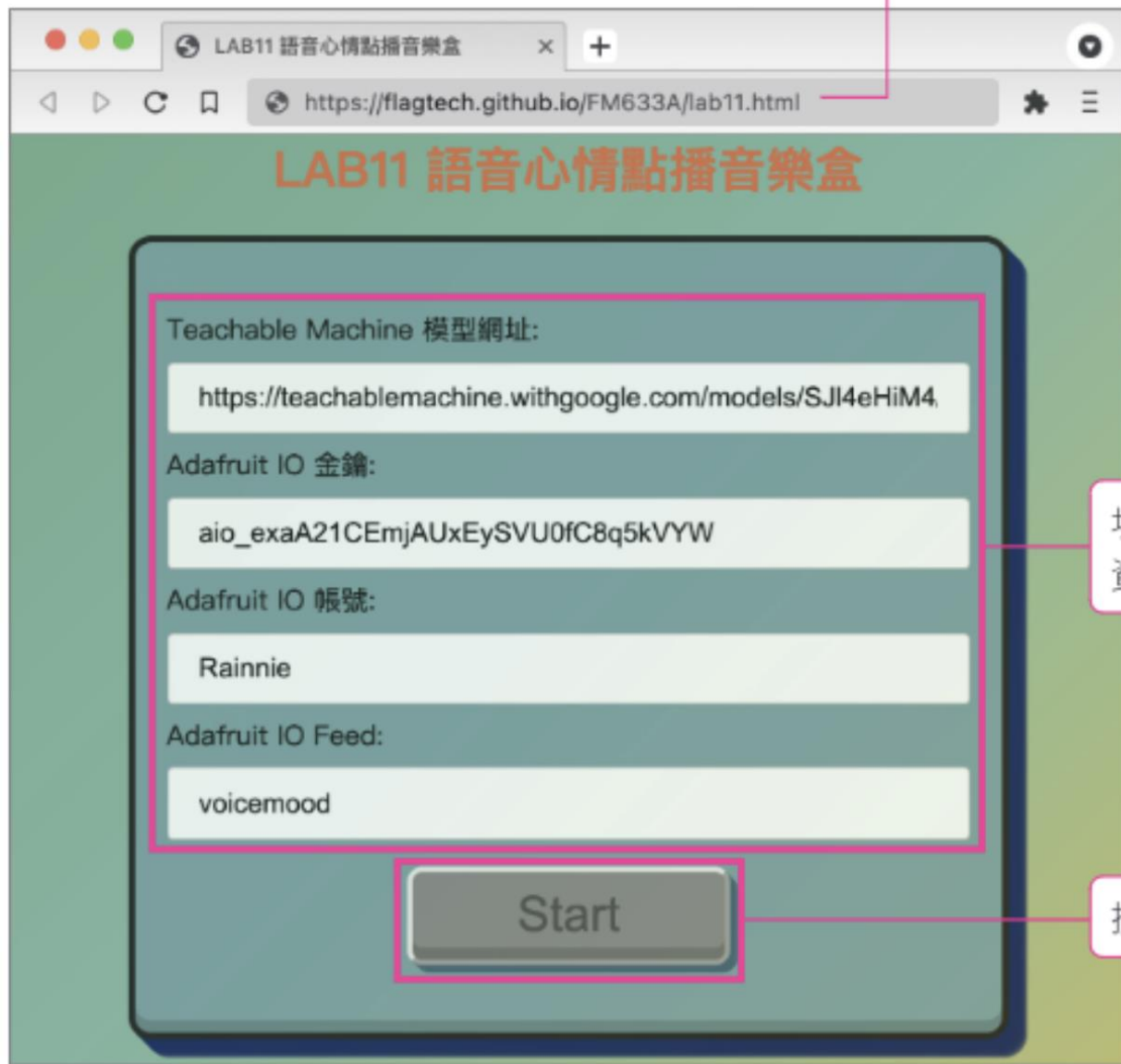
LAB11.PY 程式解說

```

35 def get_cmd(topic, msg):
36     print(topic,msg)
37     if msg == b"1":
38         print('play music1')
39         tune = RTTTL("Halloween:d=4,o=5,b=180:8d6,8g,8g,8d6,8g,8g,8d6,8g,8d#6,8 ● ● ●
40         buzzer.duty(512)
41         for freq, msec in tune.notes():
42             play_tone(freq, msec)
43         buzzer.duty(0)
44     elif msg == b"2":
45         print('play music2')
46         tune2 = RTTTL("BobMarleyNowomannocry:d=8,o=5,b=180:4c.6,b.,c.6,b.,4b.,2 ● ● ●
47         buzzer.duty(512)
48         for freq, msec in tune2.notes():
49             play_tone(freq, msec)
50         buzzer.duty(0)
51     elif msg == b"3":
52         print('play music3')
53         tune3 = RTTTL("Fugees-KillingMeSoftly:d=4,o=5,b=90:p,8e,f.,8g,a.,8g,d,g ● ● ●
54         buzzer.duty(512)
55         for freq, msec in tune3.notes():
56             play_tone(freq, msec)
57         buzzer.duty(0)
59 client.connect()
60 client.set_callback(get_cmd)
61 client.subscribe(client.user.encode() + b"/feeds/voicemood");
62
63 while True:
64     client.check_msg()

```

連線 <https://flagtech.github.io/FM633A/lab11.html>



填入各項
資訊

按 Start



顯示辨識結果

LAB12 台語問答練習機

針對一個類別增加不同樣本,讓機器學習更廣泛,再將辨識結果透過 AIO 資料庫,若成功答題則讓蜂鳴器發出音效。

學習一種語言最好的方法就是多用多練習,而台語又因為各地有不同說法,這時候就可以針對同一個詞彙使用不同樣本訓練,再把聲音辨識模型匯入網頁後就可以當作台語小老師。



連線 <https://itaigi.tw/>



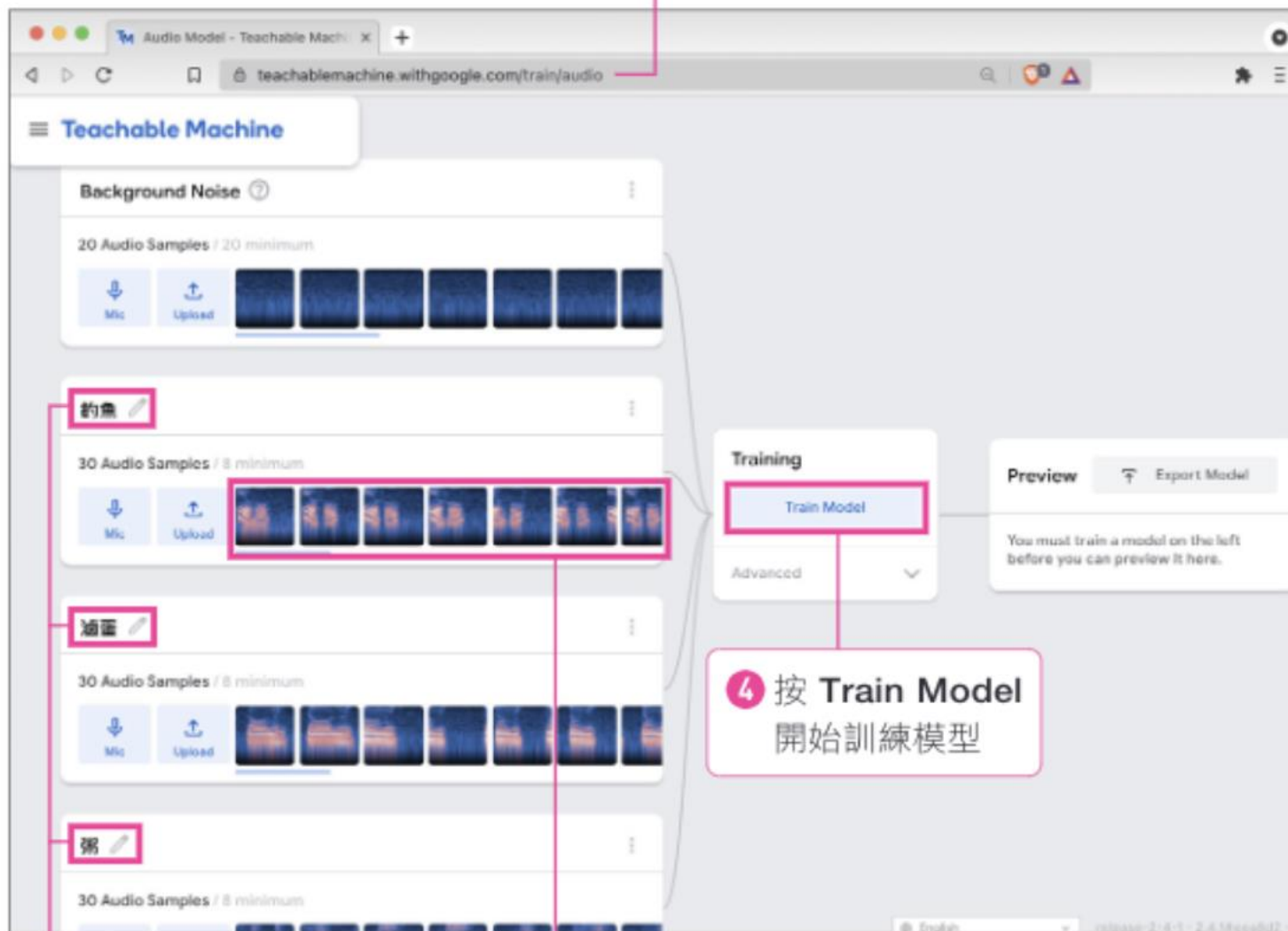
按講台語



按此播放
語音

不同地方會有不同說法

1 連線 <https://teachablemachine.withgoogle.com/train/audio>



2 新增並修改類別名稱為：
釣魚、滷蛋、粥

3 依照類別錄製
相對應發音

4 按 Train Model
開始訓練模型

Preview **Export Model**

Input ON

Switch Microphone

Overlap Factor: 0.5

Output

Backg... Noise	80%
釣魚	
滷蛋	
粥	

按此輸出模型

Export your model to use it in projects.

Tensorflow.js **上傳模型** Tensorflow Lite

Export your model:

Upload (shareable link) Download **Update my cloud model**

Your sharable link:

<https://teachablemachine.withgoogle.com/models/1NTjLrAdb/> **按此可複製網址** **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.

LAB12.PY 程式解說

```
1 import network, time
2 from machine import Pin, PWM
3 from umqtt.robust import MQTTClient
4
5 from rtttl import RTTTL
6
7 buzzer = PWM(Pin(5))
8 buzzer.duty(0)
9
10 def play_tone(freq, msec):
11     if freq > 0:
12         buzzer.freq(freq)
13         buzzer.duty(512)
14         time.sleep(msec*0.001)
15     buzzer.duty(0)
16     time.sleep(0.05)
17
18 # 連線到無線網路
19 sta_if = network.WLAN(network.STA_IF)
20 sta_if.active(True)
21 sta_if.connect("無線網路基地台", "無線網路密碼")
22
23 # 循環測試網路直到網路連線成功
24 while not sta_if.isconnected():
25     pass
26 print("控制板已連線")
27
28 # 建立 MQTT 客戶端物件
29 client = MQTTClient(
30     client_id="",
31     server="io.adafruit.com",
32     user="AIO 帳號",
33     password="AIO 金鑰",
34     ssl=False)
```

LAB12.PY 程式解說

```
36 # 註冊收到訂閱資料時的處理函式
37 def get_cmd(topic, msg):
38     print(topic,msg)
39     if msg == b"100":
40         print('play music')
41         tune = RTTTL("winner:d=4,o=5,b=140:16e6,16e6,32p,8e6,16c6,8e6,8g6,8p,8g...")
42         for freq, msec in tune.notes():
43             play_tone(freq, msec)
44         buzzer.duty(0)
45
46 client.connect()
47 client.set_callback(get_cmd)
48 client.subscribe(client.user.encode() + b"/feeds/tw");
49
50 while True:
51     client.check_msg()
```

連線 <https://flagtech.github.io/FM633A/lab12.html>

LAB12 台語問答練習機

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

填入各項
資訊

按 Start



語音辨識 API

google.com/intl/en/chrome/demos/speech.html



Web Speech API Demonstration

Click on the microphone icon and begin speaking for as long as you like.



中文

中文(台灣)

LAB13 哼唱音樂盒

使用 Web Speech API 辨識說出的簡譜數字, 將結果傳至 AIO 資料庫, 開發板再透過 MQTT 取得數字後, 讓蜂鳴器發聲。

面對一篇音樂樂譜或簡譜, 可能許多人都可以辨認出音名、唱名來, 比如簡譜上的 1 代表音名為 C, 唱名為 Do, 2 是 D(Re)...等, 雖然知道唱名, 但對於音準不是那麼好的人, 要唱對音可能不是那麼容易, 接下來的哼唱音樂盒只要將簡譜數字用講的, 就算不會辨認簡譜音名、唱名, 蜂鳴器也會播放正確的音出來。



開發板的程式中先使用元組建立數字對應的音高（頻率）變數：

```
rhythm = (261, 294, 330, 349, 392, 440, 494)
```

▲ Tuple (元組) 由零或多個元素組成，使用逗號分隔各個元素，前後為小括號，內容不可變動，所以很適合建立要事先設定好不會再變動的資料。

在接收到資料後，將接收到的數字當作索引 (index)，從變數 **rhythm** 中取值以 `play_tone(freq, msec)` 函式播放：

```
def get_cmd(topic, msg):  
    msgs = msg.decode()  
    for i in msgs:  
        print(rhythm[int(i)])  
        buzzer.duty(512)  
        play_tone(rhythm[int(i)], 500)
```

```
buzzer.duty(0)  
time.sleep(0.1)
```

▲ 使用 `index` 取值必須為整數，所以 MQTT 取得的資料類型需要使用 `int()` 轉換後才能使用。

LAB13.PY 程式解說

```
1 import network, time, neopixel
2 from machine import Pin, PWM
3 from umqtt.robust import MQTTClient
4
5 # 對應頻率
6 rhythm = (261, # C4
7           294, # D4
8           330, # E4
9           349, # F4
10          392, # G4
11          440, # A4
12          494) # B4
13
14 buzzer = PWM(Pin(5))
15 buzzer.duty(0)
16
17 def play_tone(freq, msec):
18     if freq > 0:
19         buzzer.freq(freq)
20         buzzer.duty(512)
21         time.sleep(msec*0.001)
22     buzzer.duty(0)
23     time.sleep(0.05)
```

LAB13.PY 程式解說

```
25 # 連線到無線網路
26 sta_if = network.WLAN(network.STA_IF)
27 sta_if.active(True)
28 sta_if.connect("無線網路基地台", "無線網路密碼")
29
30 # 循環測試網路直到網路連線成功
31 while not sta_if.isconnected():
32     pass
33 print("控制板已連線")
34
35 # 建立 MQTT 客戶端物件
36 client = MQTTClient(
37     client_id="",
38     server="io.adafruit.com",
39     user="AIO 帳號",
40     password="AIO 金鑰",
41     ssl=False)
```

連線 <https://flagtech.github.io/FM633A/lab13.html>

LAB13 哼唱音樂盒

Adafruit IO 金鑰:
aio_exaA21CEmjAUxEySVU0fC8q5kVYX

Adafruit IO 帳號:
Rainnie

Adafruit IO Feed:
music

設定完成

填入各項
資訊

按 設定完成

LAB13 🎵 哼唱音樂盒 🎵

5334221234555



按此開始或停止語音辨識

開始唱~

清除全部

中文

按此讓蜂鳴器開始播放

課程大綱

- 00：實驗環境設定
- 01：Teachable Machine：快速建立 AI 影像辨識
- 02：互動中心- ESP8266 單晶片控制板
- 03：影像辨識 IoT 物聯網應用
- 04：影像辨識分類大應用
- 05：聲音辨識分類大應用
- 06：姿態辨識分類大應用



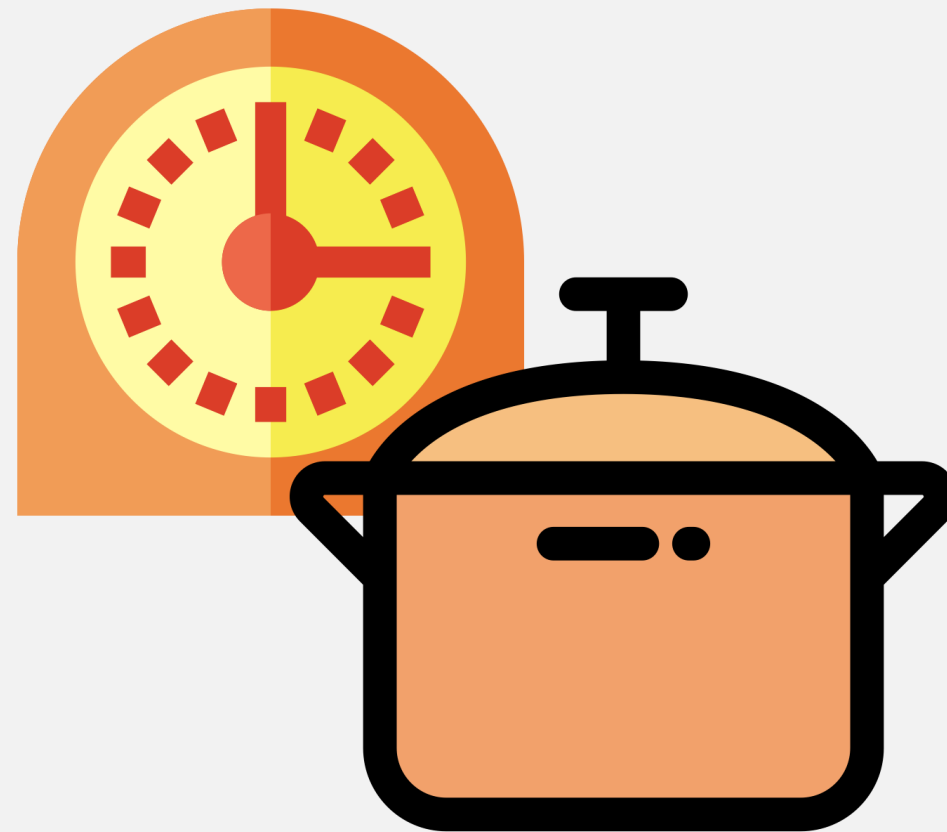
06 姿態辨識分類大應用

Teachable Machine 除了可以製作影像分類辨識與聲音分類辨識專案外，還提供了姿態分類辨識，本章將以姿態分類辨識完成應用。

LAB14 姿態料理計時器

運用姿態辨識特定姿勢啟動計時器, 時間到的時候會觸發蜂鳴器發出警示, 這時再做出特定姿勢就可以停止計時。

做料理的時候, 雙手可能總是沾滿了食材油漬或是調味料粉末, 但卻時常需要使用計時器輔助料理, 若不想每次操作都要清潔雙手, 這時候運用姿態控制的計時器就是個不可或缺的料理好幫手。



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

Start
Stop
none

新增並修改類別
名稱為：**start**、**stop**、**none**

按 Webcam 開啟攝影機

Teachable Machine

start

Add Pose Samples:

Webcam Upload

stop

Add Pose Samples:

Webcam Upload

none

Add Pose Samples:

Training

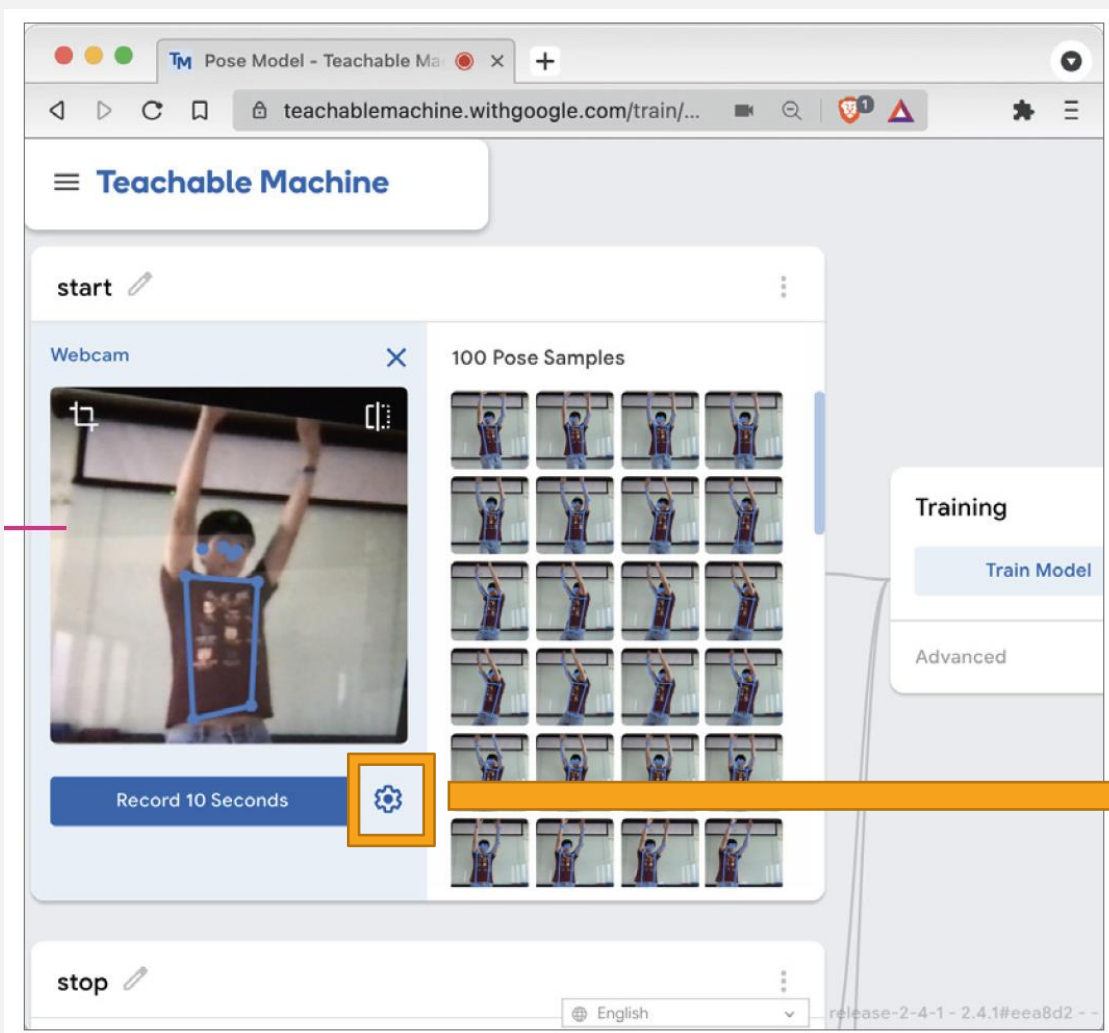
Train Model

Advanced

Preview Export Model

You must train a model on the left before you can preview it here.

拍攝姿態樣本



Teachable Machine

start

100 Pose Samples

Webcam Upload

stop

100 Pose Samples

Webcam Upload

平舉代表 stop

none

201 Pose Samples

Webcam Upload

Training

Train Model

Advanced


Preview Export Model

You must train a model on the left before you can preview it here.

拍攝完按此訓練模型

Preview **Export Model** 按此輸出模型

Input ON Webcam



Output

start	<div style="width: 100%;"></div> 100%
stop	<div style="width: 0%;"></div>
none	<div style="width: 0%;"></div>

Export your model to use it in projects. ✕

Tensorflow.js i

上傳模型

按此可複製網址

Export your model:

Upload (shareable link) Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/UA-n3Isu5/>

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.

範例程式：LAB14.PY

```
# 連線到無線網路
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect("無線網路基地台", "無線網路密碼")

# 循環測試網路直到網路連線成功
while not sta_if.isconnected():
    pass
print("控制板已連線")

# 建立 MQTT 客戶端物件
client = MQTTClient(
    client_id="D1mini",
    server="io.adafruit.com",
    user="AIO 帳號",
    password="AIO 金鑰",
    ssl=False)
```



或 F5

互動環境 (Shell) ×

```
MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
```

控制板已連線

https://flagtech.github.io/FM633A/lab14.html

LAB14 姿態料理計時器

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

填入各項資訊

按 Start

LAB14 姿態料理計時器實驗結果

LAB14 姿態料理計時器

0

0分5秒

擺出姿勢開始計時器或停止警示音

倒數計時秒數：5

設定秒數

LAB I5 姿態辨識鬧鐘

在網頁設定好鬧鐘, 若時間到再透過 AIO 資料庫讓硬體發出鬧鐘鈴聲, 停止則必須透過特定的姿態辨識。

一般的鬧鐘一按就停, 或者有貪睡功能的鬧鐘也讓人很輕易地不斷按下貪睡, 等到睡過頭了才後悔不已, 這時候就需要一個姿態辨識鬧鐘! 鬧鐘響起後, 若沒有離開床鋪到鏡頭前擺出特定姿勢是無法停止的, 可以藉此增加起床的成功率。



本實驗使用與上一個實驗 **Lab14 姿態料理計時器** 相同模型，完成控制板程式設計後可直接連線至應用網頁實測。

要讓鈴聲響起後可以隨時中斷停止而不用播完整首音樂，就不能寫成一個迴圈從頭直接播到尾，要改成每次只播一個音，然後檢查是否繼續播放。先將 RTTTL 音符資訊放到變數，再算出音符數量：

```
# 將音符資訊放入變數
tune_list = []
for freq, msec in tune.notes():
    tune_list.append((freq, msec))

# 計算音符數量
tune_lenth = len(tune_list)

# 建立播放索引變數和鬧鐘狀態變數
tune_index = 0
buzzer_on = False
```

▲ 使用 len() 可以計算出 str (字串) 中有幾個字元、list 跟 tuple 有幾個元素。



註冊收到訂閱資料時的處理函式中，若收到數值 **100** 就將狀態改為 **True**，收到 **0** 則改為 **False**：

```
def get_cmd(topic, msg):
    global buzzer_on
    print(topic,msg)
    if msg == b"100":
        buzzer_on = True
        print('Times up!')
    else:
        buzzer_on = False
        print('CLEAR.')
```

播放的時候必須每播一個音符就檢查鬧鐘狀態是否成立，若是則播放一個音符後遞增播放索引值，大於音符數量則索引值歸零：

```
while True:
    client.check_msg()
    if buzzer_on :
        # 分別從音符資料中取值
        freq = tune_list[tune_index][0]
        msec = tune_list[tune_index][1]
        if freq > 0:
            buzzer.freq(freq)
            buzzer.duty(512)
            time.sleep(msec*0.001)
        buzzer.duty(0)
        time.sleep(0.05)
        # 準備播放下一個音符
        tune_index += 1
        if tune_index >= tune_lenth:
            tune_index = 0
    else:
        buzzer.duty(0)
        # 重置播放順序索引
        tune_index = 0
```

連線 <https://flagtech.github.io/FM633A/lab15.html>

LAB15 姿態辨識鬧鐘

Teachable Machine 模型網址:

Adafruit IO 金鑰:

Adafruit IO 帳號:

Adafruit IO Feed:

填入各項
資訊

按 **Start**

LAB15 姿態辨識鬧鐘


11:22:35AM

設定鬧鐘時間：

11 25 00 AM

設定鬧鐘

請設定鬧鐘



按此完成鬧鐘設定

調整鬧鐘時間

Detailed description: The image shows a web browser window with the title 'LAB15 姿態辨識鬧鐘' and the URL 'https://flagtech.github.io/FM633A/lab15.html'. The main content area has a dark blue background. At the top, the title is in green. Below it, the current time '11:22:35AM' is displayed in white. A time selection interface follows, with the label '設定鬧鐘時間：' in white. The time is set to '11 25 00 AM', where each number and 'AM' is in a white box with a dropdown arrow. A pink box highlights these time selection boxes. To the right, a pink callout box points to the time selection area with the text '按此完成鬧鐘設定'. Below the time selection, a green button labeled '設定鬧鐘' is highlighted with a pink box. A second pink callout box points to this button with the text '調整鬧鐘時間'. Below the green button is a blue button with the text '請設定鬧鐘' in orange. At the bottom, there is a small video frame showing a person with a blue pose estimation overlay on their body, indicating the application's functionality. The browser's address bar and window controls are visible at the top.

LAB16 AI 瑜珈教練

以不同的瑜珈姿勢樣本進行訓練, 將模型載入網頁互動, 最後將結果傳至開發板給予使用者回饋。

若是我們能夠訓練模型學會辨識各種不同的瑜珈姿勢, 再搭配網頁互動, 這一來就可以打造自己的瑜珈私人教練, 工作累了還可以隨時開啟應用, 舒展筋骨一番, 是不是方便實用呢?





半月式

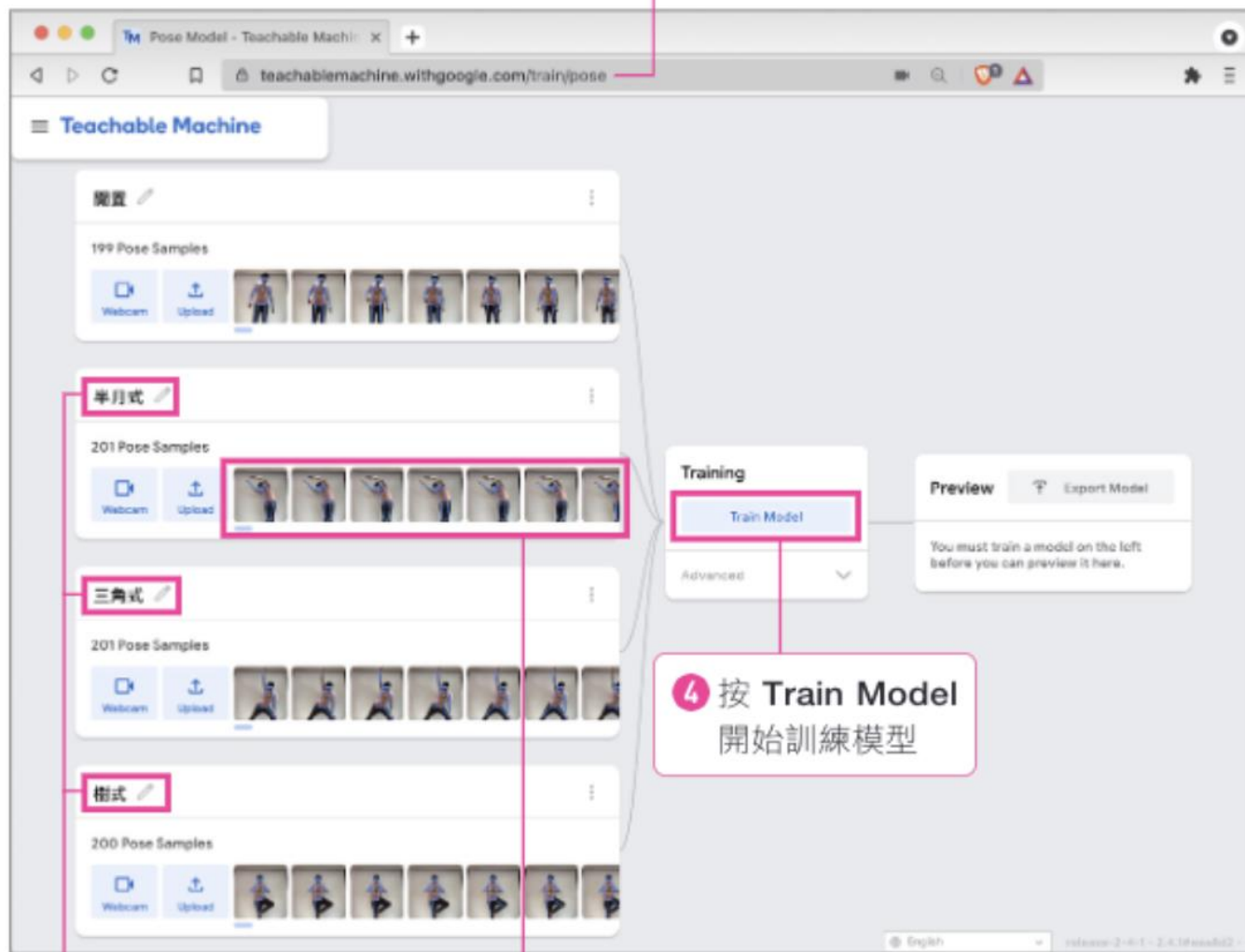


三角式



樹式

1 連線 <https://teachablemachine.withgoogle.com/train/pose>




2 新增並修改類別名稱
為：閒置、半月式、
三角式、樹式

3 依照類別錄製相對
應肢體動作

Preview **Export Model**

Input ON Webcam



Output

- 圓置 2%
- 半月式
- 三角式
- 樹式 88%

按此輸出模型

Export your model to use it in projects. ✕

Tensorflow.js ⓘ

上傳模型

按此可複製網址

Export your model:

Upload (shareable link) Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/Rr8oHhwu3/>

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.

連線 <https://flagtech.github.io/FM633A/lab16.html>

LAB16 AI 瑜珈教練

Teachable Machine 模型網址:

<https://teachablemachine.withgoogle.com/models/RrBoHhwu3/>

Adafruit IO 金鑰:

aio_exaA21CEmjAUxEySVU0fC8q5kVYX

Adafruit IO 帳號:

Rainnie

Adafruit IO Feed:

yoga

填入各項
資訊

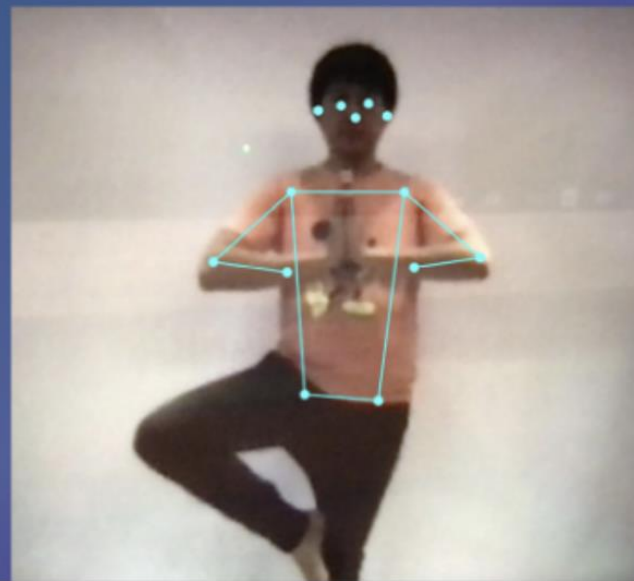
Start

按 Start

LAB16 AI 瑜珈教練

太厲害了！姿勢很標準呢！

樹式



範例程式：LAB16.PY

開發板程式與Lab12 台語問答練習機程式相同, 唯feed名稱不同。

```
19 ta_if = network.WLAN(network.STA_IF)
20 ta_if.active(True)
21 ta_if.connect("無線網路基地台", "無線網路密碼")
22
23 循環測試網路直到網路連線成功
24 while not sta_if.isconnected():
25     pass
26 print("控制板已連線")
27
28 建立 MQTT 客戶端物件
29 client = MQTTClient(
30     client_id="",
31     server="io.adafruit.com",
32     user="AIO 帳號",
33     password="AIO 金鑰",
34     ssl=False)
```



Q & A

謝謝大家