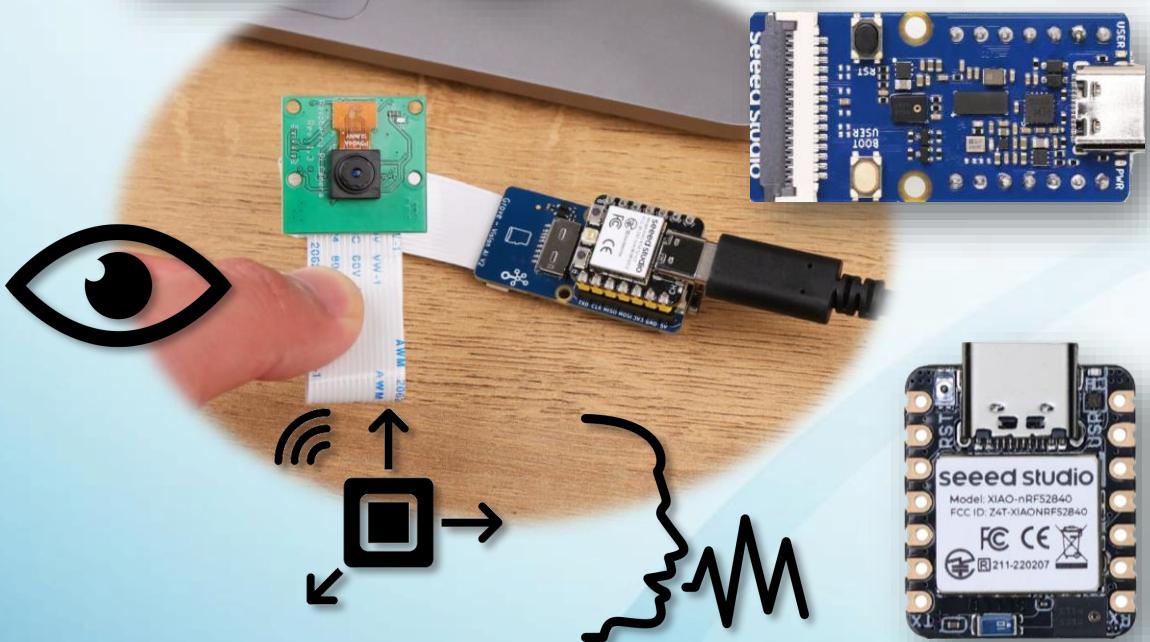


# OmniXRI TinyML 小學堂 2025



歡迎加入  
邊緣人俱樂部



沒有最邊



只有更邊

Cortex-M  
Processor

Ethos-U  
MicroNPU



【第 8 講】  
聲音辨識應用—  
環境音辨識



歐尼克斯實境互動工作室 (OmniXRI Studio)  
許哲豪 (Jack Hsu)

# 簡報大綱



- 8.1. 喚醒詞偵測簡介
- 8.2. 聲音資料集建置
- 8.3. TensorFlow Lite 模型訓練
- 8.4. Arduino 模型部署與測試

本課程完全免費，請勿移作商業用途！  
歡迎留言、訂閱、點讚、轉發，讓更多需要的朋友也能一起學習。

完整課程大綱：<https://omnixri.blogspot.com/2025/03/omnixri-tinyml-2025-0.html>  
課程直播清單：<https://www.youtube.com/@omnixri1784streams>



## 8.1. 喚醒詞偵測簡介

# 喚醒詞偵測

對一小段（數秒）聲音（連續信號）進行分類（辨識）如環境聲音（動物、機械、自然等）、生理訊號、機械振動、感測器變化等。

可以是非語言人聲或語音關鍵詞（非句子）。

常見同義詞

- **Keyword Spotting (KWS)**：語音關鍵詞定位
- **Keyword Detection**：語音關鍵詞偵測
- **Voice Command**：語音命令
- **Wake-Up-Word**：喚醒詞

常見應用

- 智能音箱、動作辨識、生理監看等



# 案例分享 — 聲控電風扇

正確示範（美女版） 錯誤示範（阿媽版） 正確示範（台灣國語）



<https://www.youtube.com/shorts/X8yJjRj7Uus>



<https://www.youtube.com/shorts/xlz1m2Cp3IY>



# 案例分享 – 2024 總統盃黑客松卓越團隊



Youtub : <https://youtu.be/0E5gW9PcSgs>

資料來源：<https://www.president.gov.tw/News/28977>

## 寧靜追蹤師 Quiet Tracker

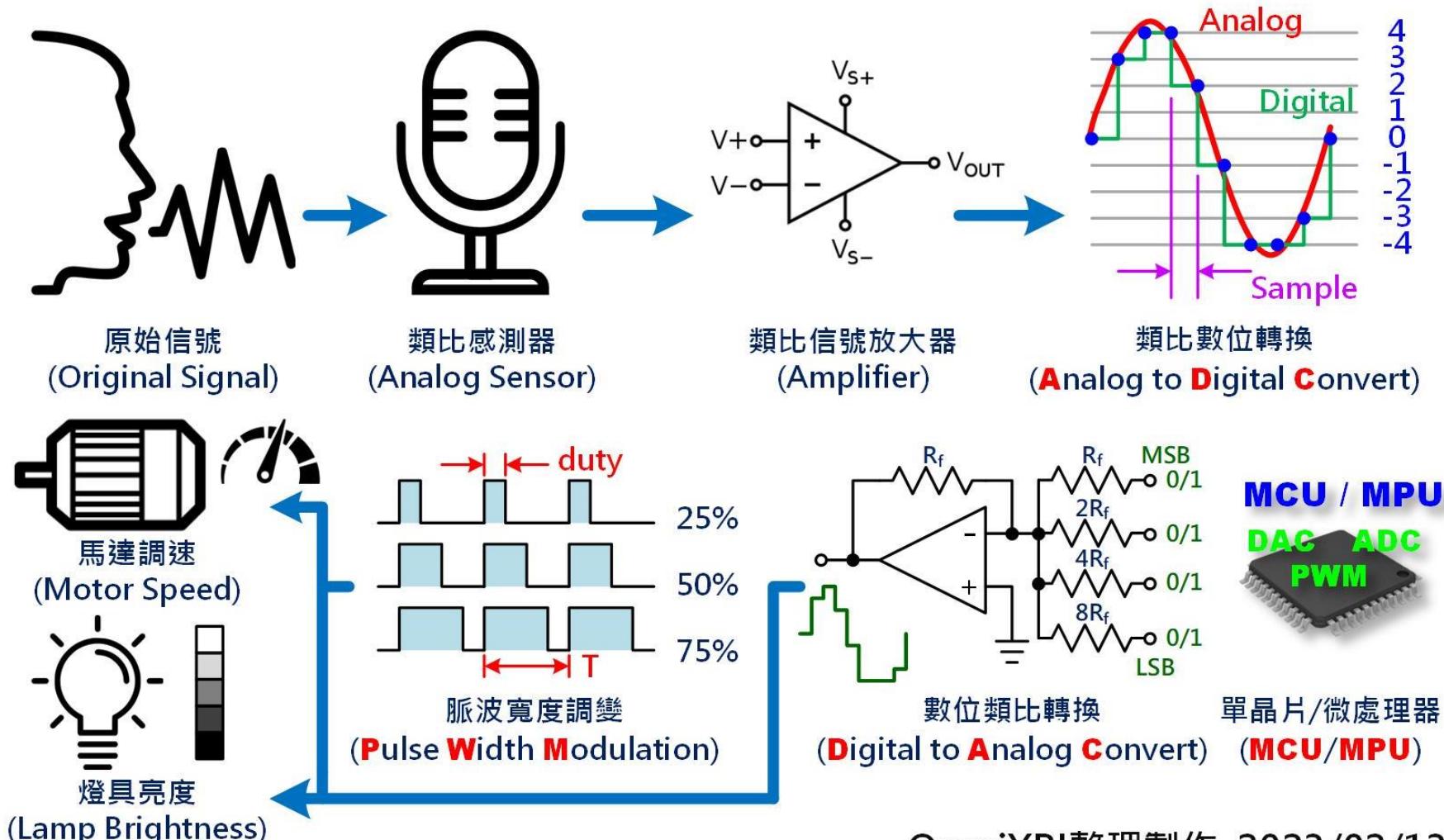
中央研究院人文社會科學研究中心詹大千研究員帶領下，共同打造結合大數據、低成本、可廣設之聲音盒子，藉由即時監測的聲音數據建構「安靜適宜性指數」(Quietness Suitability Index, QSI)地圖，未來可針對不易稽查、瞬間消逝的噪音進行即時監測並主動告發。





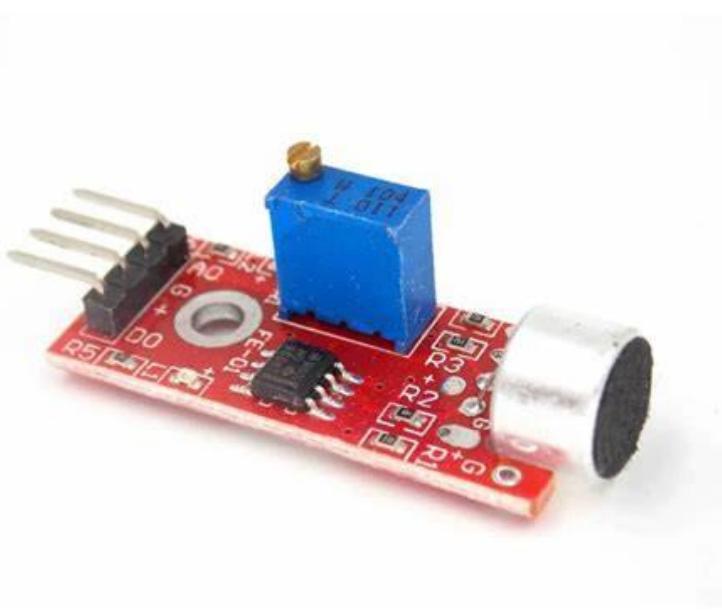
## 8.2. 聲音資料集建置

# 感測器類比/數位信號互換



資料來源：<https://omnixri.blogspot.com/2023/02/vmaker-edge-ai-02-ai.html>

# 常見錄音裝置



電容式類比麥克風  
(類比電壓輸出)

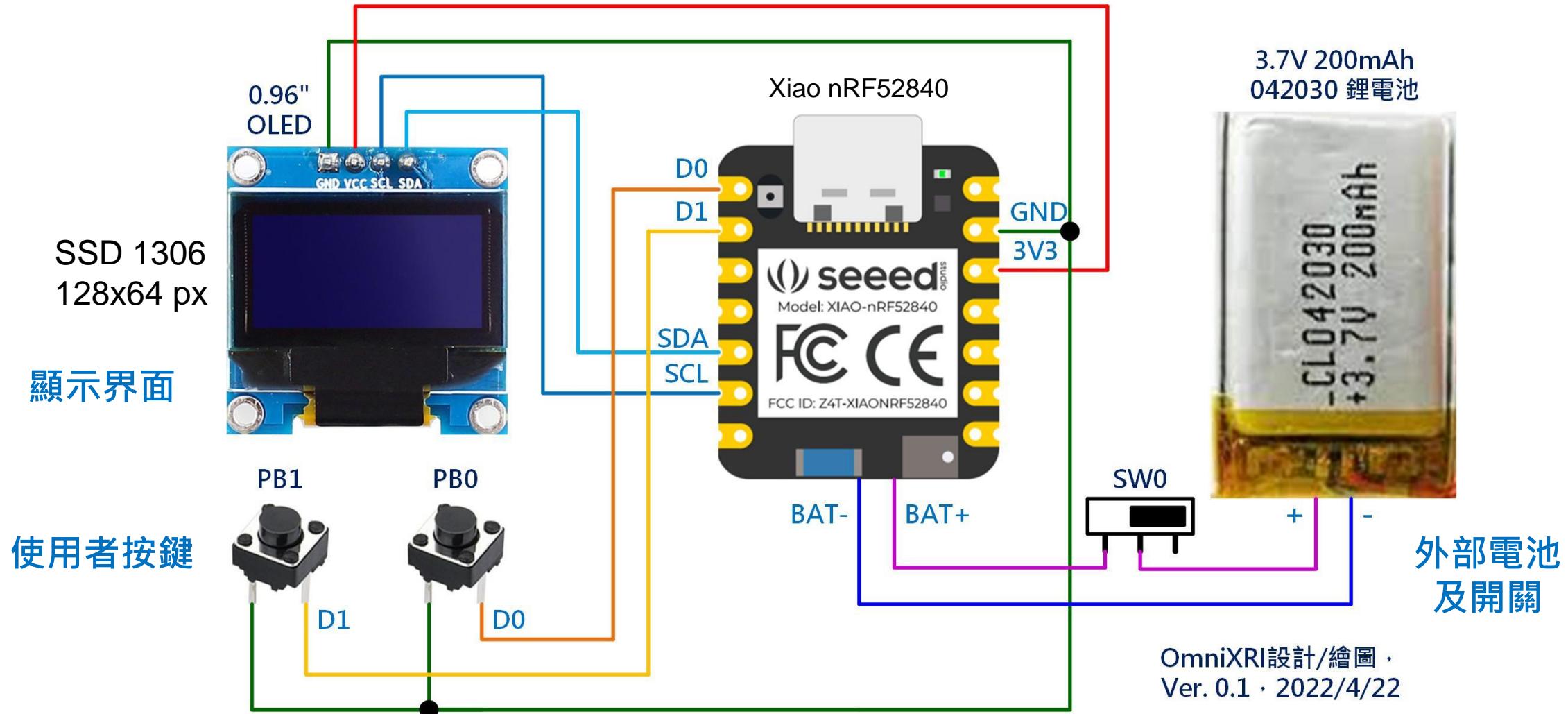


微機電式數位麥克風  
(I2C, I2S 數位輸出)

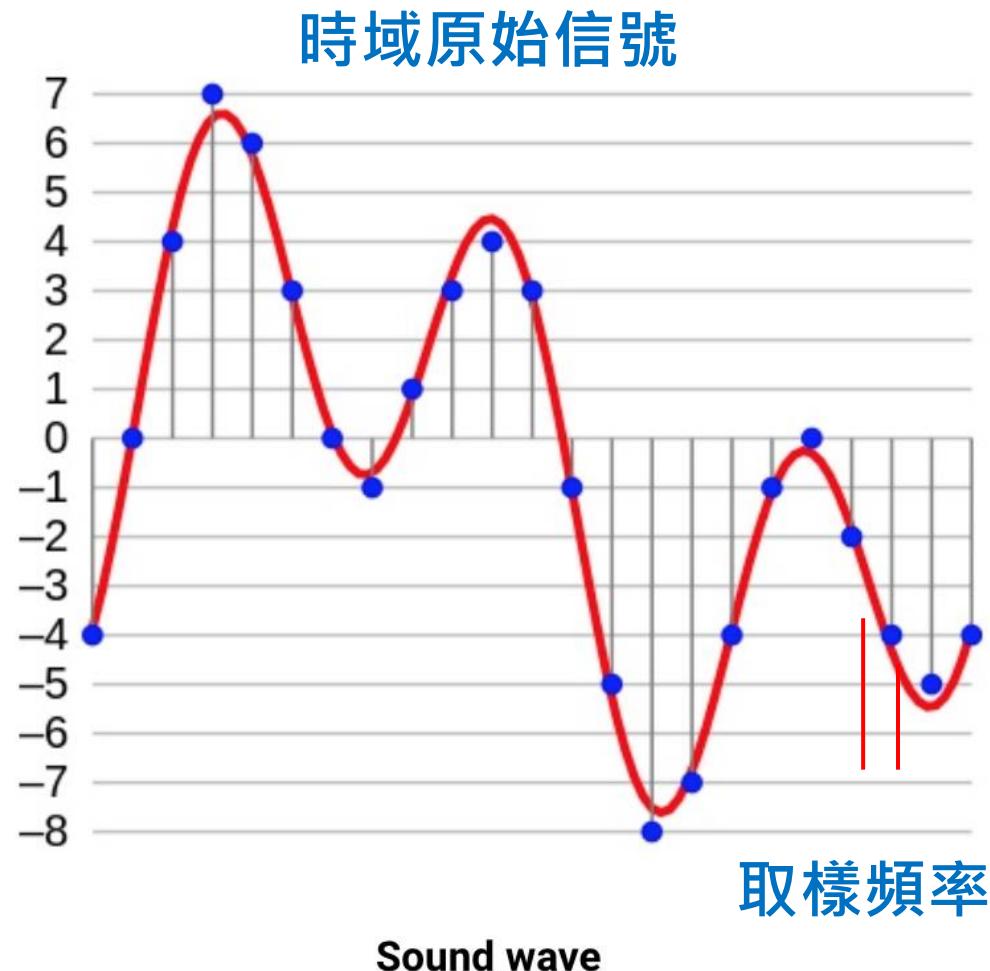


手機、電腦麥克風  
(檔案輸出)

# Xiao nRF52840 Sense 模組 – 參考電路



# 聲音訊號—取樣、量化



資料來源：<https://mikesmales.medium.com/sound-classification-using-deep-learning-8bc2aa1990b7>

**量化數位資料**

( 量化位元數 8 / 12 / 16 bit)  
 ( 量化階數 256 / 4096 / 65536)



[ -4, 0, 4, 7, 6, 3, 0, -1, 1, 3, 4, 3,  
 -1, -5, -8, -7, -4, -1, 0, -2, -4, -5, -4 ]

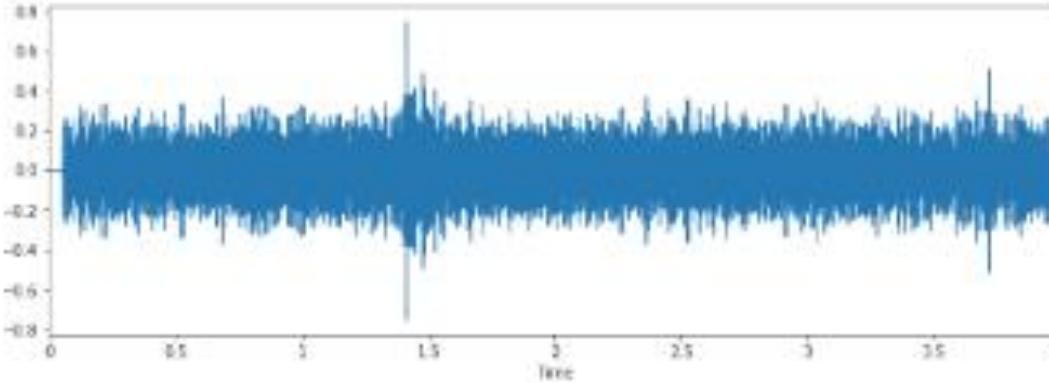
**資料大小 = 取樣頻率 \* 量化位元數**

人耳 / CD 品質  
 (1K~16K / 44.1 KHz)

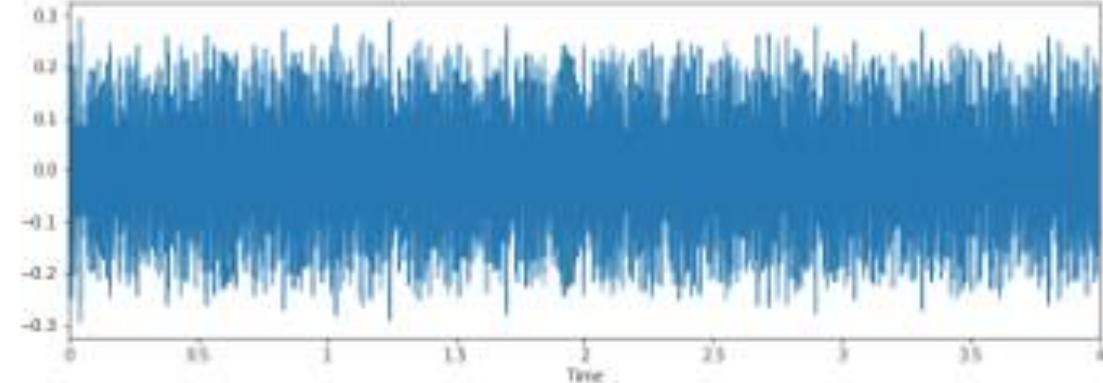
**Array**

# 常見聲音訊號波形（時間域）

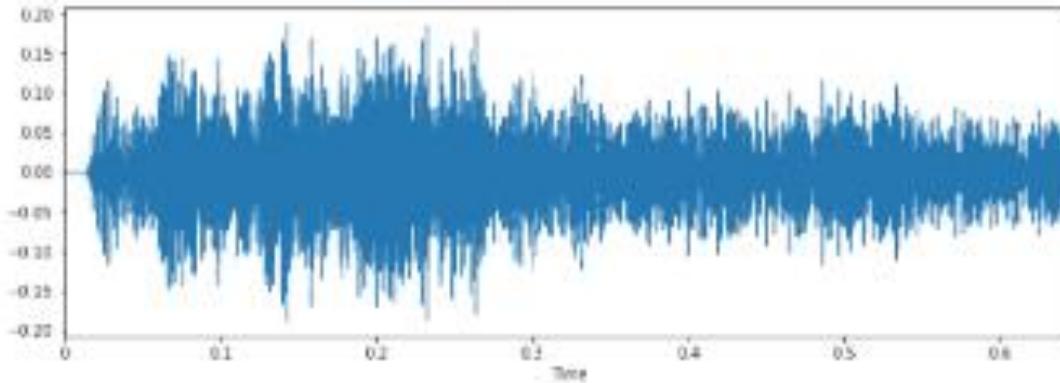
Air Conditioner



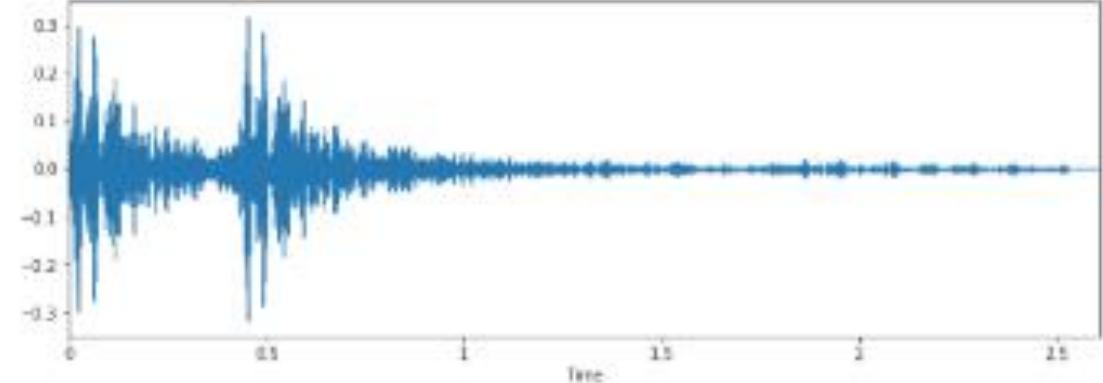
Engine Idling



Car horn



Gunshot



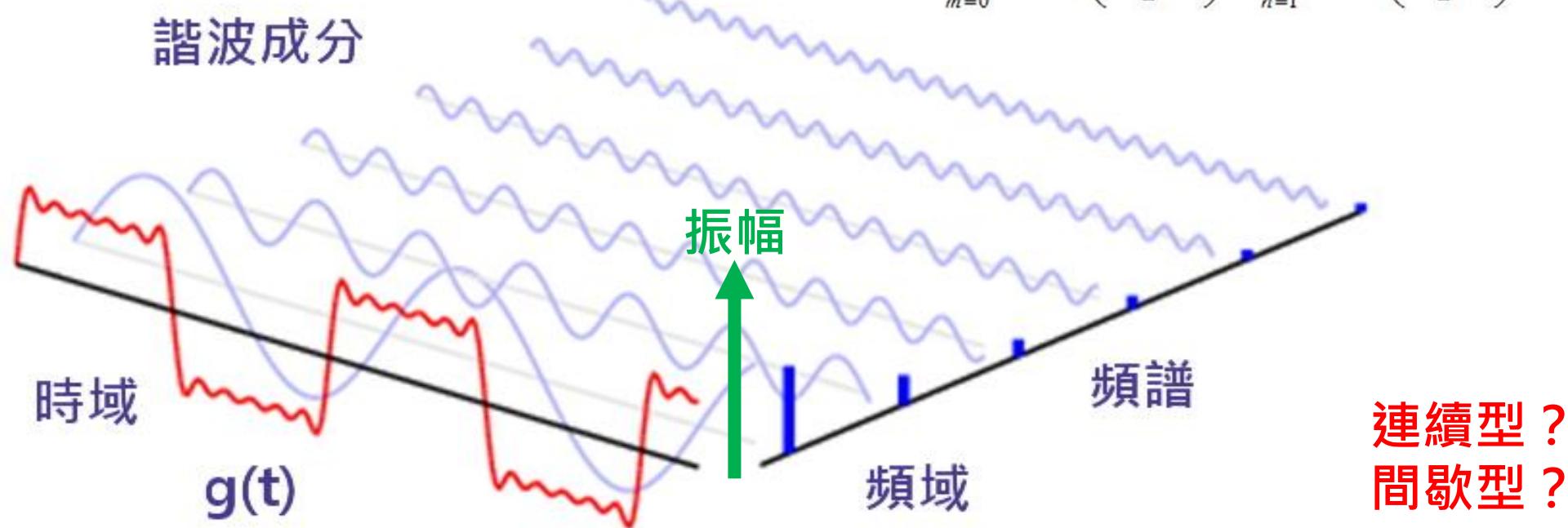
資料來源：<https://mikesmales.medium.com/sound-classification-using-deep-learning-8bc2aa1990b7>

# 特徵提取 – 時間域、頻率域轉換

**傅立葉級數（轉換）**  
所有信號都可拆解為不同頻率弦波之組合

$$g(t) = a_0 + \sum_{m=1}^{\infty} a_m \cos\left(\frac{2\pi mt}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi nt}{T}\right)$$

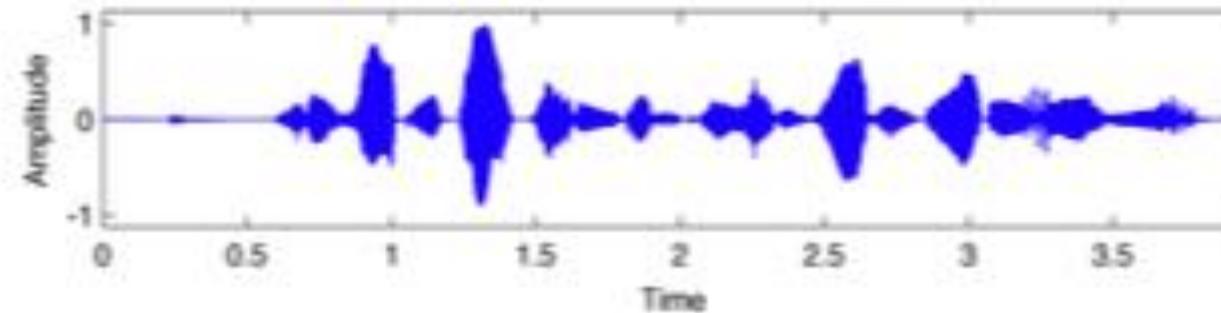
$$= \sum_{m=0}^{\infty} a_m \cos\left(\frac{2\pi mt}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi nt}{T}\right)$$



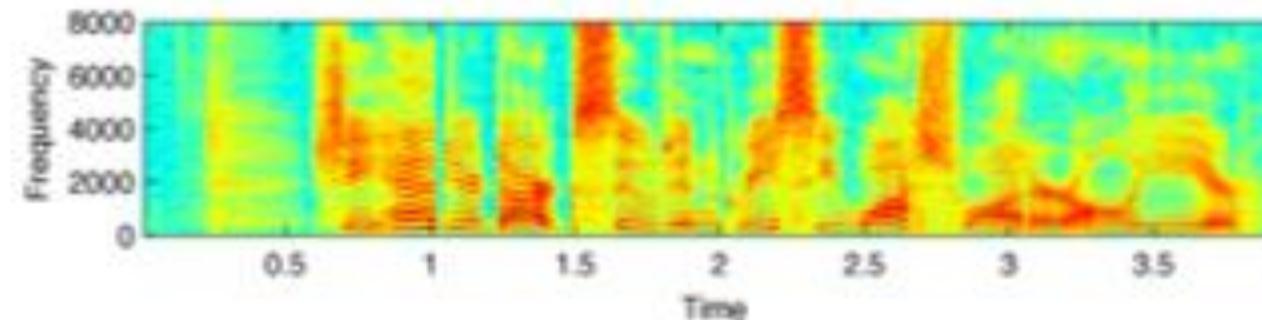
資料來源：[https://yhuang1966.blogspot.com/2019/12/praat\\_5.html](https://yhuang1966.blogspot.com/2019/12/praat_5.html)

# 特徵提取 – 線性頻譜、梅爾倒頻譜表示法

**Time Domain  
Waveform**

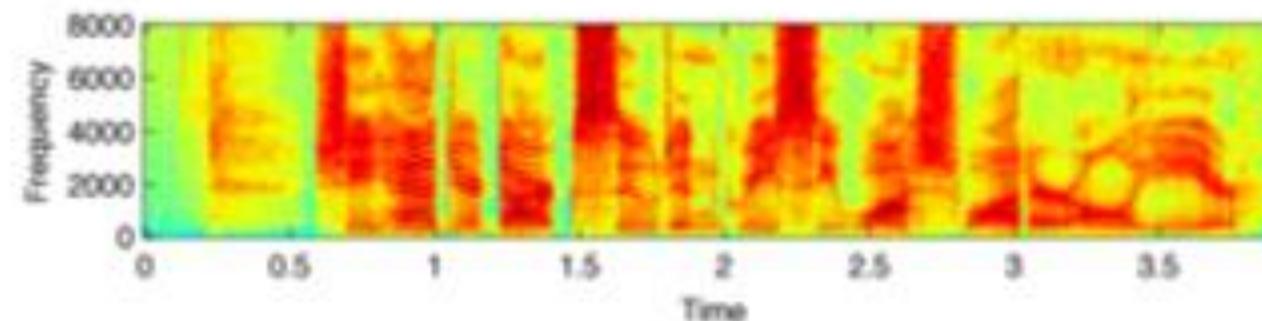


**線性  
頻譜  
Spectrogram**



**對數  
頻譜  
MFCC  
Spectrogram**

Mel-Frequency Cepstral  
Coefficients ( MFCC )



資料來源：<https://mikesmales.medium.com/sound-classification-using-deep-learning-8bc2aa1990b7>

# Google Speech Commands 公開資料集

共有常見語音命令35種，每筆資料1秒，包含數字、移動方向、常見動物及人名。曾多次釋出不同版本 v0.0.1, v0.0.2, v0.0.3，2 ~ 3GB。

資料集包含不同性別、各種口音及速度因此很適合用於一般智慧家庭應用。

實驗時為令訓練時間縮短可擷取較少數量進行訓練，待模型推論精度不足後再提升數量，以免浪費太多時間。

[https://www.tensorflow.org/datasets/catalog/speech\\_commands](https://www.tensorflow.org/datasets/catalog/speech_commands)

## Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition

Word	Number of Utterances		
Backward	1,664	Nine	3,934
Bed	2,014	No	3,941
Bird	2,064	Off	3,745
Cat	2,031	On	3,845
Dog	2,128	One	3,890
Down	3,917	Right	3,778
Eight	3,787	Seven	3,998
Five	4,052	Sheila	2,022
Follow	1,579	Six	3,860
Forward	1,557	Stop	3,872
Four	3,728	Three	3,727
Go	3,880	Tree	1,759
Happy	2,054	Two	3,880
House	2,113	Up	3,723
Learn	1,575	Visual	1,592
Left	3,801	Wow	2,123
Marvin	2,100	Yes	4,044
		Zero	4,052

資料來源：<https://arxiv.org/abs/1804.03209>

# ESC-50 環境音公開資料集

常見環境音，共分五大類（動物、自然聲及水聲、人聲及非講話聲、室內聲、戶外聲及都市雜音），每個分類又有十個小類，共五十分類。每個小類有四十個樣本，合計有兩千個樣本。

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

## ESC-50: Dataset for Environmental Sound Classification

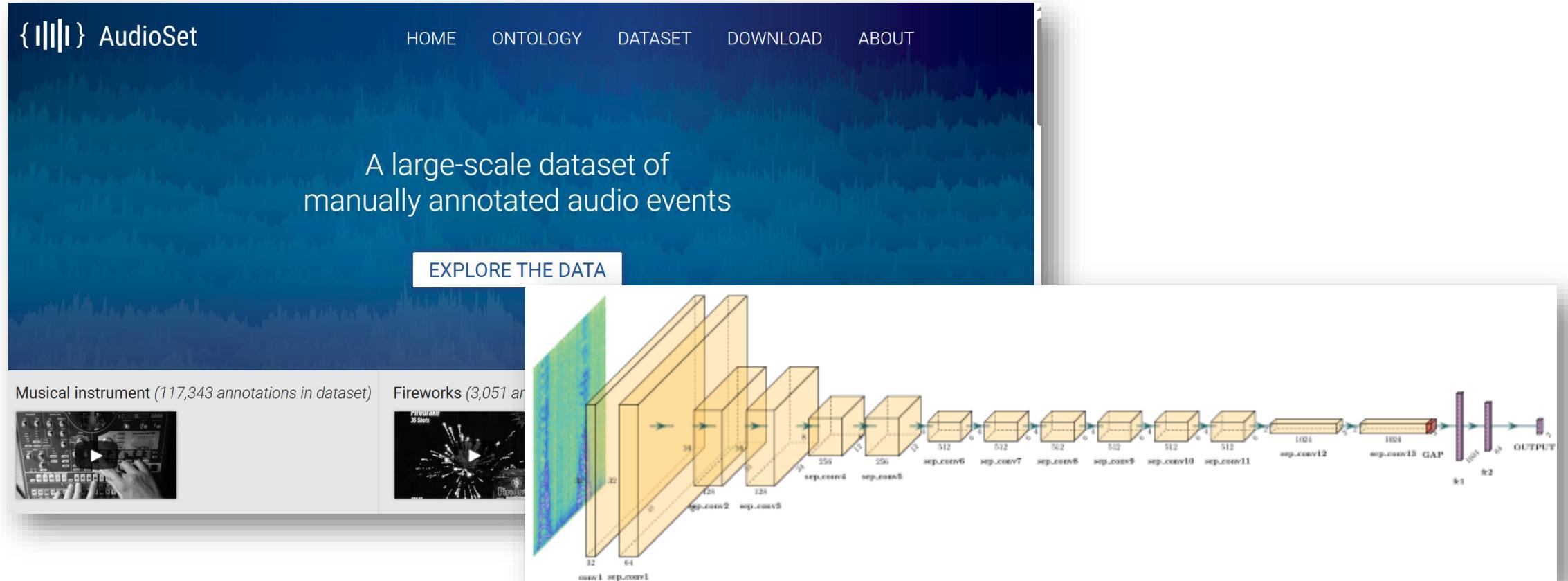
dog - 5-231762-A-0.wav



<https://github.com/karolpiczak/ESC-50>

# Google AudioSet 聲音公開資料集

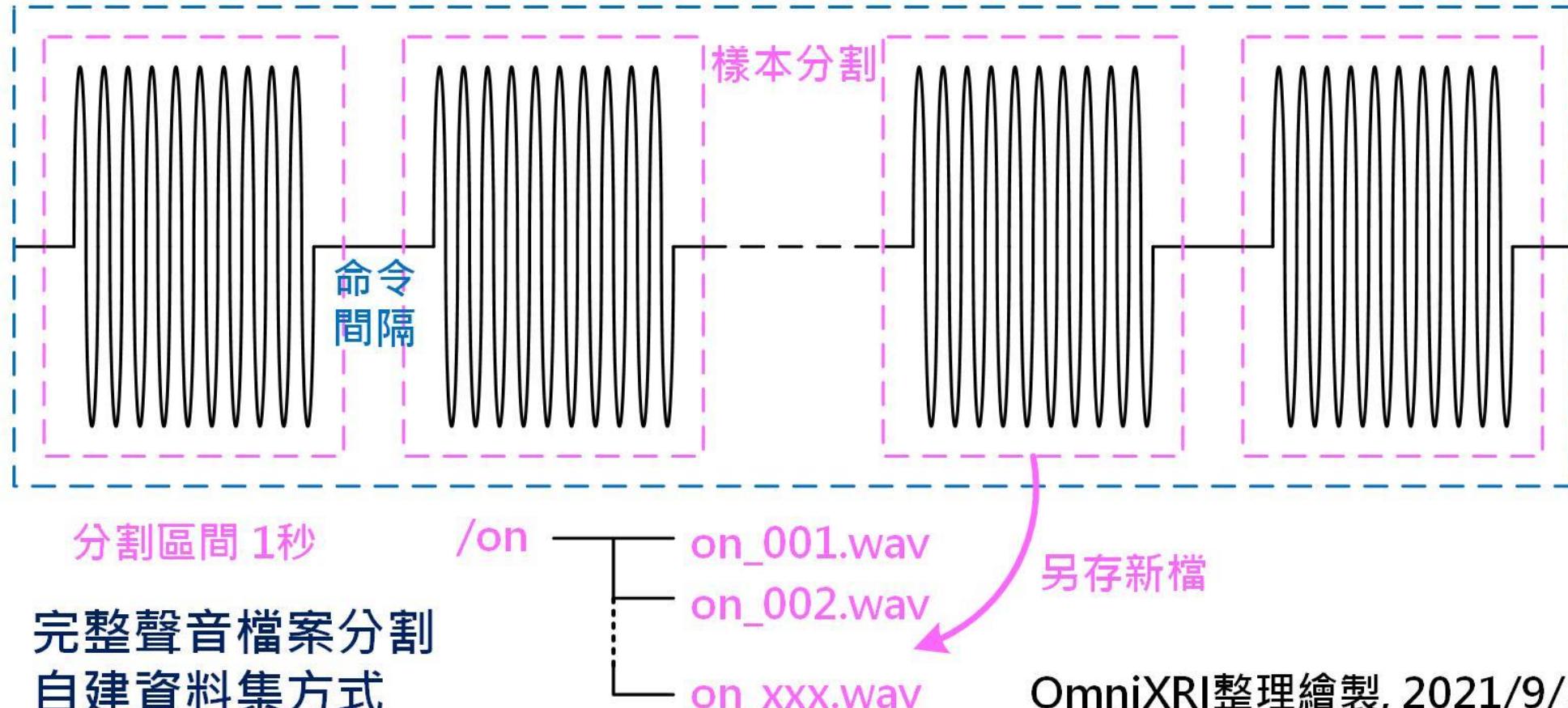
從Youtube中收集512類常見生活中聲音，可搭配 YAMNet 進行聲音分類。



<https://research.google.com/audioset/>

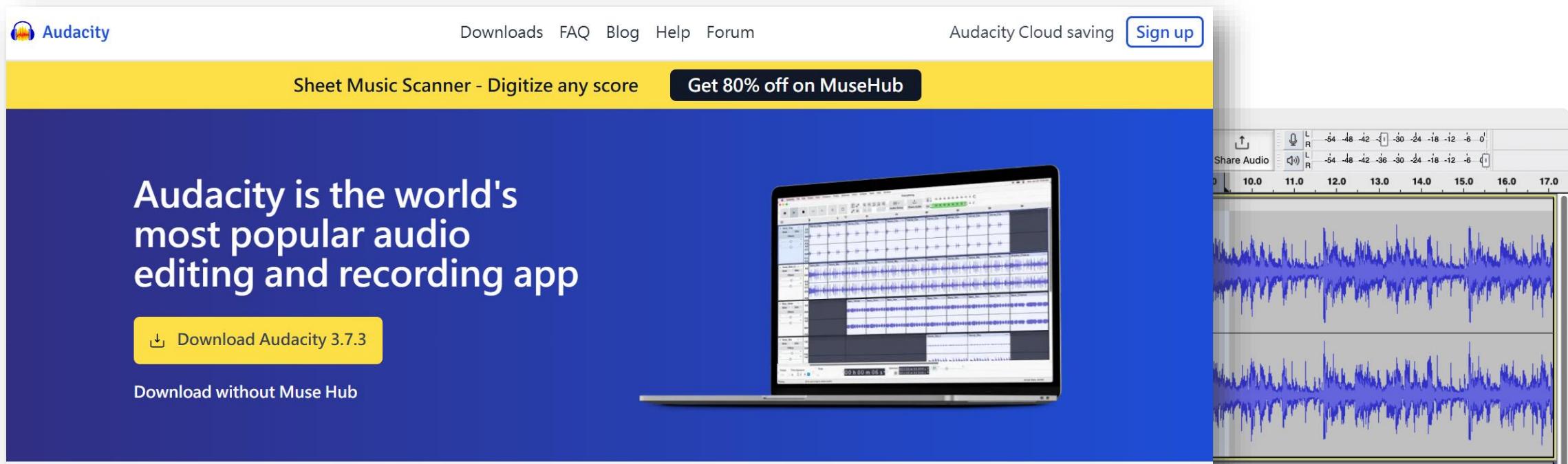
# 自定義語音命令錄音技巧

完整聲音錄製檔案 (on.wav) – 建議取樣頻率 16KHz · 以非壓縮WAV格式儲存。



資料來源：<https://github.com/OmniXRI/iThomeIronMan2021/blob/main/Day17.md>

# 聲音分割開源工具 Audacity



Audacity is the world's most popular audio editing and recording app

Download Audacity 3.7.3

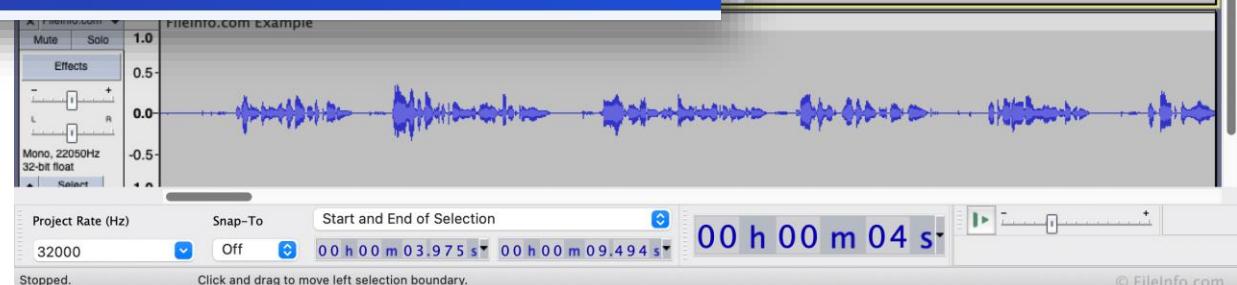
Download without Muse Hub

Sheet Music Scanner - Digitize any score Get 80% off on MuseHub

Audacity Cloud saving Sign up

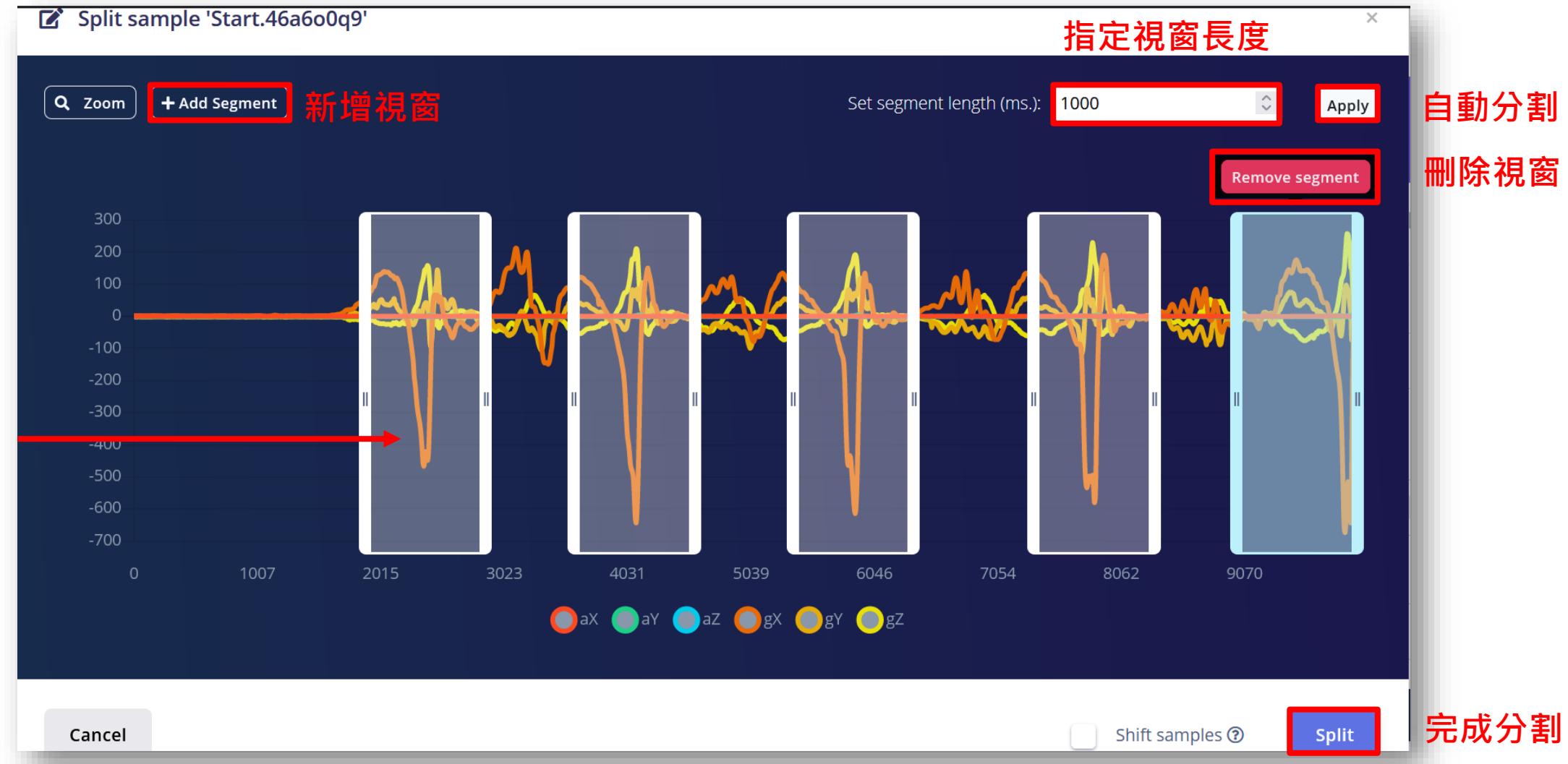
The screenshot shows the Audacity website homepage. It features a yellow header bar with links for Downloads, FAQ, Blog, Help, Forum, and a 'Sign up' button for Audacity Cloud saving. Below the header is a banner for 'Sheet Music Scanner - Digitize any score' with a 'Get 80% off on MuseHub' button. The main content area has a dark blue background with white text stating 'Audacity is the world's most popular audio editing and recording app'. It includes a yellow button to 'Download Audacity 3.7.3' and a link to 'Download without Muse Hub'. To the right, there's a preview of the Audacity software interface showing multiple tracks of audio waveform and various editing tools. At the bottom, there's another smaller screenshot of the software interface.

可用於聲音錄音、編輯、分割、合併、混音等。



<https://www.audacityteam.org/>

# Edge Impulse 連續信號自動分割工具





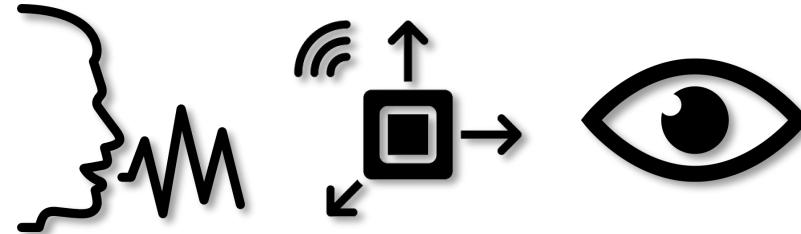
## 8.3. TensorFlow Lite 模型訓練

# TinyML 開發流程選項

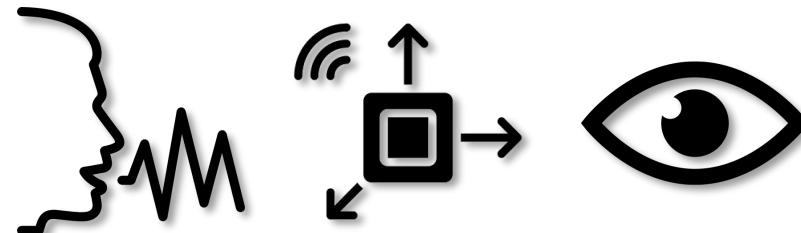
## 模型訓練工具名稱



**TensorFlow Lite**  
(現更名LiteRT)



**EDGE  
IMPULSE**



**SenseCraft**

## 適用感測器

## 主要特色

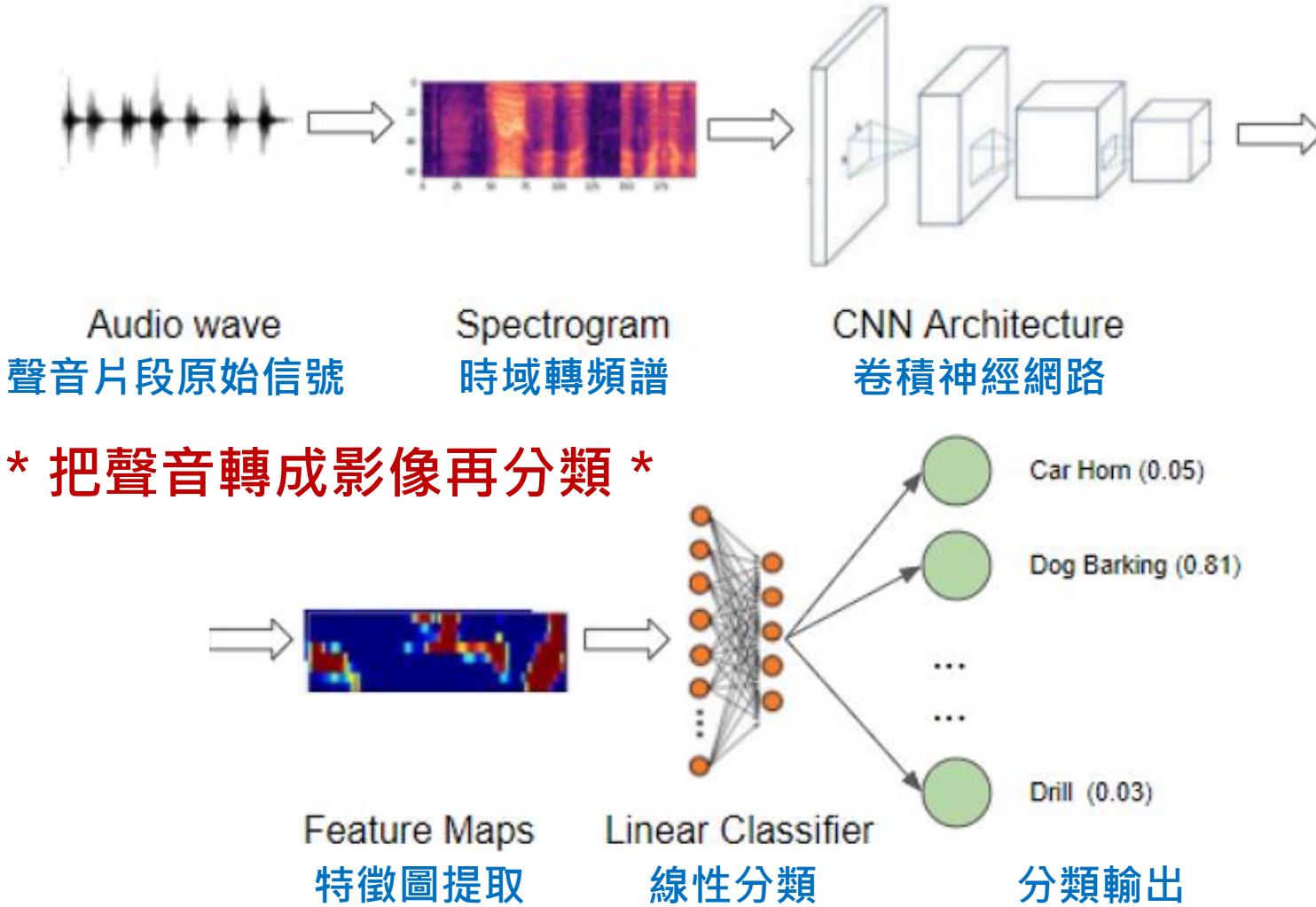
免費版：資料集需自行建置，僅提供模型訓練及轉成 **TensorFlow Lite for Microcontroller** 格式。可利用雲端 Google Colab 或本地端 Jupyter Notebooks 進行模型訓練及部署。

免費版：可支援各式感測器及開發板，提供一定額度資料集建置、資料前處理（特徵提取）模型訓練、模型優化及多種部署方式。亦可導入已訓練好的 **TensorFlow Lite, ONNX** 等模型。



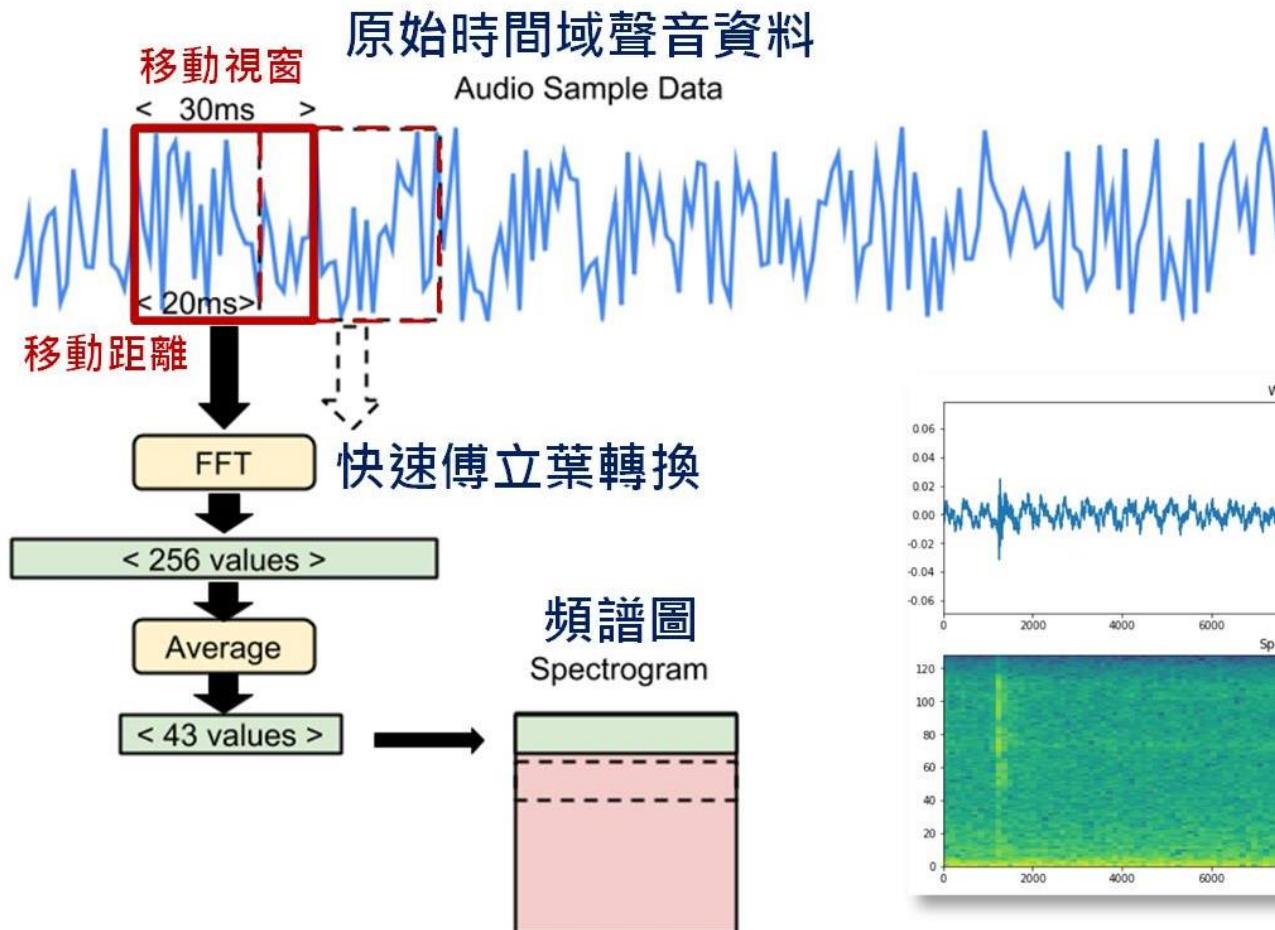
免費版：影像分類可多分類、物件偵測同一模型僅能一類。事件觸發只能設一種條件。可支援雲端資料收集、模型訓練及部署，亦導入外部已訓練好之 **TensorFlow Lite** 模型。

# 深度學習模型聲音分類

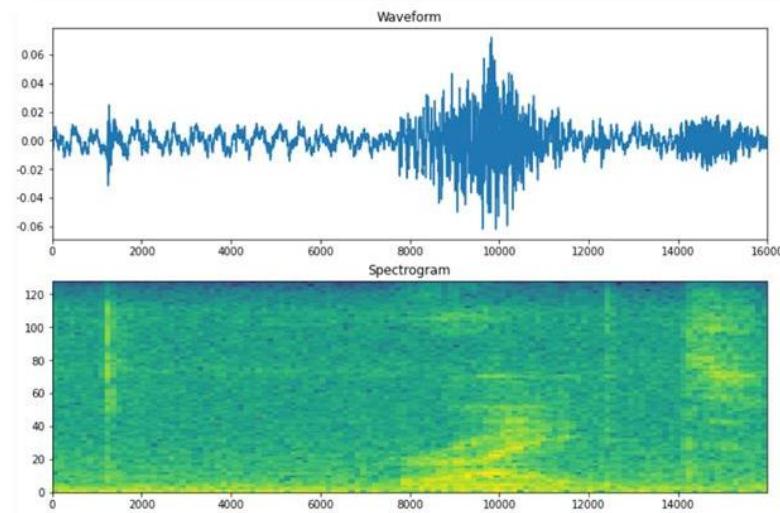


資料來源：<https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5>

# 關鍵詞偵測(KWS) 特徵提取



TensorFlow Lite Micro  
Micro Speech  
聲音頻譜圖轉換範例



時間域格式

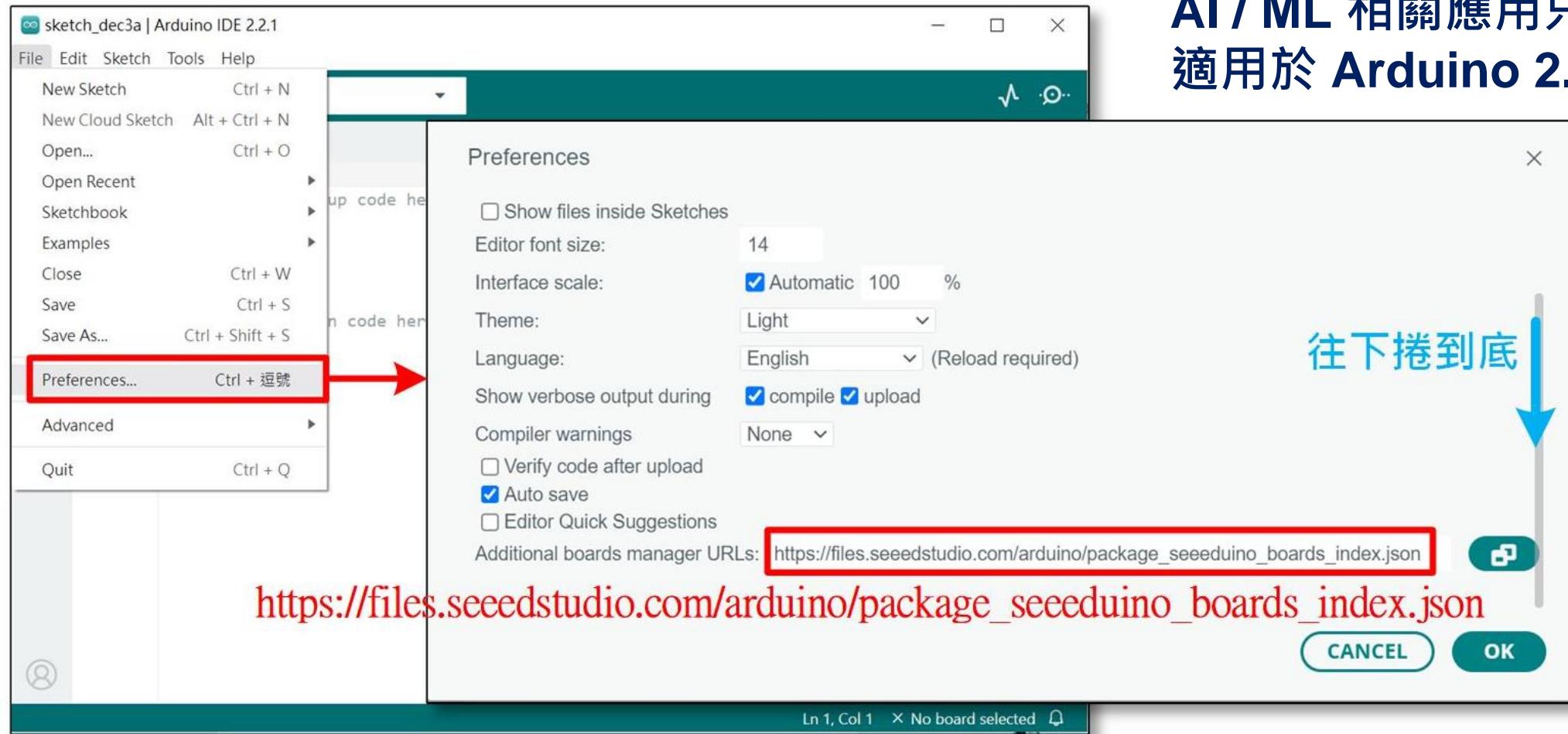
頻譜圖格式

OmniXRI整理繪製, 2021/9/30

資料來源：<https://github.com/OmniXRI/iThomeIronMan2021/blob/main/Day17.md>

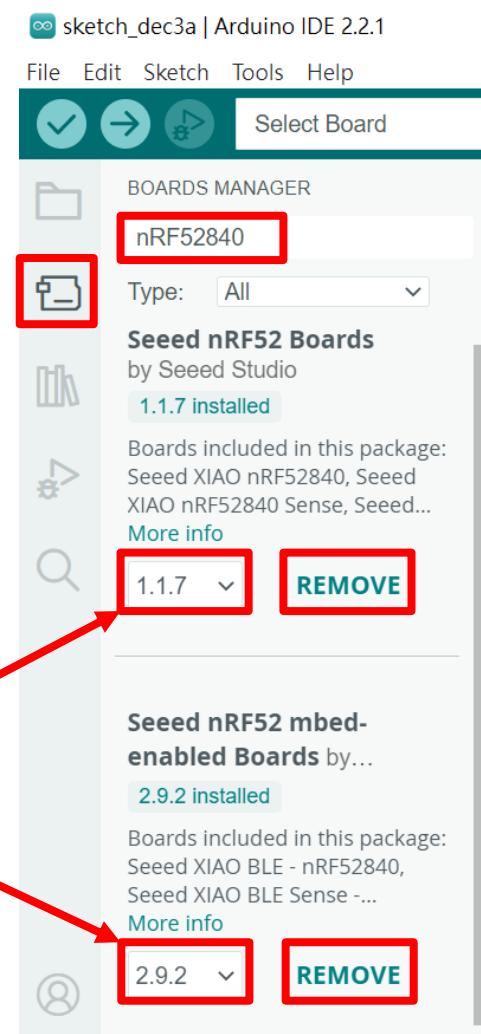
# Arduino 新增開發板設定

AI / ML 相關應用只  
適用於 Arduino 2.x 。



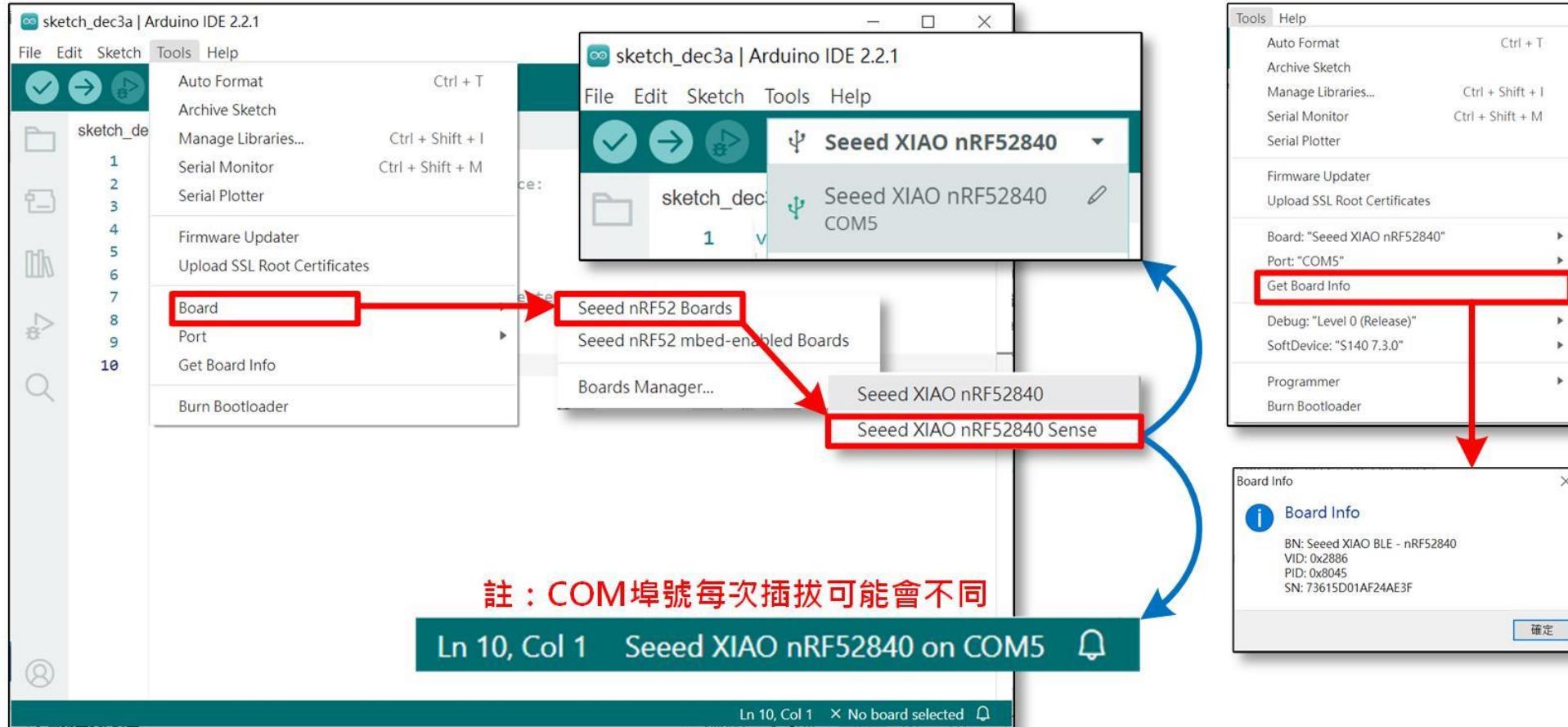
OmniXRI 整理製作, 2023/12/08

# 安裝Seeed nRF52840函式庫



- 點選選單 Tools > Board > Boards Manager...，或直接點選左側第二個「開發板」圖示。
- 輸入 **nRF52840** 搜尋 Seeed nRF52840 開發板相關函式庫。
- 點選「**INSTALL**」安裝下列二個函式庫。（版本可取最新的）
  - **Seeed nRF52 Boards ( BLE, 低功耗功能 )**
  - **Seeed nRF52 mbed-enabled Boards ( PDM, IMU, ML )**
- 安裝後若不需要時，可點選「**REMOVE**」解除安裝。

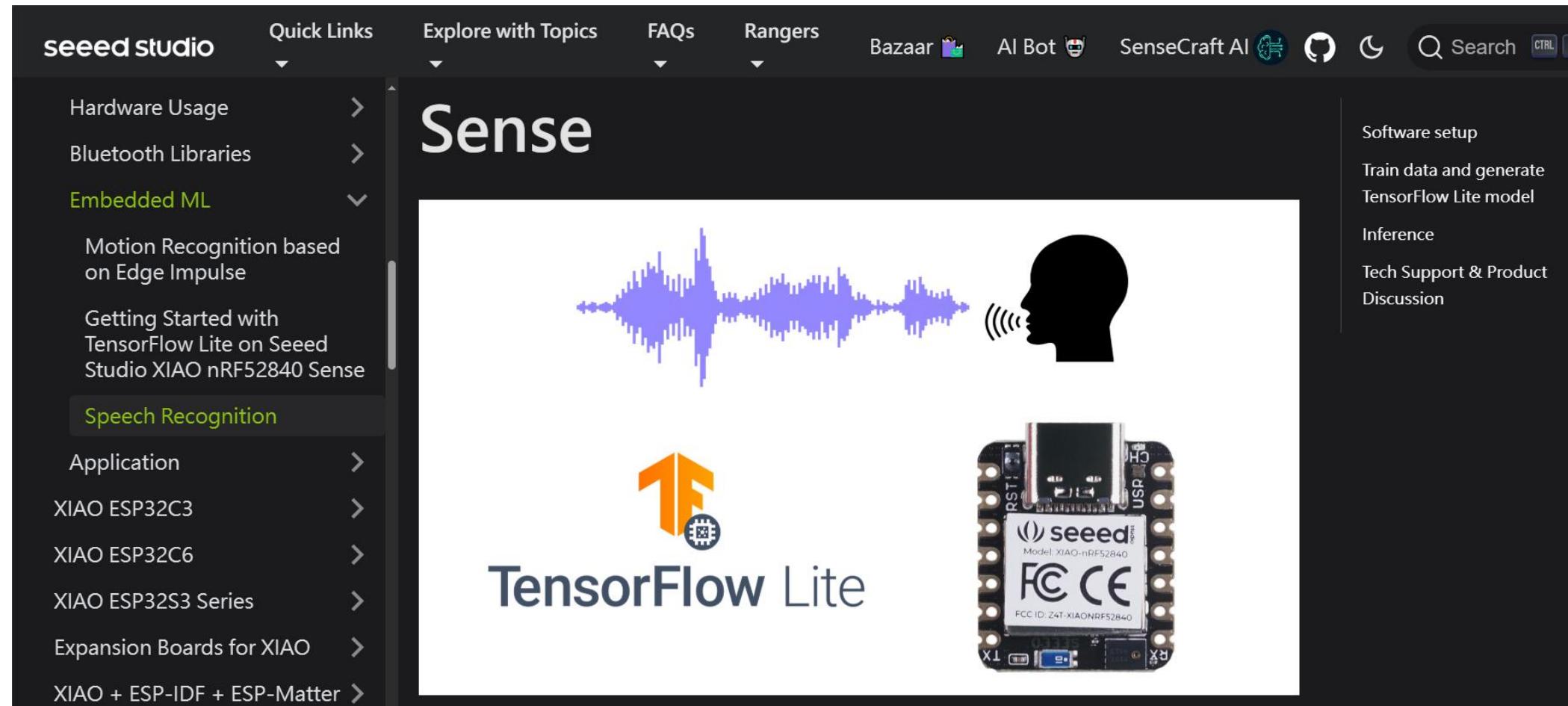
# 指定工作開發板及埠號



- 選擇開發板  
Seeed XIAO  
nRF52840  
Sense
- 選擇對應埠  
號
- 檢查是否連  
線

OmniXRI 整理製作, 2023/12/08

# Xiao nRF52840 Sense 語音辨識

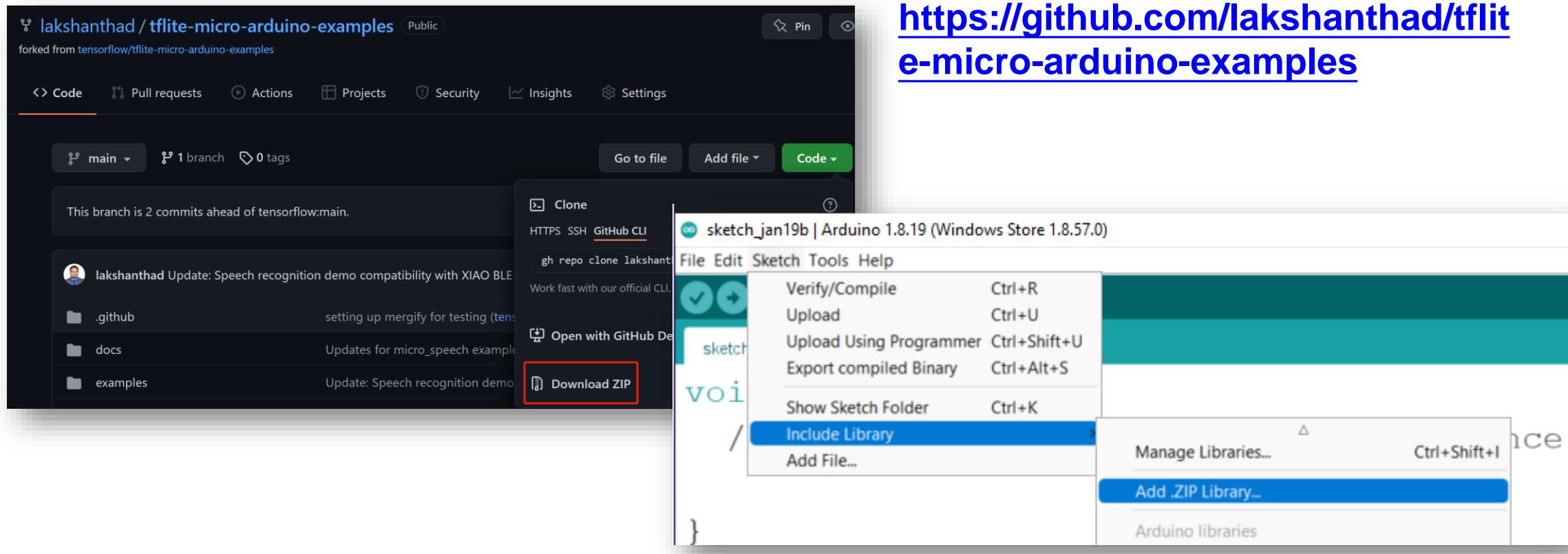


The screenshot shows the Seeed Studio Sense page. The left sidebar has a 'Speech Recognition' section highlighted. The main content area features a sound wave graphic, a profile of a human head with sound waves, the TensorFlow Lite logo, and an image of the XIAO nRF52840 Sense development board.

<https://wiki.seeedstudio.com/XIAO-BLE-Sense-TFLite-Mic/>

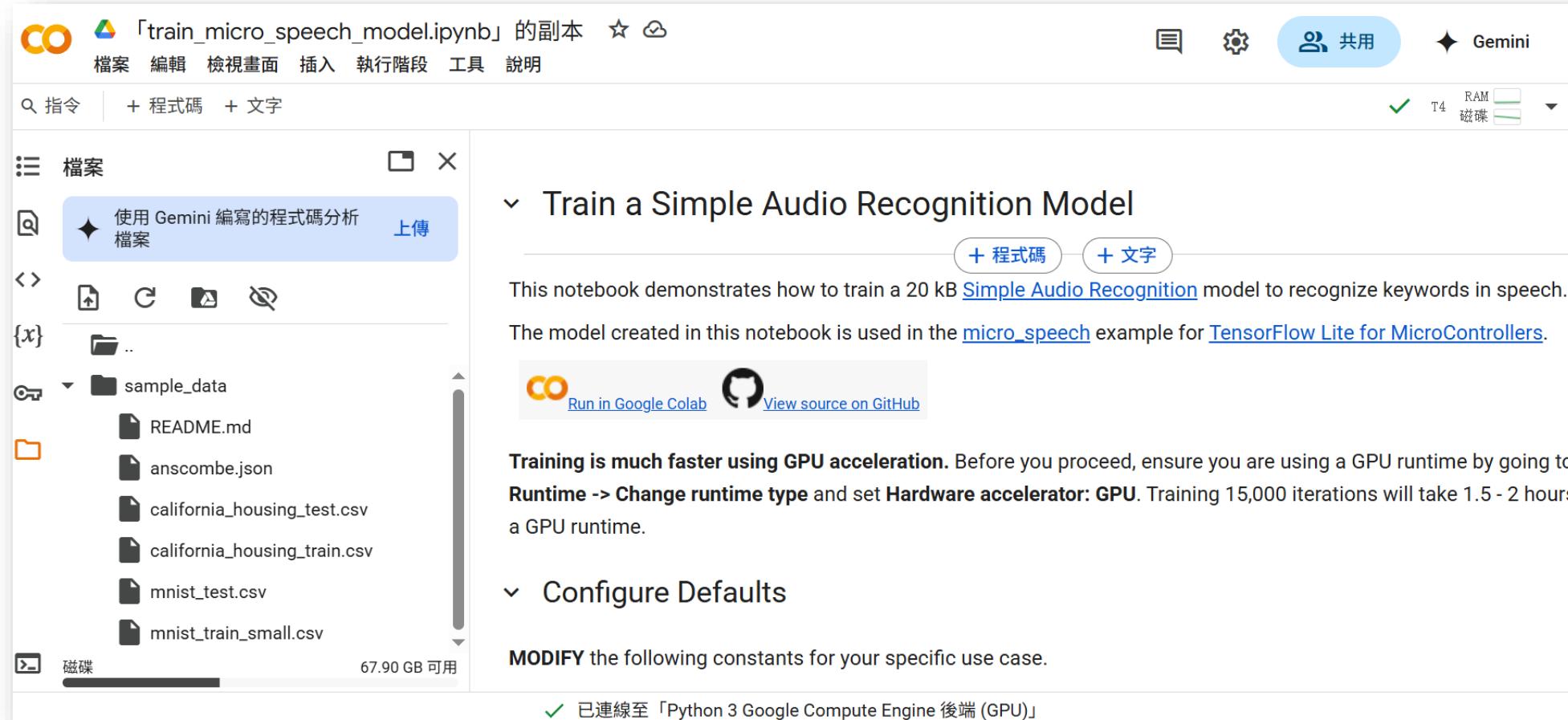
# Arduino 預安裝模組

下載 **tflite-micro-arduino-examples** library (\*.zip) 後，再安裝到 Arduino 函式庫中。



資料來源：<https://wiki.seeedstudio.com/XIAO-BLE-Sense-TFLite-Mic/>

# 啟動 Google Colab 模型訓練範例



「train\_micro\_speech\_model.ipynb」的副本

檔案 編輯 檢視畫面 插入 執行階段 工具 說明

共用 Gemini

T4 RAM 磁碟

指令 程式碼 文字

檔案

使用 Gemini 編寫的程式碼分析 檔案 上傳

sample\_data

- README.md
- anscombe.json
- california\_housing\_test.csv
- california\_housing\_train.csv
- mnist\_test.csv
- mnist\_train\_small.csv

67.90 GB 可用

已連線至「Python 3 Google Compute Engine 後端 (GPU)」

## Train a Simple Audio Recognition Model

This notebook demonstrates how to train a 20 kB [Simple Audio Recognition](#) model to recognize keywords in speech. The model created in this notebook is used in the [micro\\_speech](#) example for [TensorFlow Lite for MicroControllers](#).

[Run in Google Colab](#) [View source on GitHub](#)

Training is much faster using GPU acceleration. Before you proceed, ensure you are using a GPU runtime by going to Runtime -> Change runtime type and set Hardware accelerator: GPU. Training 15,000 iterations will take 1.5 - 2 hours a GPU runtime.

## Configure Defaults

MODIFY the following constants for your specific use case.

[https://colab.research.google.com/github/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/examples/micro\\_speech/train/train\\_micro\\_speech\\_model.ipynb](https://colab.research.google.com/github/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/examples/micro_speech/train/train_micro_speech_model.ipynb)

# 修正 Colab 範例訓練參數

## ▼ Configure Defaults

MODIFY the following constants for your specific use case.

挑選所需語音命令，一開始可只選擇二種加速模型訓練

```
[ ] # A comma-delimited list of the words you want to "train" to "right", "on", "off", "stop", "go",  
# The options are: yes, no, up, down, left, right, zero, one, two, three, four,  
# All the other words will be used to "train" an "unknown" label and silent  
# audio data with no spoken words will be used to train a "silence" label  
WANTED_WORDS = "yes, no"  
  
# The number of steps and learning rates can be specified as comma-separated  
# lists to define the rate at each stage. For example,  
# TRAINING_STEPS=12000, 3000 and LEARNING_RATE=0.001, 0.0001  
# will run 12,000 training loops in total, with a rate of 0.001 for the first  
# 8,000, and 0.0001 for the final 3,000.  
TRAINING_STEPS = "12000, 3000"  
LEARNING_RATE = "0.001, 0.0001"
```

訓練次數可降至1200, 300 加速模型訓練，若推論效果不佳再往上加

# Colab 開始執行模型訓練



訓練時長依設定參數可能一到數個小時  
**加速方式：**

- 減少種類**
- 減少每個種類樣本數**
- 減少訓練次數**

learning rates can be specified as comma-separated  
e at each stage. For example,  
and LEARNING\_RATE=0.001, 0.0001  
loops in total, with a rate of 0.001 for the first  
he final 3,000.

er of steps, which is used to identify the checkpoint

# Colab 完成模型訓練

## ▼ Deploy to a Microcontroller

Follow the instructions in the [micro\\_speech](#) README.md for [TensorFlow Lite for MicroControllers](#) to deploy this model on a specific microcontroller.

**Reference Model:** If you have not modified this notebook, you can follow the instructions as is, to deploy the model. Refer to the [micro\\_speech/train/models](#) directory to access the models generated in this notebook.

**New Model:** If you have generated a new model to identify different words: (i) Update `kCategoryCount` and `kCategoryLabels` in [micro\\_speech/micro\\_features/micro\\_model\\_settings.h](#) and (ii) Update the values assigned to the variables defined in [micro\\_speech/micro\\_features/model.cc](#) with values displayed after running the following cell.

```
# Print the C source file
!cat {MODEL_TFLITE_MICRO}

✉ unsigned char g_model[] = {
    0x20, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
    0x14, 0x00, 0x20, 0x00, 0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00,
    0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00,
    0x1c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x9c, 0x00, 0x00, 0x00,
    0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x0c, 0x00, 0x0c, 0x00,
    0xb, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x0c, 0x00, 0x00, 0x00,
    0x16, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x16
};

unsigned int g_model_len = 19160;
```

複製此段程式（網路結構、權重及數量）到  
**tflite-micro-arduino-examples library**

**unsigned int g\_model\_len = 19160;**



## 8.4. Arduino模型部署與測試

# 更新函式庫 (1/2)

This PC > Documents > Arduino > libraries > tflite-micro-arduino-examples-main > examples > micro\_speech

Name	Date modified	Type	Size
data	2/23/2022 10:21 AM	File folder	
arduino_audio_provider	2/23/2022 10:21 AM	C++ Source File	7 KB
arduino_command_responder	2/23/2022 10:21 AM	C++ Source File	3 KB
arduino_main	2/23/2022 10:21 AM	C++ Source File	1 KB
audio_provider	2/23/2022 10:21 AM	C Header Source F...	3 KB
command_responder	2/23/2022 10:21 AM	C Header Source F...	2 KB
feature_provider	2/23/2022 10:21 AM	C++ Source File	6 KB
feature_provider	2/23/2022 10:21 AM	C Header Source F...	3 KB
main_functions	2/23/2022 10:21 AM	C Header Source F...	2 KB
micro_features_micro_features_generator	2/23/2022 10:21 AM	C++ Source File	5 KB
micro_features_micro_features_generator	2/23/2022 10:21 AM	C Header Source F...	2 KB
micro_features_micro_model_settings	2/23/2022 10:21 AM	C++ Source File	1 KB
micro_features_micro_model_settings	2/23/2022 10:21 AM	C Header Source F...	3 KB
micro_features_model	2/23/2022 10:21 AM	C++ Source File	118 KB
micro_features_model	2/23/2022 10:21 AM	C Header Source F...	2 KB
micro_speech	2/23/2022 10:21 AM	INO File	8 KB

- 找到函式庫原始路徑 Documents > Arduino > libraries > tflite-micro-arduino-examples
- 進到 examples > micro\_speech
- 開啟 micro\_features\_model.cpp
- 將剛才訓練好的模型覆蓋掉。

```
const unsigned char g_model[] DATA_ALIGN_ATTRIBUTE = {
    0x20, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x12, 0x00, 0x1c, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00,
    0x10, 0x00, 0x14, 0x00, 0x00, 0x00, 0x18, 0x00, 0x12, 0x00, 0x00, 0x00,
```

- 修改模型長度：
- const int g\_model\_len = 19160;**

# 更新函式庫 (2/2)

➤ **修改類別種類**：開啟 **micro\_features\_micro\_model\_settings.cpp**

```
#include "micro_features_micro_model_settings.h"
```

```
const char* kCategoryLabels[kCategoryCount] = {  
    "silence",  
    "unknown",  
    "yes",  
    "no",  
};
```

➤ **修改類別數量**：開啟 **micro\_features\_micro\_model\_settings.h**

```
constexpr int kCategoryCount = 4;
```

# NO Code 測試結果

```
Initialization complete  
Heard yes (205) @22400ms  
Heard yes (212) @29920ms  
Heard no (202) @84400ms
```

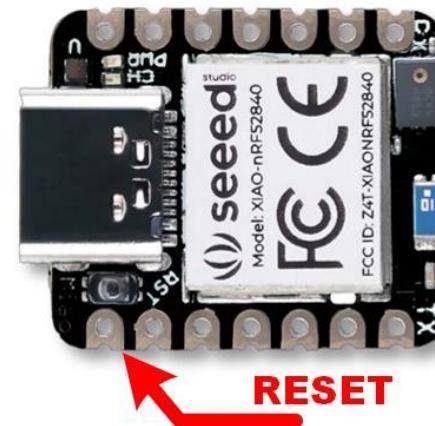


## No Code

Virtual COM 輸出字串結果，不用改動任何一行程式碼。可令PC端或另一個MCU UART接收到結果後再自行解譯。

## Low Code

如有必要可微調幾行驅動MCU上實體GPIO或其它週邊裝置。



OmniXRI整理製作, 2024/05/09

# Arduino 2.0 快速(增量)編譯



**非常重要：**  
**強烈建議一定要採用Arduino CLI 增量編譯方式更新程式修改，否則每次將耗費大量時間，增加碳排量。**

工作內容	Arduino IDE	Arduino CLI
第一次編譯上傳	19~20min	19~20min
未修改程式第二次編譯上傳	19~20min	< 1min
修改程式後編譯上傳	19~20min	1~2min

<https://omnixri.blogspot.com/2024/10/arduino-20.html>

# 參考文獻

- 許哲豪，臺灣科技大學資訊工程系「人工智慧與邊緣運算實務」（2021~2023）  
<https://omnixri.blogspot.com/p/ntust-edge-ai.html>
- 許哲豪，OmniXRI's Edge AI & TinyML 小學堂 Youtube 直播課程總結  
<https://omnixri.blogspot.com/2024/06/omnixris-edge-ai-tinyml-youtube.html>
- 許哲豪，歐尼克斯實境互動工作室系列發文—TinyML(MCU AI)系列  
<https://hackmd.io/1PK1URhIQ7GutcWgpgsWbg#TinyMLMCU-AI%E7%B3%BB%E5%88%97>
- 許哲豪，如何讓 Arduino 2.0 快速編譯（增量編譯）  
<https://omnixri.blogspot.com/2024/10/arduino-20.html>

# 延伸閱讀

---

- Seeed, Xiao nRF52840 Sense – Embedded ML - Speech Recognition on Seeed Studio XIAO nRF52840 Sense

<https://wiki.seeedstudio.com/XIAO-BLE-Sense-TFLite-Mic/>

- Github, lakshanthad / tflite-micro-arduino-examples

<https://github.com/lakshanthad/tflite-micro-arduino-examples>

- Google, Colab Example - train\_micro\_speech\_model.ipynb

[https://colab.research.google.com/github/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/examples/micro\\_speech/train/train\\_micro\\_speech\\_model.ipynb](https://colab.research.google.com/github/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/examples/micro_speech/train/train_micro_speech_model.ipynb)

- Google, Speech Commands

[https://www.tensorflow.org/datasets/catalog/speech\\_commands](https://www.tensorflow.org/datasets/catalog/speech_commands)

沒有最邊



只有更邊



歡迎加入  
邊緣人俱樂部



**YOUTUBE 直播 :** <https://www.youtube.com/@omnixri1784streams>



歐尼克斯實境互動工作室  
(OmniXRI Studio)

許哲豪 (Jack Hsu )

[Facebook : Jack Omnidri](#)

[FB社團 : Edge AI Taiwan 邊緣智能交流區](#)

[電子郵件 : omnixri@gmail.com](#)

[部落格 : https://omnidri.blogspot.tw](#)

[開 源 : https://github.com/OmniXRI](#)