

CoreMaker-01



兩天自動關窗模擬

指導單位：
INDUSTRIAL DEVELOPMENT BOARD
INSTITUTE OF ECONOMIC AFFAIRS
經濟發展委員會

主辦單位： 財團法人資訊工業策進會
INSTITUTE FOR INFORMATION INDUSTRY

 物聯網智造基地
IOT SERVICE HUB

合作單位：智慧立方

大綱



Ch. 1
概述



Ch. 2
硬體介紹



Ch. 3
開發流程解說



Ch. 4
成果演示



第一章 概述

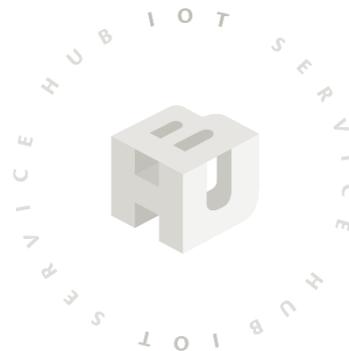
1-1. 文件說明



1-2. 情境應用概述



1-3. 案例示意圖



1-1. 文件說明

本文介紹CoreMaker-01這塊中光電智能感測所推出的板子以及它的應用，將介紹如何用CoreMaker-01板子、Arduino以及馬達來模擬下雨天時自動開關窗的情況。

第一章大概介紹此次案例和描述使用情景。而使用到的元件材料如CoreMaker-01板子規格、旋轉馬達、線路接線以及硬體操作則會在第二章說明。

因為此次範例用到了超音波感測器(GPIO)、溫濕度感測器(I2C) 以及伺服馬達(GPIO)與WE-I Plus板子相連、以及AI模型的使用，有提供相關的程式碼以及函式庫，可參考第三章來操作、建立或是修改程式碼達到使用者的要求，開發者也可以根據需求自行搭配上感測器元件來單獨控制WE-I Plus板子。

餵食器的主體將以紙盒來打造並於第四章以照片來呈現餵食器的成品以及操作。而在文末的附錄頁將提供AI模型、燒錄檔以及訓練模型照片資源的擷取，方便開發者使用。

1-2. 情境應用概述



此範例使用聲音為輸入，運用CoreMaker-01可應用邊緣運算的長處，讓機器學習模型來辨別是環境聲音或是下雨聲。

當使用者外出時，透過麥克風擷取外部環境聲音交給CoreMaker-01做運算，若辨識得下雨聲便啟動馬達旋轉模擬關閉門窗，避免家中進水增加濕氣影響家具。



1-3. 案例示意圖



聲音輸入



模型輸入運算

模型結果輸出

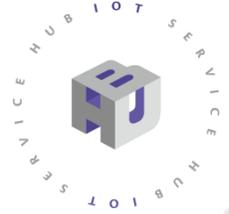
Arduino Uno



訊號給予



馬達轉動模擬開關窗



第二章 硬體介紹

2-1. CoreMaker-01 國產IC開發板簡介

2-2. CoreMaker-01 感測器位置 腳位

2-3. 範例所需材料

2-4. 外接元件介紹

2-5. 電路接線圖

2-6. CoreMaker-01腳位與元件說明

2-7. 硬體操作說明



2-1. CoreMaker-01 國產IC開發板簡介



- 尺寸 56mm(H) x 45mm(W)
- 使用 Nuvoton M484SIDAE 晶片 (Arm® Cortex®-M4F)
- 最高 192MHz 處理器時脈
- 提供完整開發包及文件
- 支援 I2C、UART、SPI、USB、SD 介面傳輸
- microUSB 接頭用來與電腦進行序列通訊以及接受外部供電給 CoreMaker-01
- 可應用 SensiML 作為機器學習的模型開發平台

2-2. CoreMaker-01 感測器位置 腳位

接口

- microUSB
- UART



- 外接電池

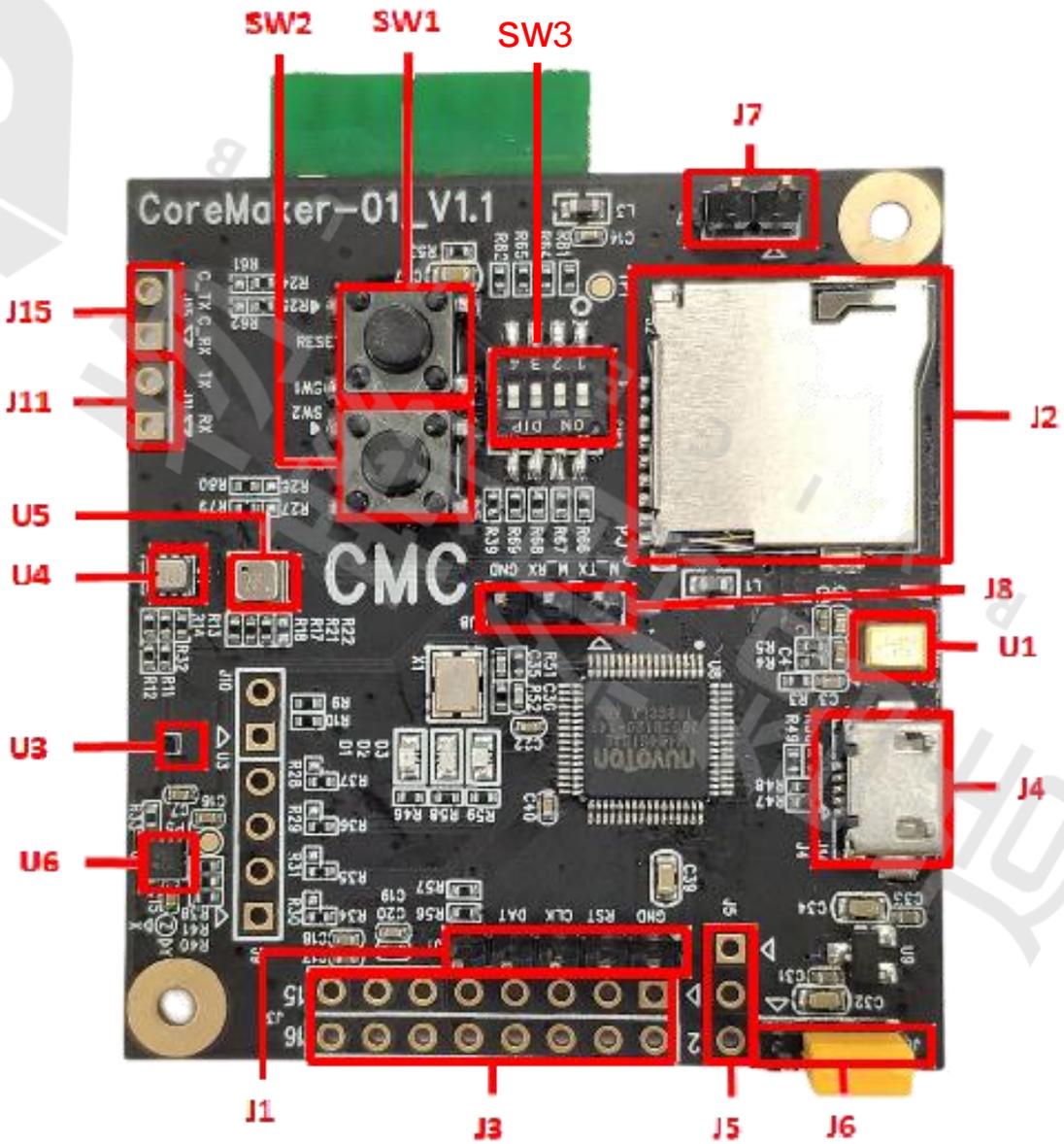


代號

J4

J8

J5



感測器元件

- 麥克風(SPU0410LR5H-QB)
- 磁力計(GMC306A)
- 壓力計(GMP102)
- 環境感測器(BME680)
- 加速度計(KX122-103)

代號

U1

U3

U4

U5

U6

按鈕

- 重置按鈕
- 用戶按鈕
- 感測器撥扭

代號

SW1

SW2

SW3

2-3. 範例所需材料

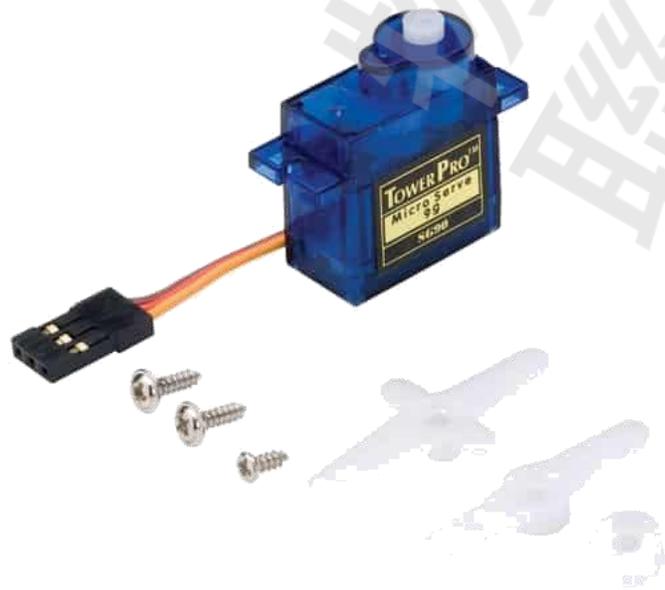


- CoreMaker-01 ×1
- Windows 10 電腦 ×1
- Arduino Uno ×1
- SG90 360度連續旋轉(伺服馬達) ×1
- Micro-USB充電傳輸線(給CoreMaker-01) ×1
- USB Type-B充電傳輸線(給Arduino Uno) ×1
- USB轉TTL序列傳輸線 ×1
- 單芯線 ×少許
- 9V電池(給Arduino Uno) ×1
- 1.5V電池(給CoreMaker-01) ×?

2-4-3. SG90 360度連續旋轉舵機介紹

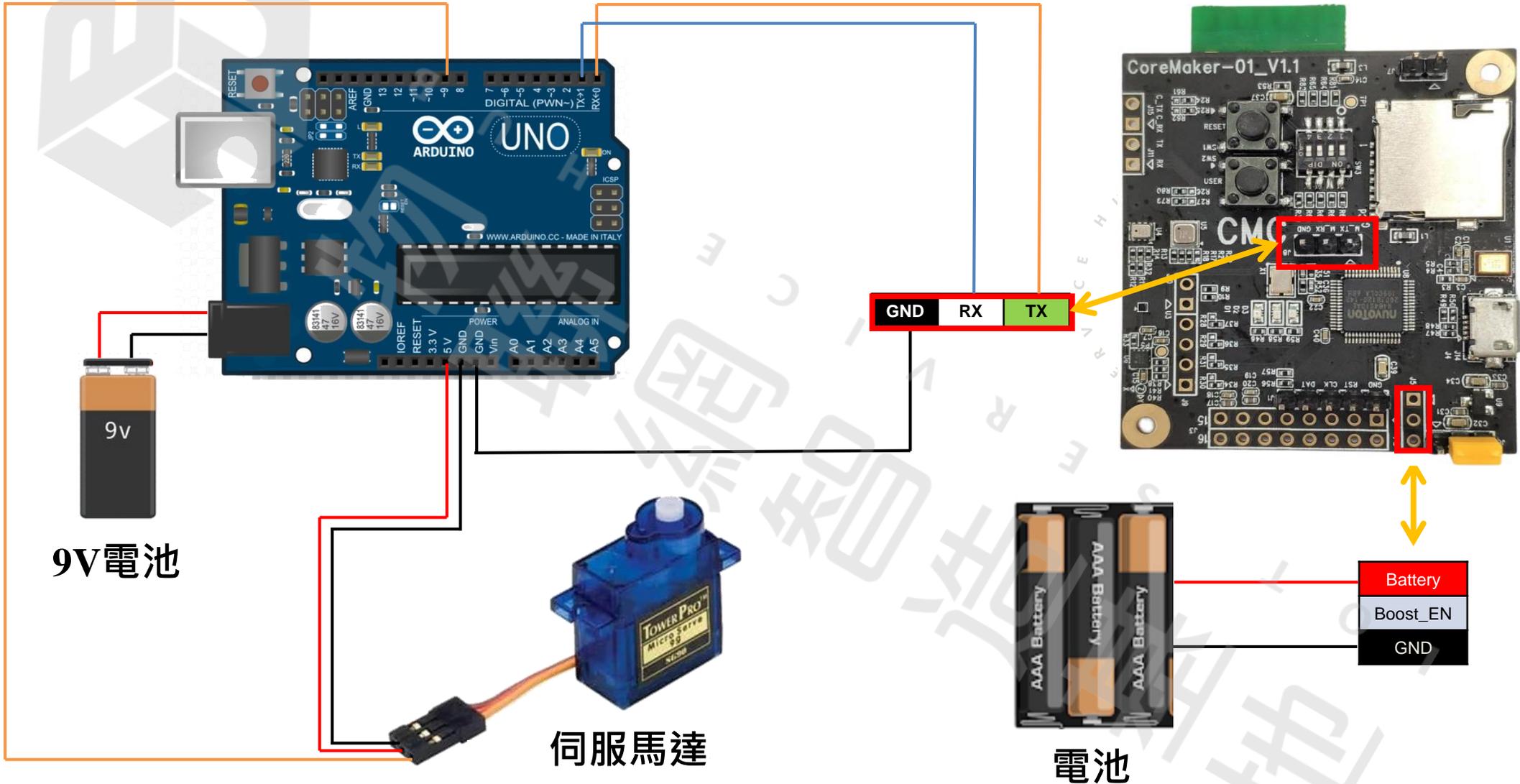
技術參數

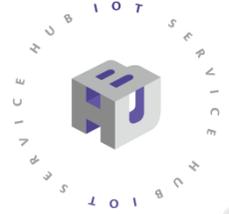
- 尺寸：23mm*12.5mm*29.5mm
- 重量：9克
- 轉動角度：360度連續旋轉（由PWM控制轉速及正反轉）
- 工作扭矩：1.8kg/cm(6V)；1.4 kg/cm（4.8V）
- 工作電壓：3.0V~7.2V（建議5V）
- 連接線長：25cm
- 接線方式:紅線→5V,棕線→地線,橙線→訊號線



2-5. 電路接線圖

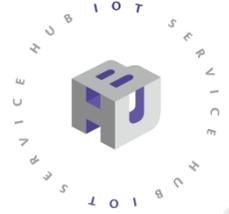
CoreMaker-01



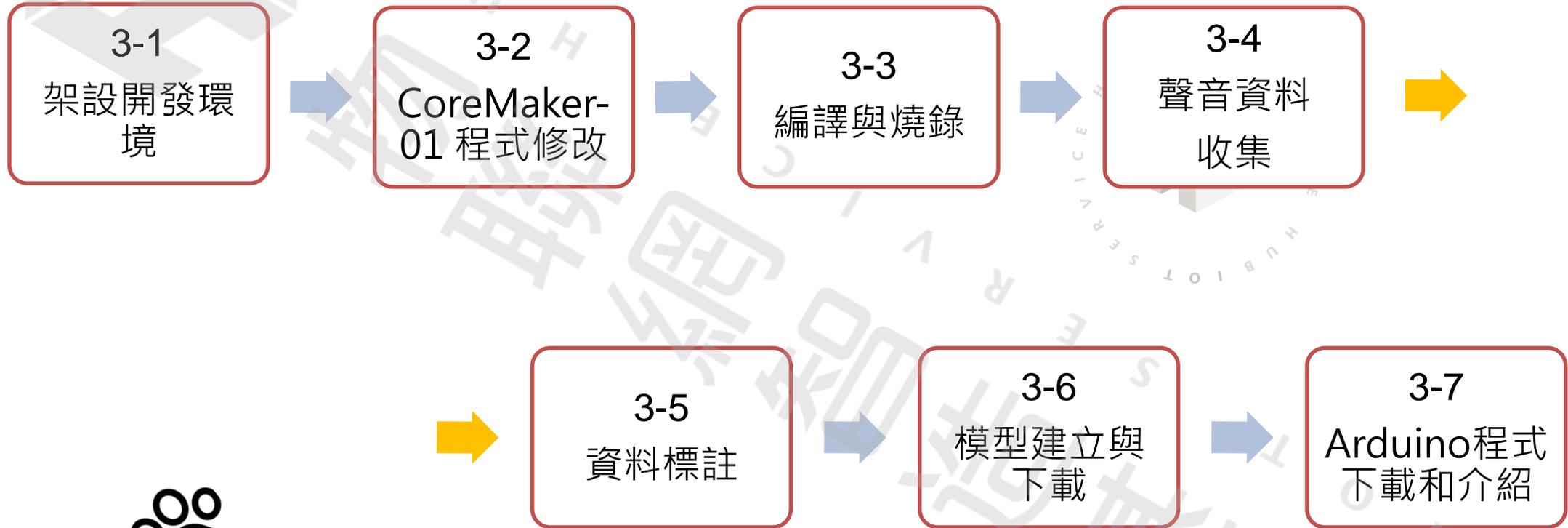


2-6. CoreMaker-01 腳位與元件說明

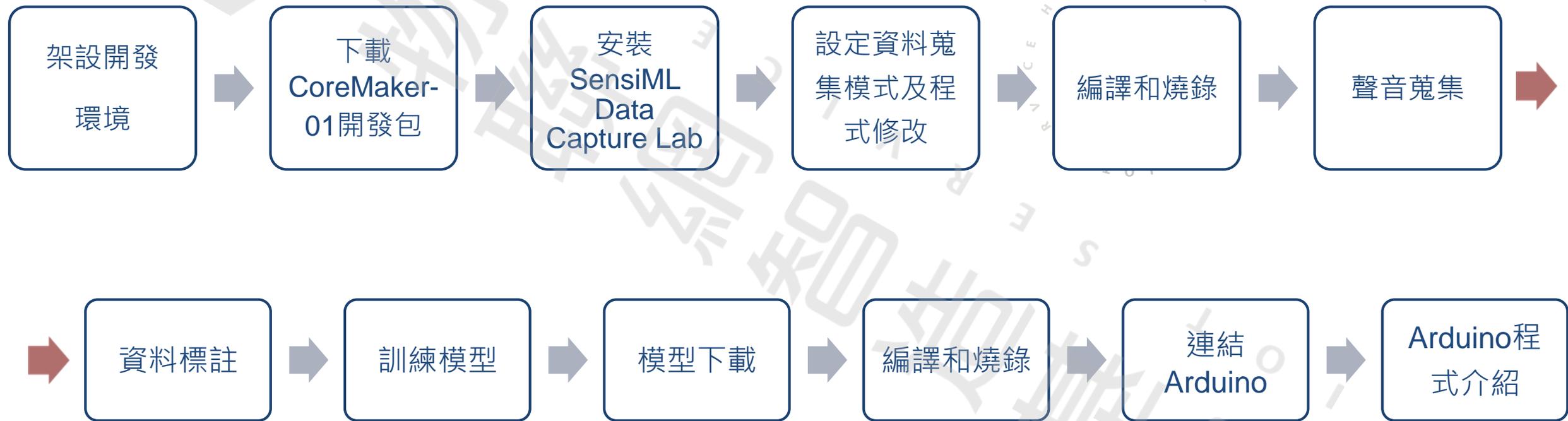
- 板子中間J8區域是UART腳位，其中板上RX和TX分別連接 Arduino Uno上的TX→1以及RX←0，將以UART模式傳送模型辨別結果。GND則對接
- 外接電池接在J5區域，正負極對接。
- 馬達使用SG90伺服馬達，將其連接在Arduino Uno上，當 Uno收到辨別結果進行處理後，再將如何轉動的訊息傳至 SG90伺服馬達。

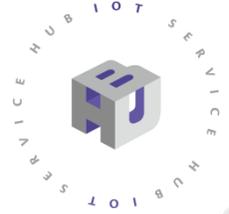


第三章 開發流程解說



開發流程詳細步驟





3-1 架設開發環境

架設CoreMaker-01開發環境相關軟體

開發工具安裝at Windows 10 Environment

- Python 3.6 以上版本
- Git 64-bit Git for Windows
- Cmake 3.19.0 以上版本
- Ninja 1.0 以上版本
- Mbed CLI 2
- GNU Arm Embedded Toolchain 安裝10.3-2021.10 版本

詳細安裝步驟可參考：

https://github.com/CoretronicMEMS/CoreMaker-01/blob/master/docs/coremaker操作指南_V1.1.pdf

以上套件安裝完成後，需重開機

3-1 架設開發環境

下載CoreMaker-01開發包

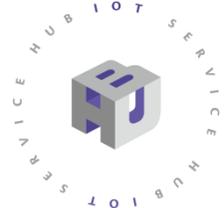
打開cmd，變更自定義路徑後，輸入：

git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git

```
命令提示字元
Microsoft Windows [版本 10.0.19044.1826]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\88692>git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git
Cloning into 'CoreMaker-01'
remote: Enumerating objects: 598, done.
remote: Counting objects: 100% (96/96), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 598 (delta 81), reused 75 (delta 70), pack-reused 502
Receiving objects: 100% (598/598), 18.29 MiB | 2.15 MiB/s, done.

Resolving deltas: 100% (368/368), done.
Submodule 'mbed-os' (https://github.com/CoretronicMEMS/mbed-os.git) registered for path 'mbed-os'
Cloning into 'C:/Users/88692/CoreMaker-01/mbed-os'...
remote: Enumerating objects: 420174, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 420174 (delta 0), reused 0 (delta 0), pack-reused 420172
Receiving objects: 100% (420174/420174), 465.25 MiB | 1.10 MiB/s, done.
Resolving deltas: 100% (293237/293237), done.
Submodule path 'mbed-os': checked out '875ce5bfca68fd74d22ffec266e542d0ebadc0b6'

C:\Users\88692>
```



3-1 架設開發環境

安裝SensiML Data Capture Lab

註冊網址:

<https://sensiml.com/plans/community-edition/>



The screenshot shows the SensiML website with a navigation bar at the top containing 'Products', 'Solutions', 'Plans', 'Resources', 'Support', 'Company', 'Developers', 'Sign In', and a 'GET PROTO EDITION' button. The main content area is titled 'The Community Edition Advantage:' and describes the benefits of the free account. On the right side, there is a 'Create Your Free Account' form with fields for Email, Company/Team Name, First Name, Last Name, and Phone. Below these are dropdowns for 'Intended Application*' and 'Password*' (with a 'Confirm Password*' field). A checkbox for 'I agree to the Terms and Conditions' and a green 'Create My Account' button are at the bottom of the form. On the left side of the screenshot, there is a section titled 'Don't Have A SensiML Account Yet?' with a 'Get For Free' button. Below this, there are two product cards: 'SensiML Data Capture Lab' (Windows 10, Version: 2022.4.0, Release Notes) and 'SensiML Analytics Studio' (Web Application, No download required). A red box highlights the 'SensiML Data Capture Lab' card and the 'Create Your Free Account' form. A blue arrow points from the registration URL to the top right of the screenshot.



Capture Lab 下載網址:

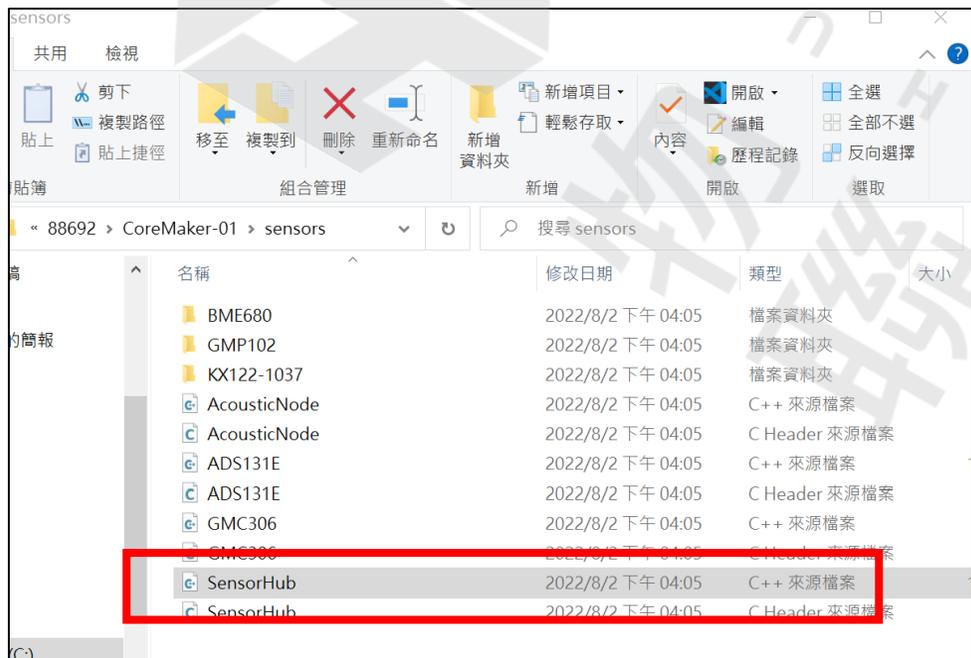
<https://sensiml.com/download/>

3-2 CoreMaker-01 程式修改

設定資料蒐集模式

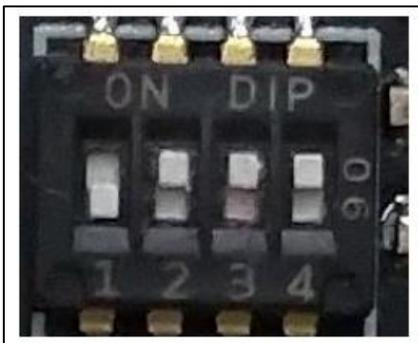
檔案位置:

CoreMaker-01/sensors/SensorHub.cpp



```

G+ SensorHub.cpp
C: > Users > 88692 > CoreMaker-01 > sensors > G+ SensorHub.cpp
47 extern USBSerial serial;
48
49
50 typedef int (*RUN_MODEL)(SENSOR_DATA_T *, int , int );
51 namespace CMC
52 {
53     SPI spi0(PA_0, PA_1, PA_2, PA_3, mbed::use_gpio_ssel);
54     I2C i2c1(I2C_SDA, I2C_SCL);
55
56     AcousticNode acoustic_node(PB_6, 16000);
57     BME680 bme680(0x76 << 1, &i2c1);
58     GMP102 gmp102(&i2c1, 256);
59     GMC306 gmc306(&i2c1, 10);
60     KX122 kx122(&spi0, PA_10, 3200);
61
62
    
```



← 把SW3 撥鈕調整至聲音蒐集模式

↑ 在SensorHub.cpp的56行
將聲音取樣類別調整至16000(原定2000)
取得較高解析度

3-2 CoreMaker-01 程式修改

程式修改

```
SensorHub.cpp
C: > Users > 88692 > CoreMaker-01 > sensors > SensorHub.cpp
250
251     if(m_DCLStatus == DCL_CONNECTED)
252     {
253         serial.send((uint8_t*)m_dataBuffer, m_dataLen);
254     }
255     else if(run_ai_model[i])
256     {
257         int ret = run_ai_model[i]((SENSOR_DATA_T*)m_dataBuffer, m_dataLen/sizeof(SENSOR_DATA_T));
258         if (ret > -1)
259         {
260             printf("AI classification result: %d\n", ret);
261             kb_reset_model(0); // Reset running model to initial state.
262         }
263     }
264     else if (ret == -2)
```

← 更改前

接續在SensorHub.cpp的260行，將模型回傳的資訊改成只需傳送模型類別的數字即可，方便之後Arduino的接收和運算。

```
SensorHub.cpp
C: > Users > 88692 > CoreMaker-01 > sensors > SensorHub.cpp
250
251     if(m_DCLStatus == DCL_CONNECTED)
252     {
253         serial.send((uint8_t*)m_dataBuffer, m_dataLen);
254     }
255     else if(run_ai_model[i])
256     {
257         int ret = run_ai_model[i]((SENSOR_DATA_T*)m_dataBuffer, m_dataLen/sizeof(SENSOR_DATA_T));
258         if (ret > -1)
259         {
260             printf("%d\n", ret);
261             kb_reset_model(0); // Reset running model to initial state.
262         }
263     }
264     else if (ret == -2)
265     {
266         printf("This segment has been filtered.\n");
267     }
268     else if (ret < -2)
269     {
270         printf("AI error: %d\n", ret);
271     }
272 }
```

← 更改後

3-3 編譯與燒錄

編譯篇

將更改的程式存檔後，打開cmd，先輸入：

`cd CoreMaker-01`

接著輸入：

`mbed-tools compile -m AIOT2101 -t GCC_ARM`

進行編譯。

```

命令提示字元
Microsoft Windows [版本 10.0.19044.1826]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\88692>cd CoreMaker-01
C:\Users\88692\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
  
```

編譯一段時間完成後，
bin檔的路徑於右圖所示

```

[2/21] Linking CXX executable AIOT_2101.elf
-- built: C:/Users/88692/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/88692/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex
C:\Users\88692\CoreMaker-01>
  
```

| | | |
|----------------------|--------------------|---------------------------|
| CMakeFiles | 2022/7/27 下午 02:51 | 檔案資料夾 |
| mbed-os | 2022/7/27 下午 02:51 | 檔案資料夾 |
| TARGET_AIOT2101 | 2022/7/27 下午 02:51 | 檔案資料夾 |
| .mbedignore | 2022/7/27 下午 02:51 | MBEDIGNORE 檔... |
| .ninja_deps | 2022/7/27 下午 02:51 | NINJA_DEPS 檔案 76 |
| ninja_log | 2022/7/27 下午 02:51 | NINJA_LOG 檔案 11 |
| AIOT_2101.bin | 2022/7/27 下午 02:51 | BIN 檔案 11 |
| AIOT_2101.elf | 2022/7/27 下午 02:51 | ELF 檔案 4,01 |
| AIOT_2101.elf.map | 2022/7/27 下午 02:51 | Linker Address M... 30,84 |
| AIOT_2101.hex | 2022/7/27 下午 02:51 | HEX 檔案 30 |
| build.ninja | 2022/7/27 下午 02:51 | NINJA 檔案 33,90 |

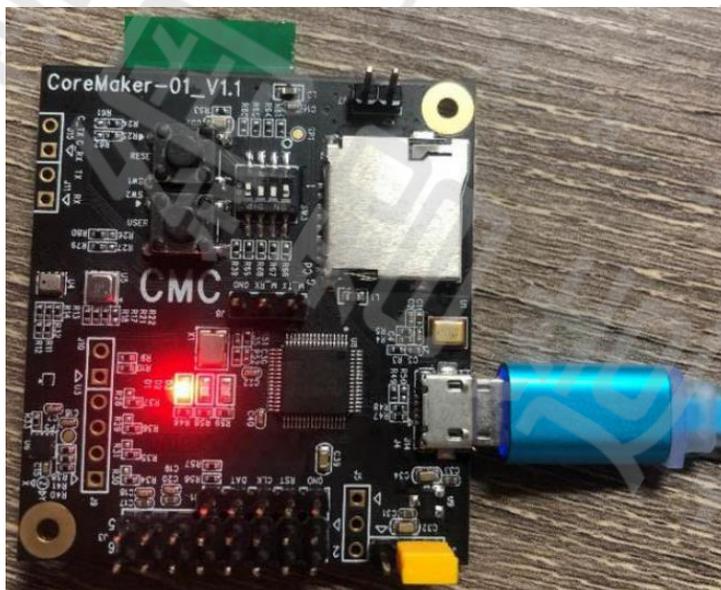
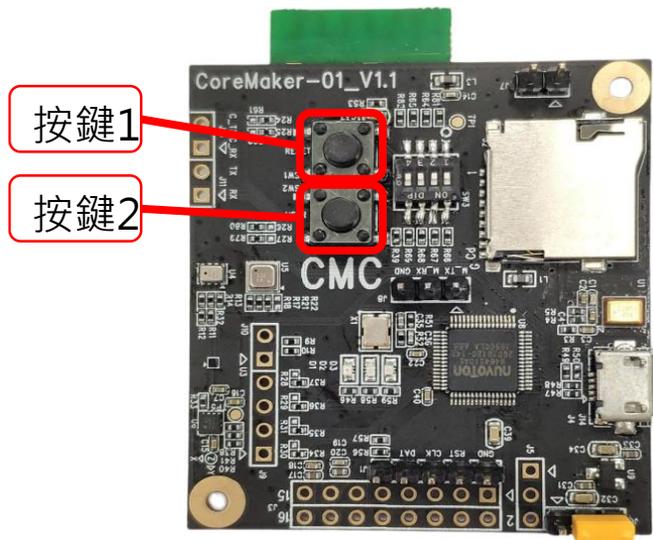
bin檔位置：

`CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM\AIOT_2101.bin`

3-3 編譯與燒錄

燒錄篇

韌體編譯完成後，要將檔案燒錄進CoreMaker-01，以Micro-USB充電傳輸線連接電腦以及CoreMaker-01要進入燒入模式，先同時按下按鍵1和按鍵2，再來放開按鍵1，後放按鍵2，當CoreMaker-01板子上的紅色LED燈熄滅，即可開始燒錄。



一般模式



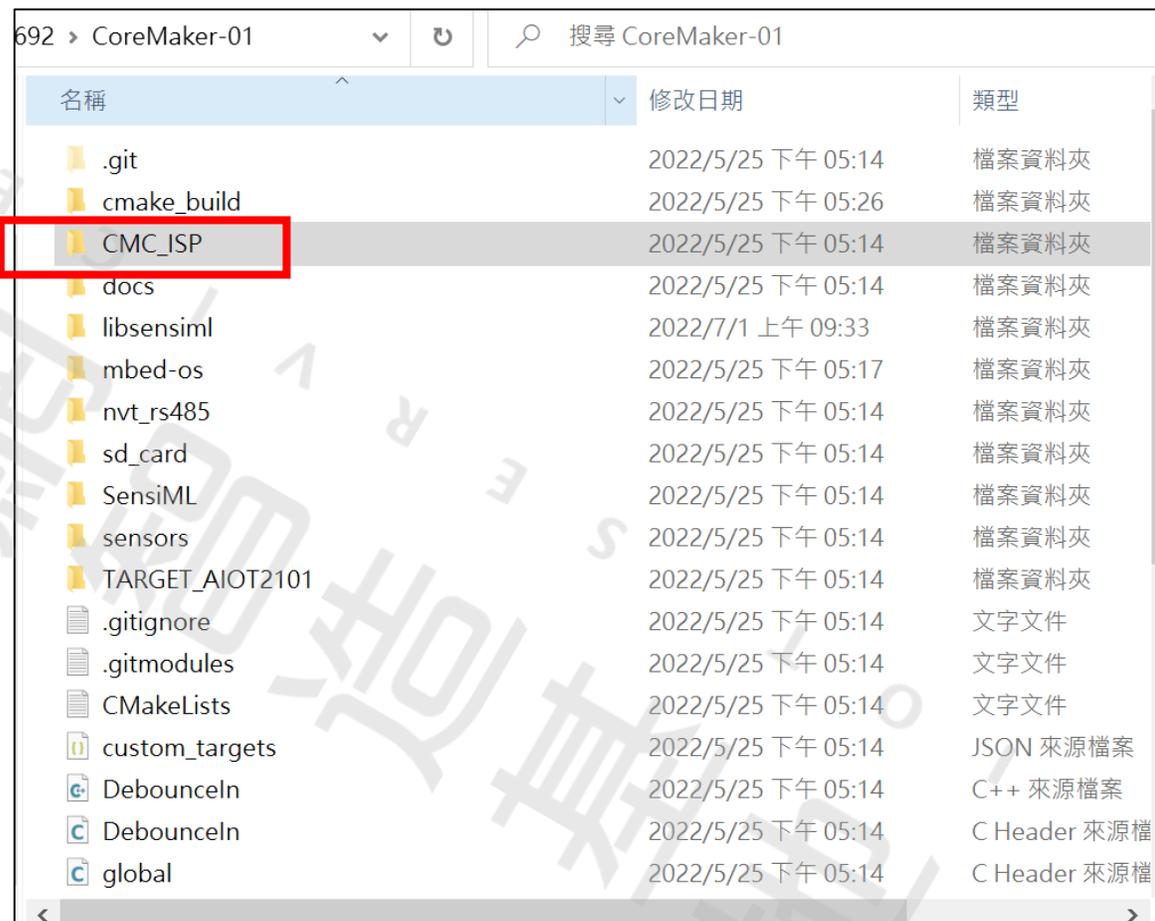
燒錄模式



3-3 編譯與燒錄

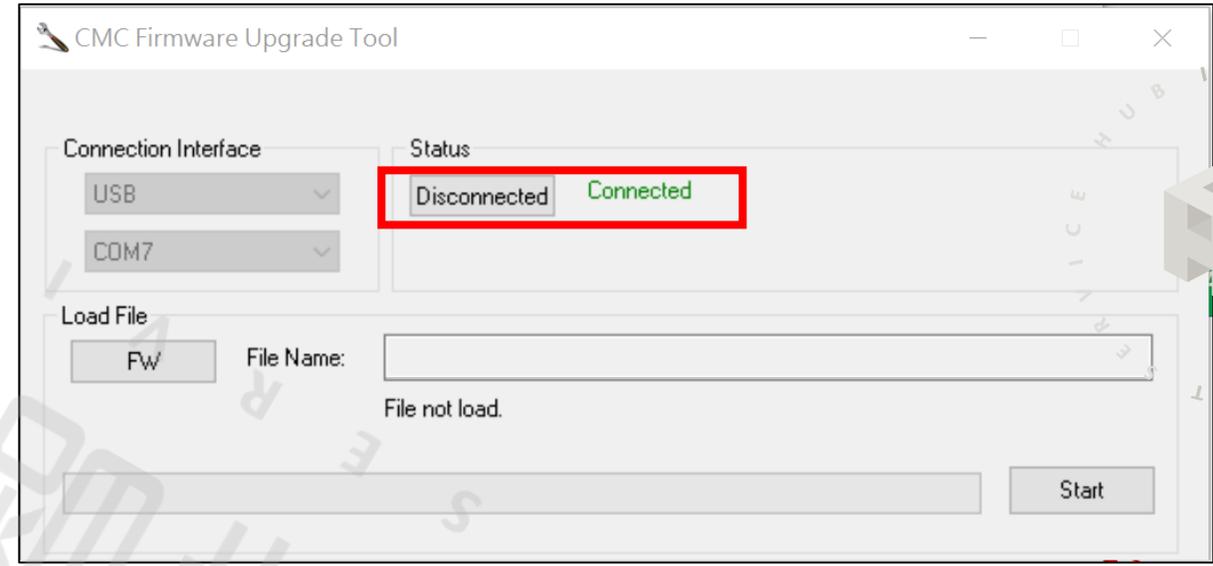
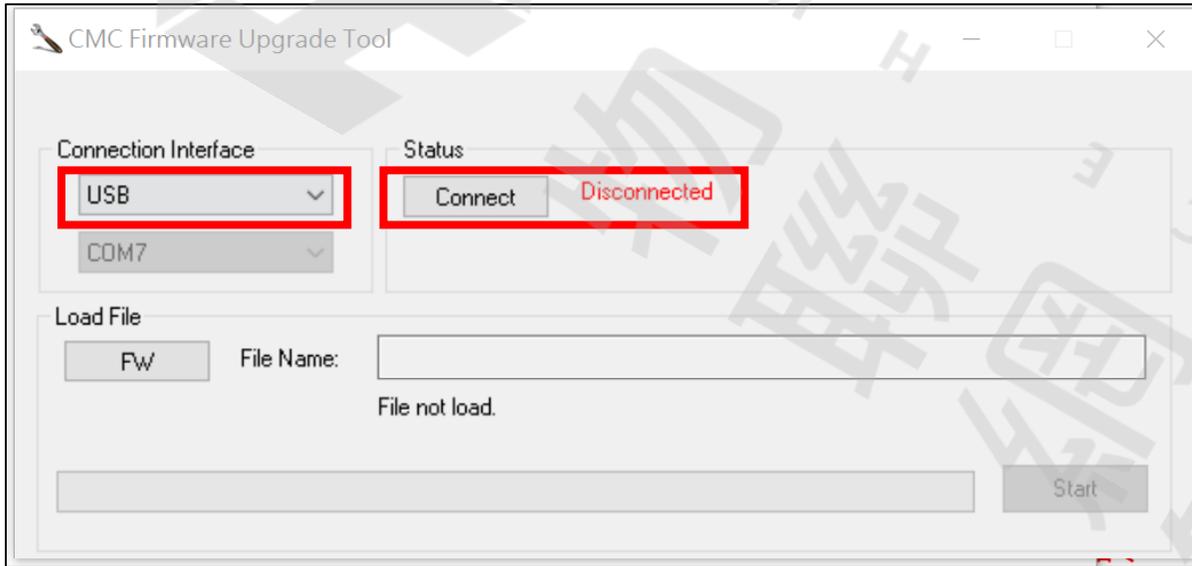
燒錄篇

開啟燒錄程式CMC_ISP.exe，
位置在CoreMaker-01/CMC_ISP



3-3 編譯與燒錄

燒錄篇

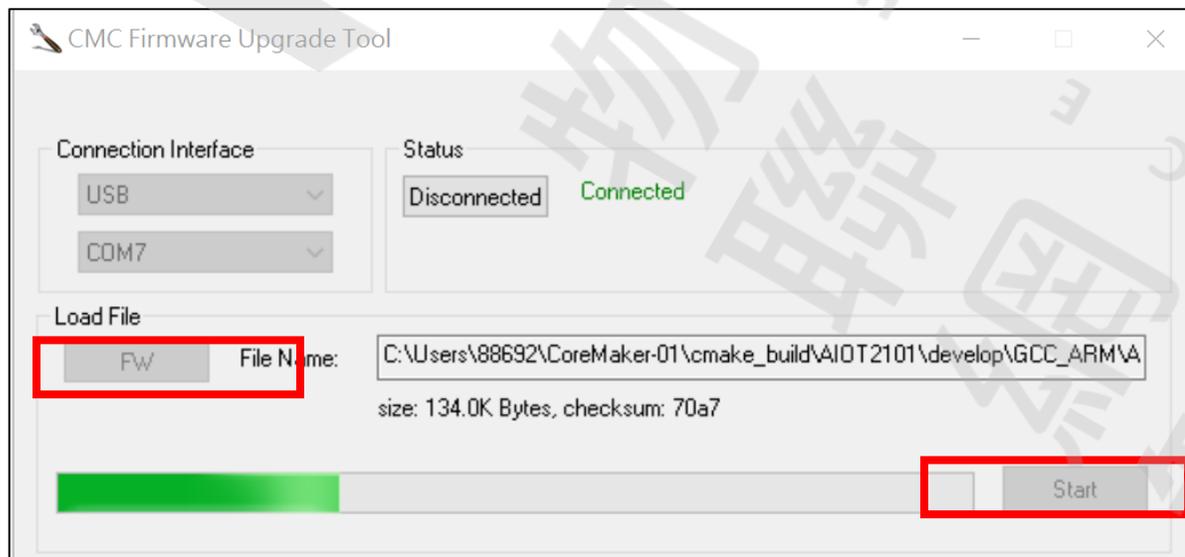


打開可執行程式後，
在Connection Interface選擇 USB，
接著點擊左側Connect 連接。

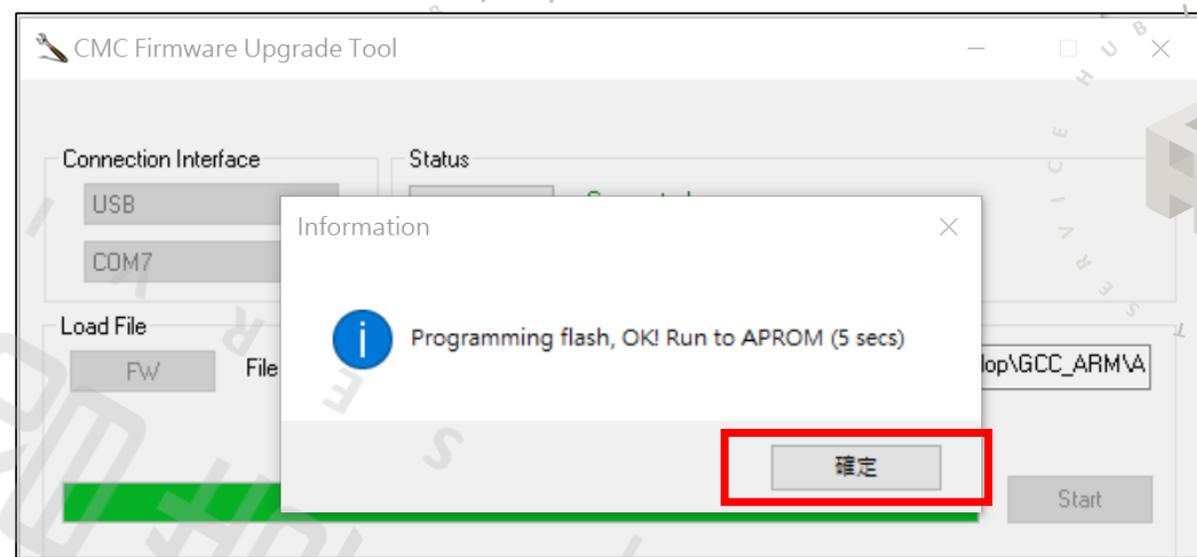
出現綠色狀態，確認連接成功

3-3 編譯與燒錄

燒錄篇



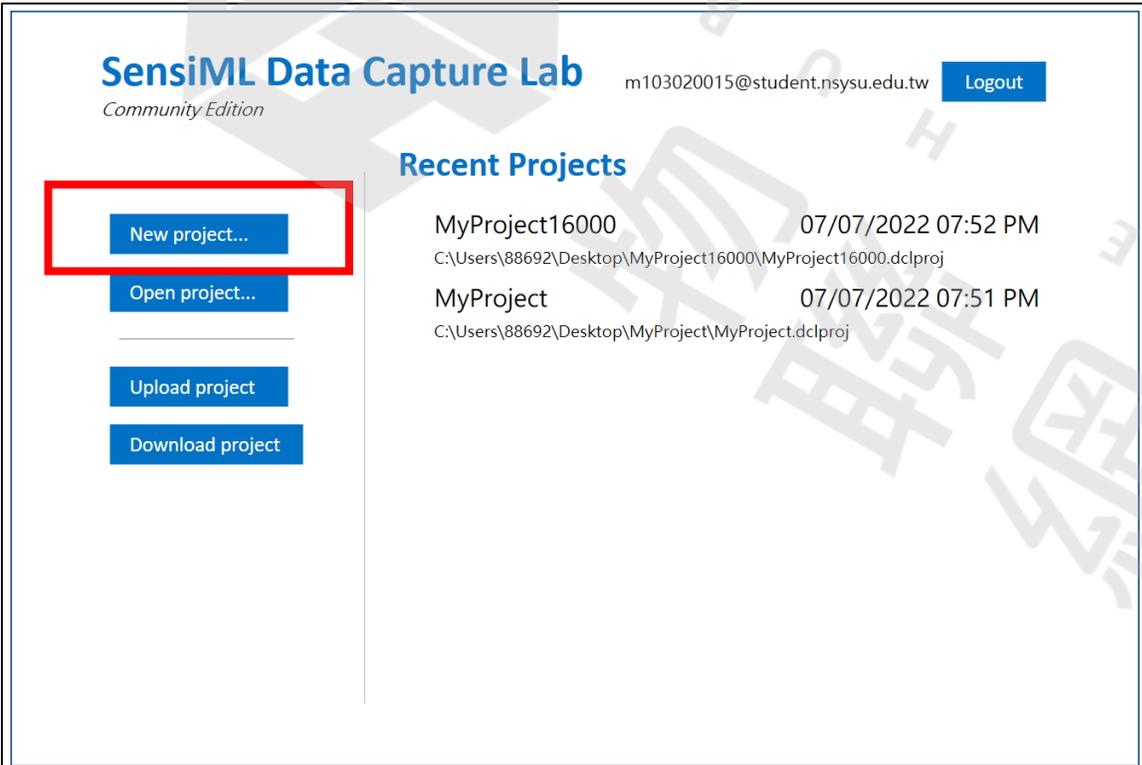
點擊FW後選擇我們bin檔的位置，
按下Start開始燒錄。



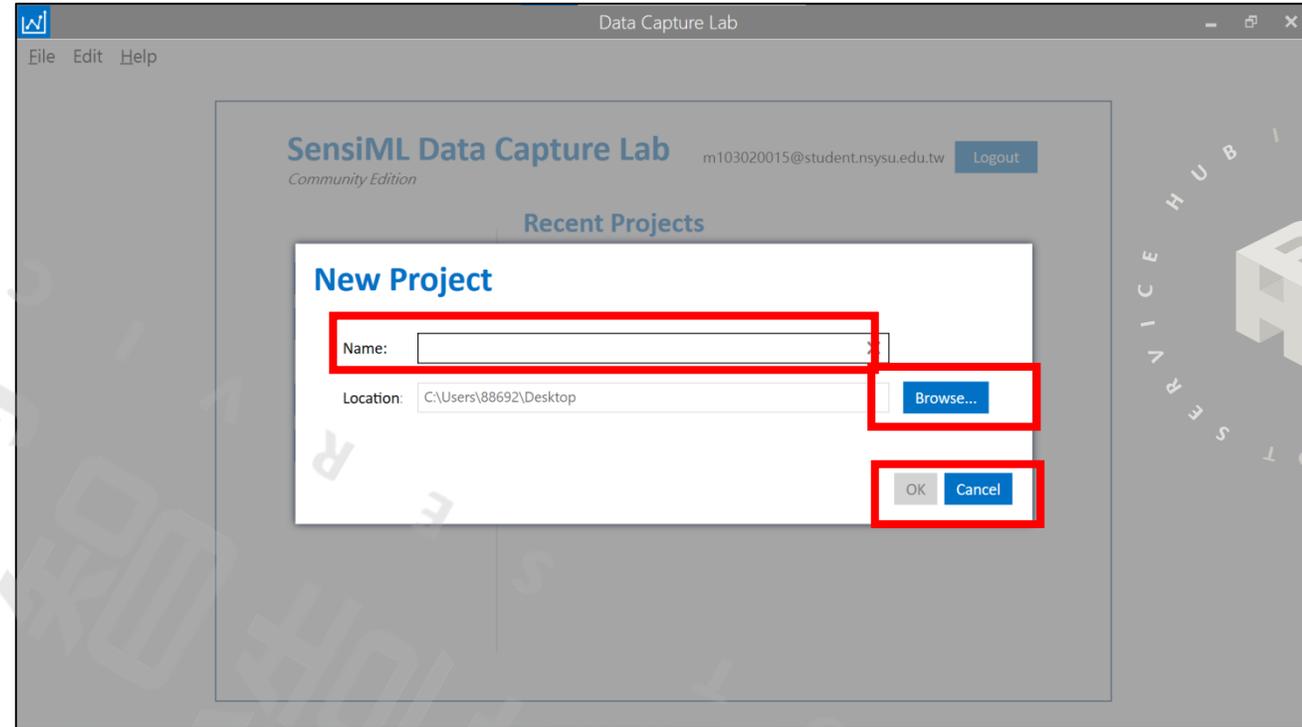
燒錄完成後出現成功畫面

3-4 聲音資料收集

SensiML Data Capture Lab 設置

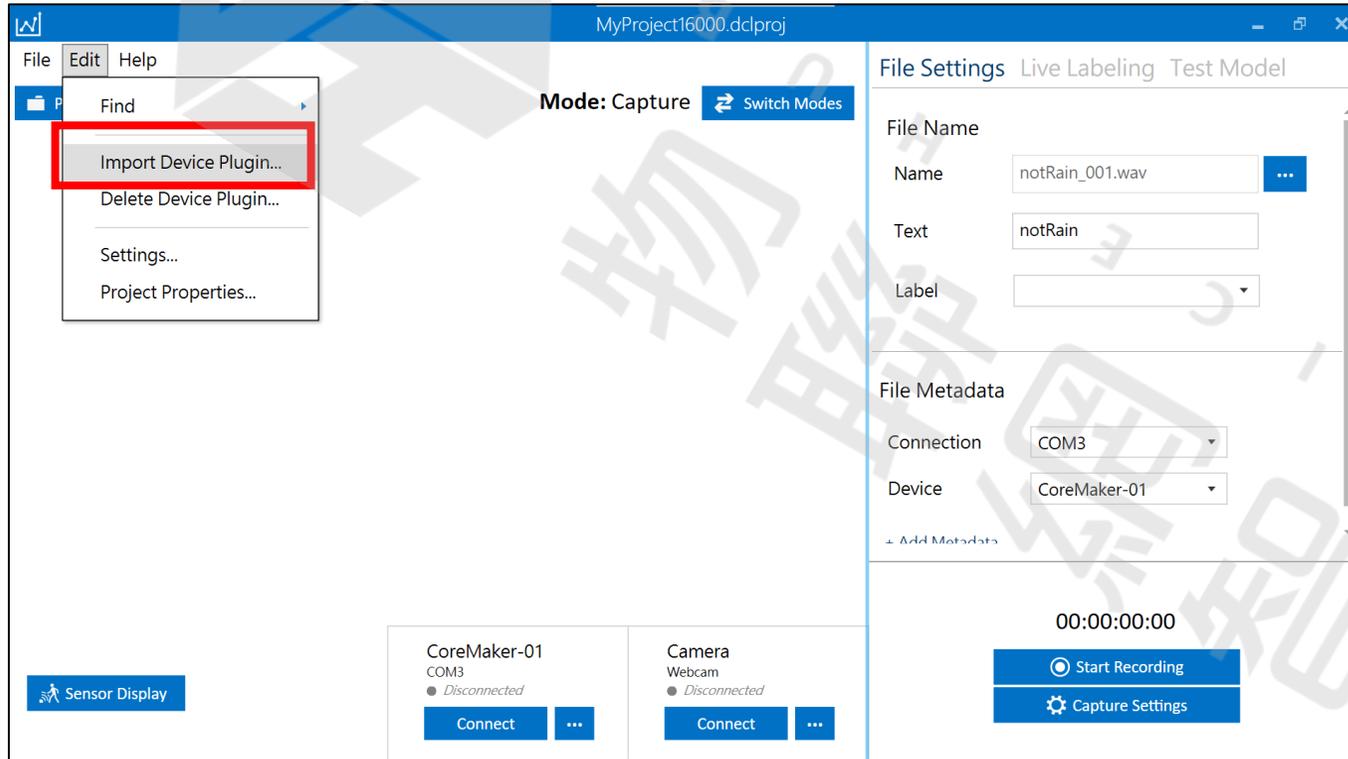


開啟下載好的 SensiML Data Capture Lab
點擊New project

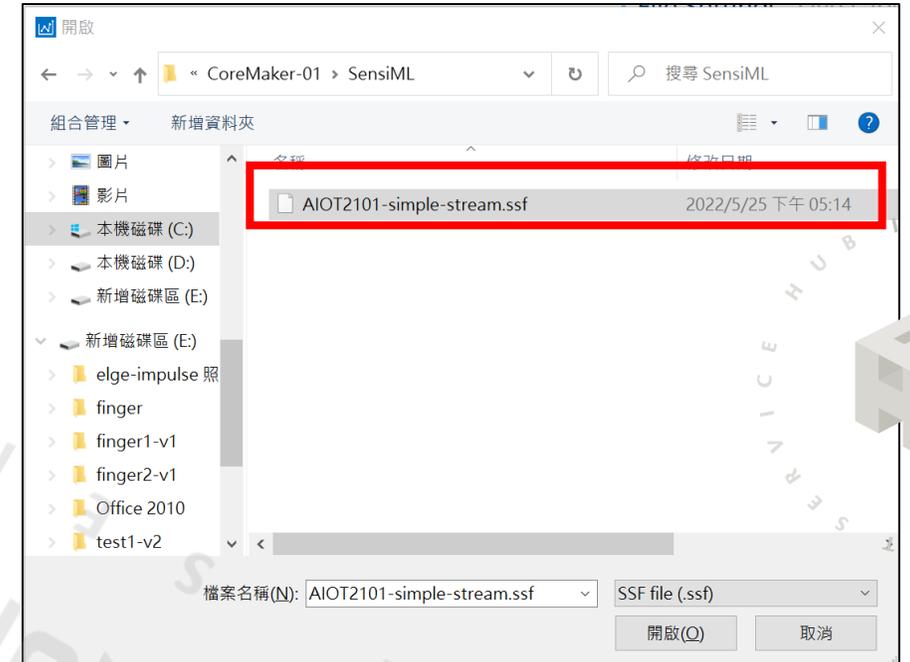


輸入Project 的名稱，選擇檔案位置，點選OK建立

3-4 聲音資料收集

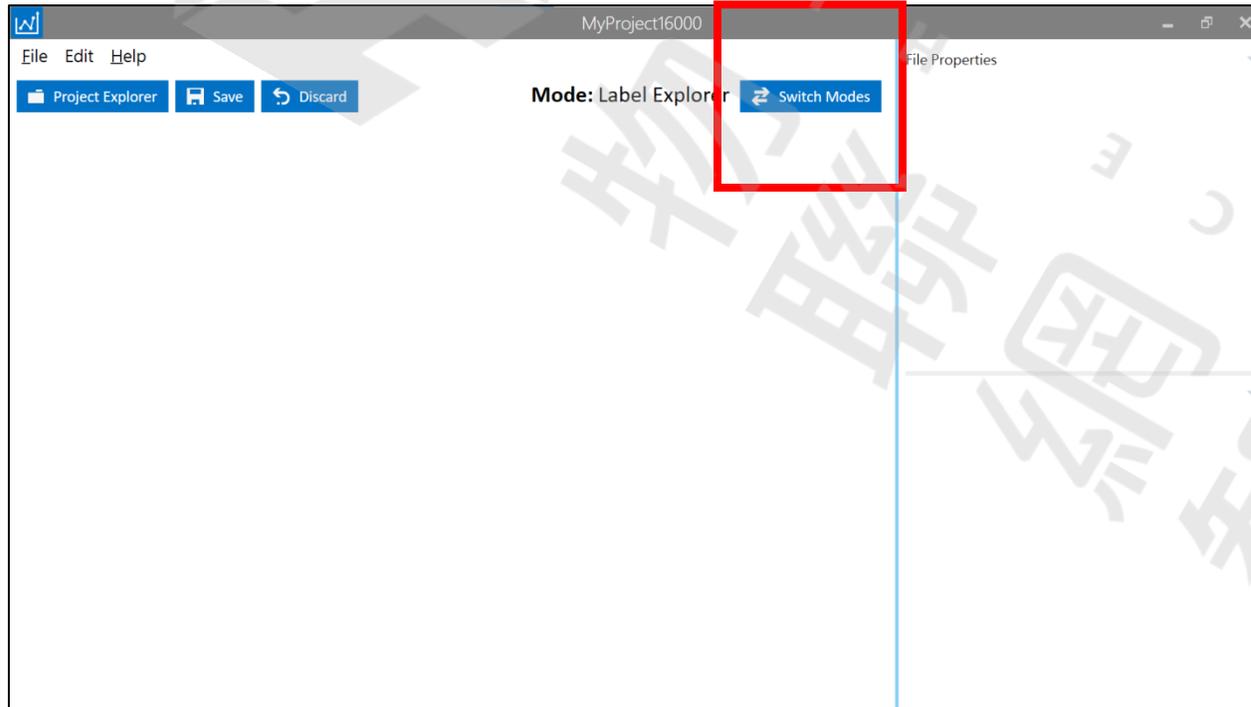


匯入設定檔，點選左上Edit的Import Device Plugins

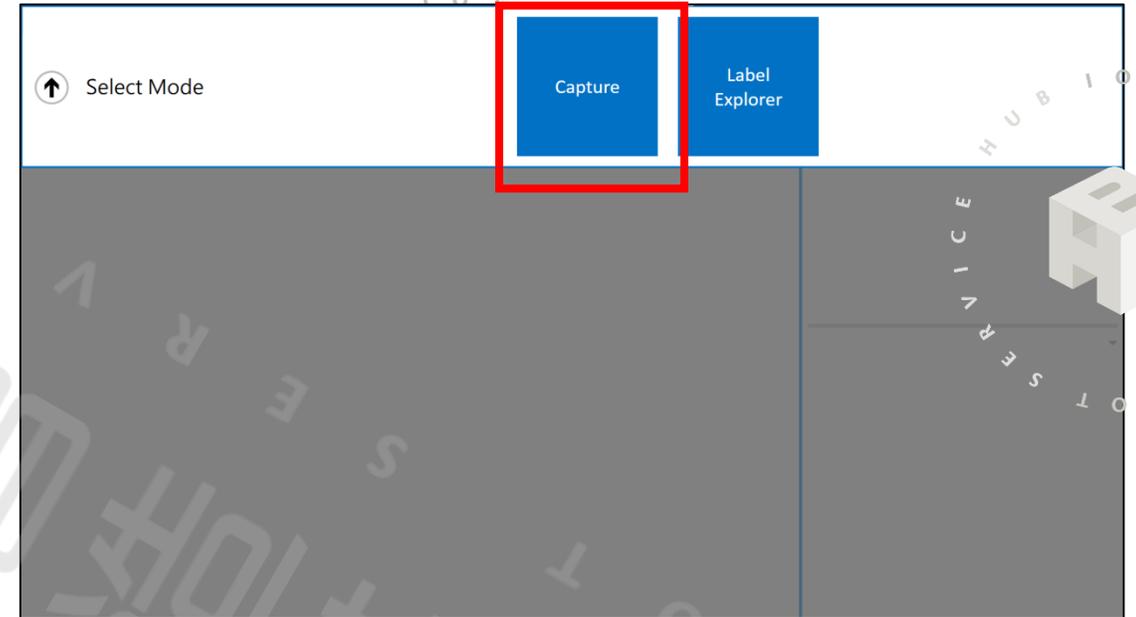


檔案位置在
CoreMaker-01/SensiML/
AIOT2101-simple-stream.ssf
開啟匯入檔案

3-4 聲音資料收集

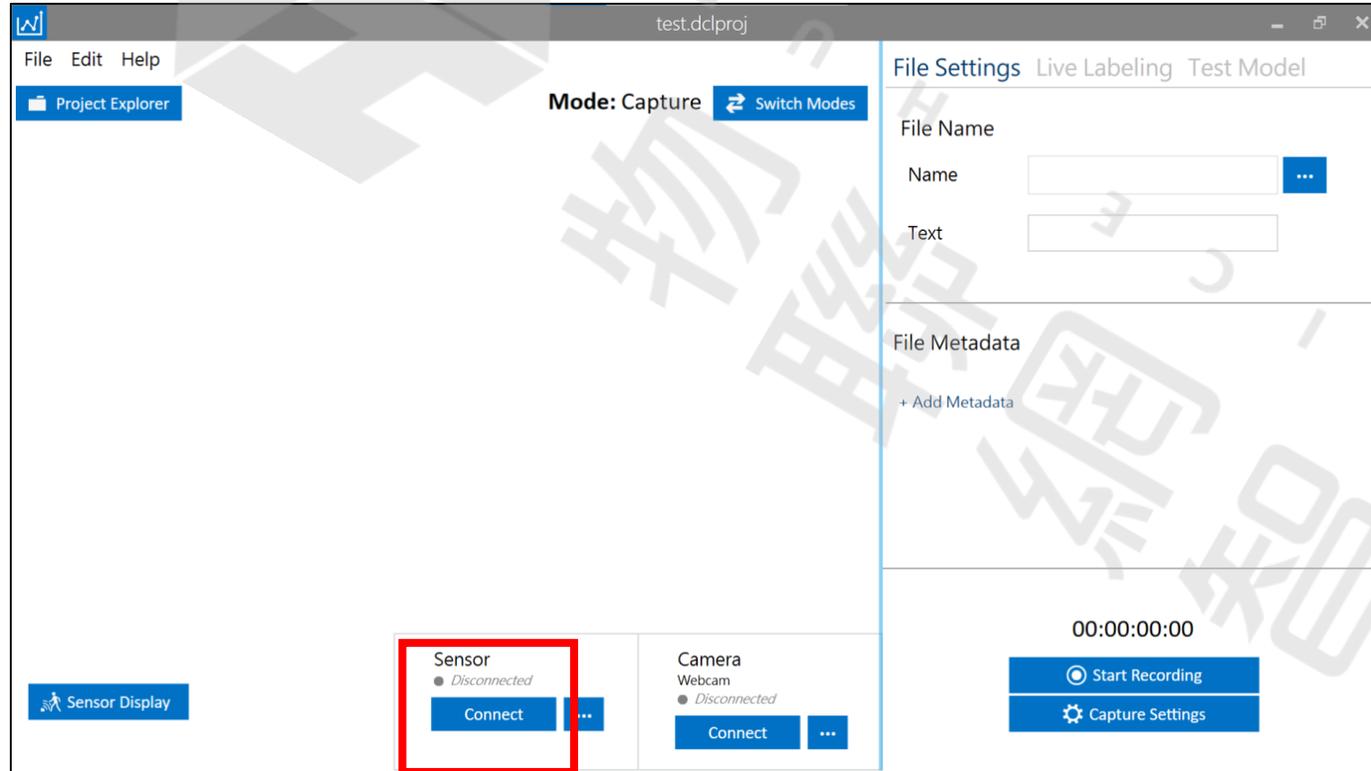


設置完後會回到主畫面 點選上方Switch Modes

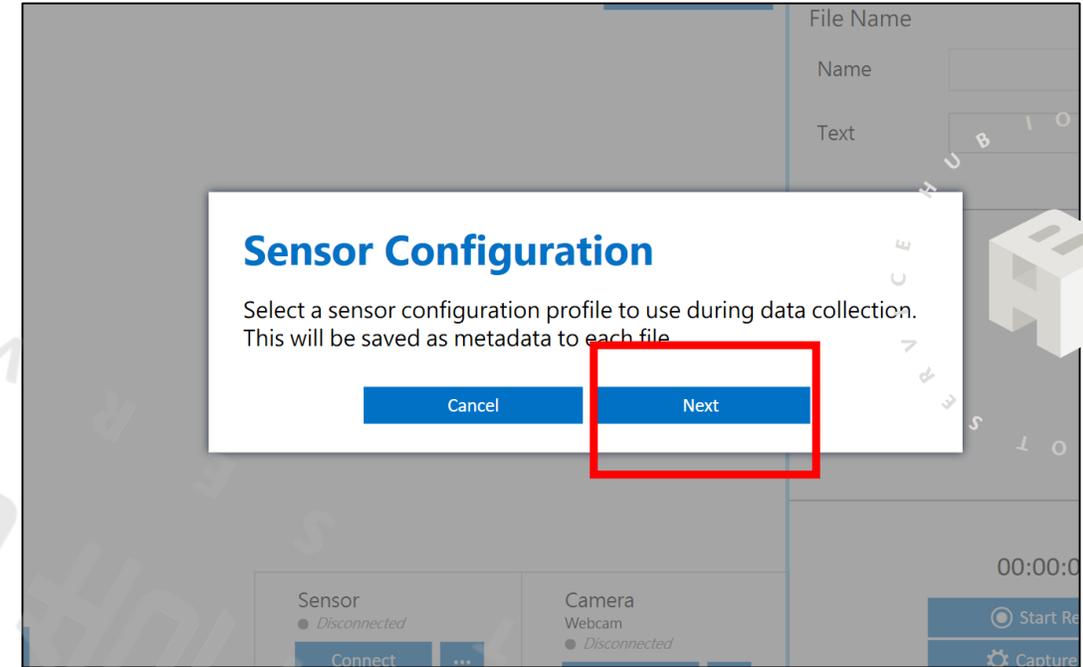


選取Capture

3-4 聲音資料收集



點選後Capture 的畫面，先選擇下方Sensor 的Connect



跳出Configuration，選擇Next

3-4 聲音資料收集

test.dclproj

Select a Device Plugin

The Data Capture Lab comes with plugins for the following devices. You can import your own plugin for any third-party device by following the instructions in the [Creating a Device Plugin documentation](#)

| Source | Manufacturer | Device | Plugin Developer | Capture Protocol | Available Sensors |
|----------------------------------|----------------------|-----------------------------|----------------------|------------------|--|
| <input checked="" type="radio"/> | CoretronicMEMS | CoreMaker-01 | CoretronicMEMS | Simple Streaming | Microphone, Temperature, Pressure, Humidity, Gas resistance, Accelerometer, Magnet |
| <input type="radio"/> | Arduino | Nano33 BLE Sense | SensiML | Simple Streaming | Accelerometer, Gyroscope, Microphone |
| <input type="radio"/> | Arduino | Nicla Sense ME | SensiML | Simple Streaming | Accelerometer, Gyroscope |
| <input type="radio"/> | Infineon | PSoC 6 Wi-Fi BT Pioneer Kit | SensiML | Simple Streaming | Accelerometer, Gyroscope, Microphone |
| <input type="radio"/> | Microchip Technology | SAMD21 ML Eval Kit | Microchip Technology | Simple Streaming | Accelerometer, Gyroscope |
| <input type="radio"/> | Nordic Semiconductor | Thingy:52 | SensiML | Custom | Accelerometer, Gyroscope, Audio |
| <input type="radio"/> | onsemi | RSL10 Sense | onsemi | Simple Streaming | Accelerometer, Gyroscope |
| <input type="radio"/> | QuickLogic | Chilkat | QuickLogic | Custom | Accelerometer, Gyroscope |
| <input type="radio"/> | QuickLogic | QuickAI | QuickLogic | Custom | Accelerometer, Gyroscope, Microphone Channel 1, Channel 2, Channel 3, Channel 4 |

Cancel Next

Plugin Details

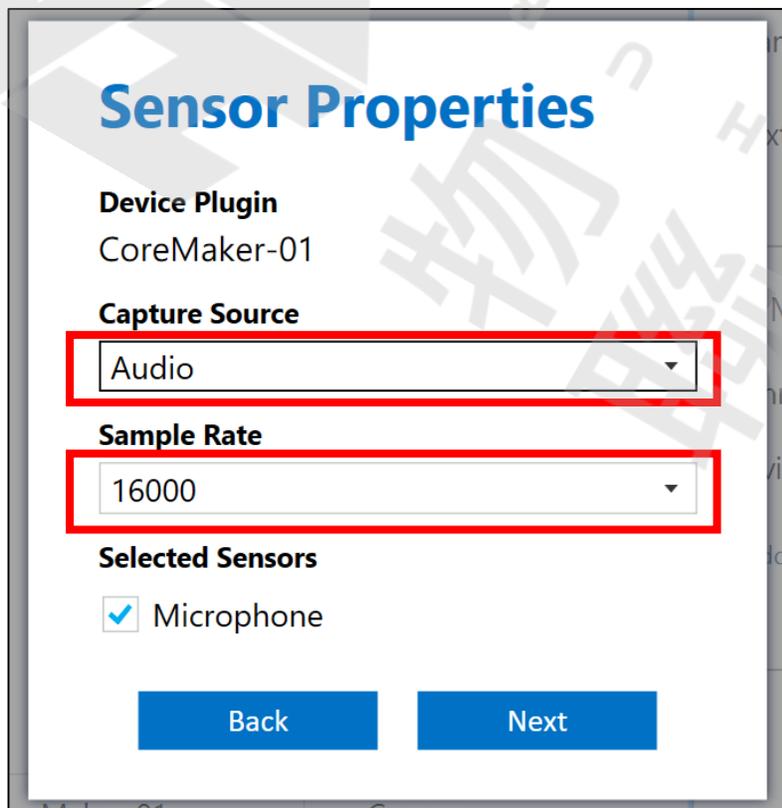
Device: CoreMaker-01
 Manufacturer: CoretronicMEMS
 Plugin Developer: CoretronicMEMS
 Capture Protocol: Simple Streaming
 Connections: Serial Port (Baud Rate: 921600)
 Wi-Fi
 Firmware Download: [Data Collection Firmware](#)
 Sensor Summary:
 Name: Audio
 Sensors: Microphone
 Sample Rates: 16000, 8000, 4000, 2000, 1000, 400, 200, 100
 Name: Environment
 Sensors: Temperature, Pressure, Humidity, Gas resistance
 Sample Rates: 1

Back Next

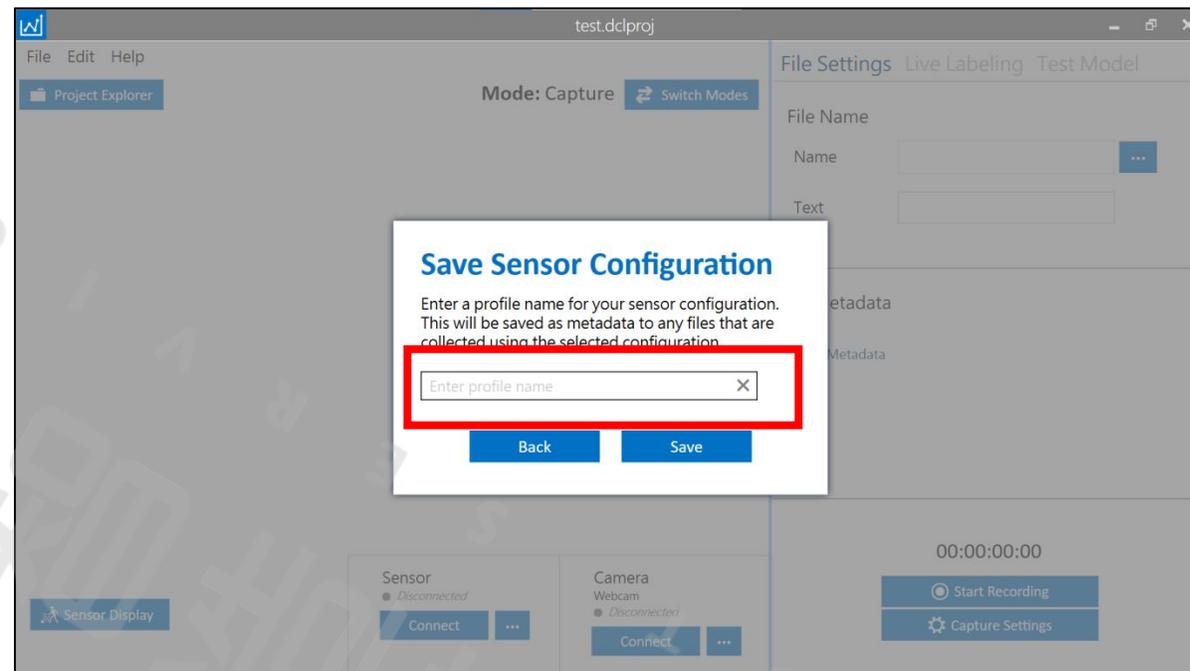
找到我們的開發版，點選CoreMaker-01

跳出設備訊息，按Next

3-4 聲音資料收集

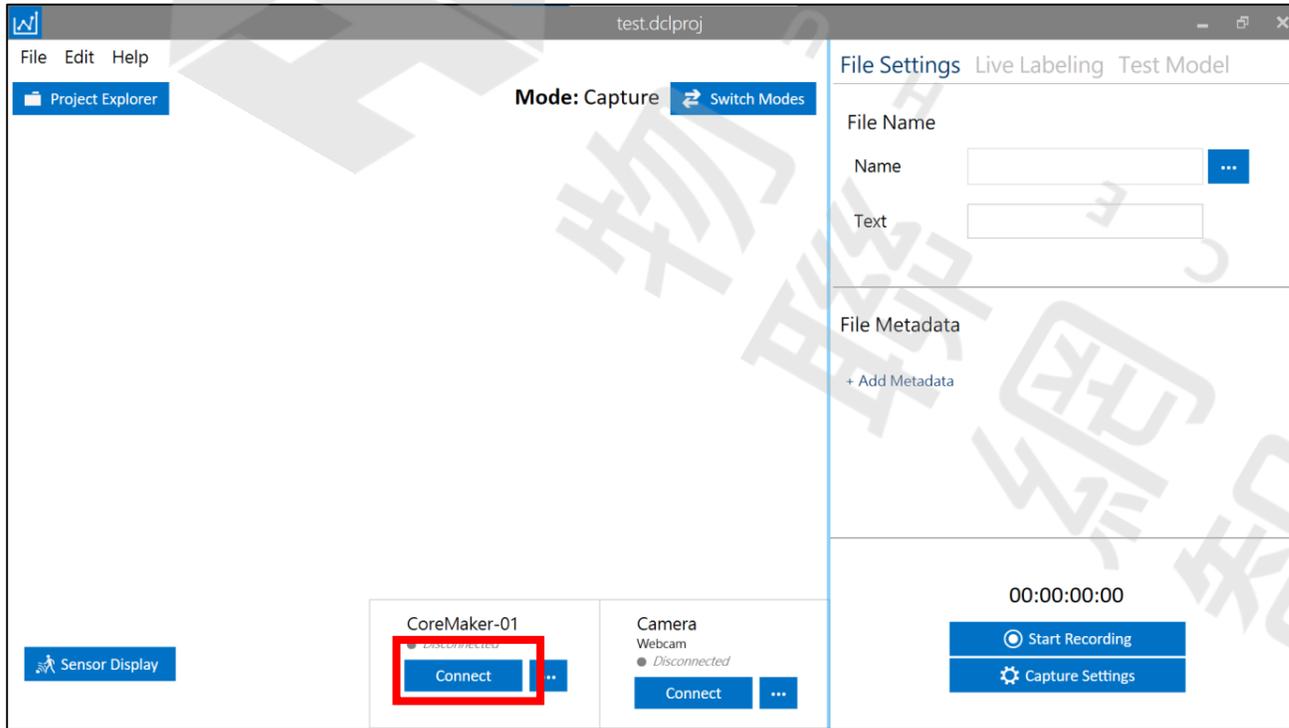


Capture Source 選擇 聲音 Audio
Sample Rate 改成 16000

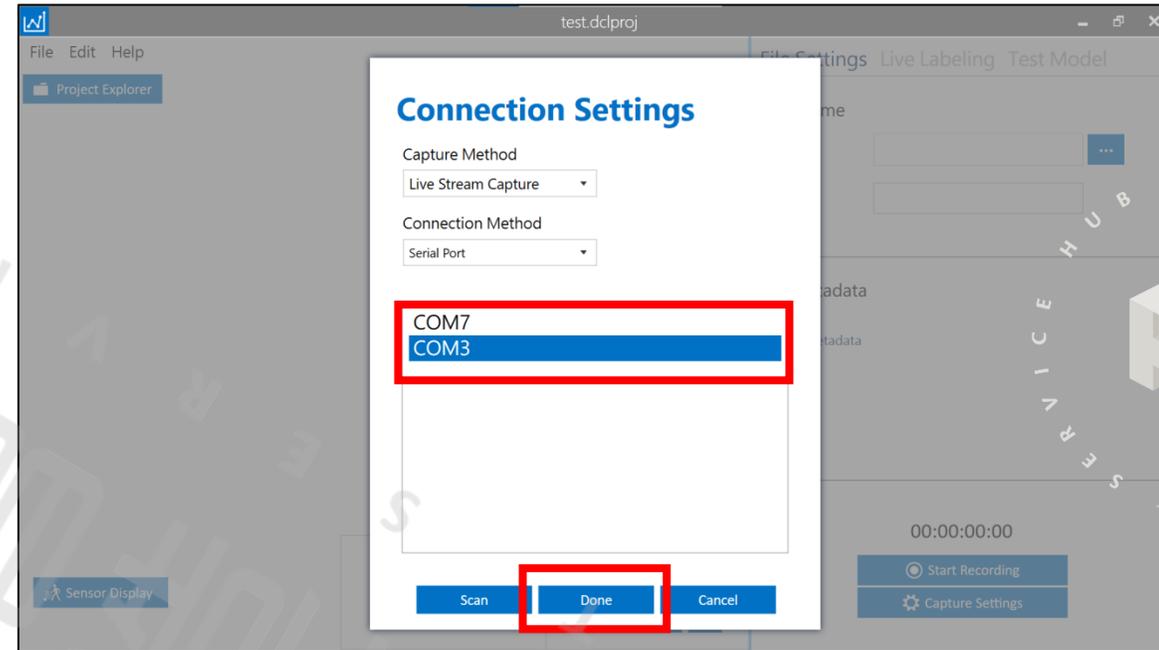


輸入使用者自訂的Sensor名稱

3-4 聲音資料收集

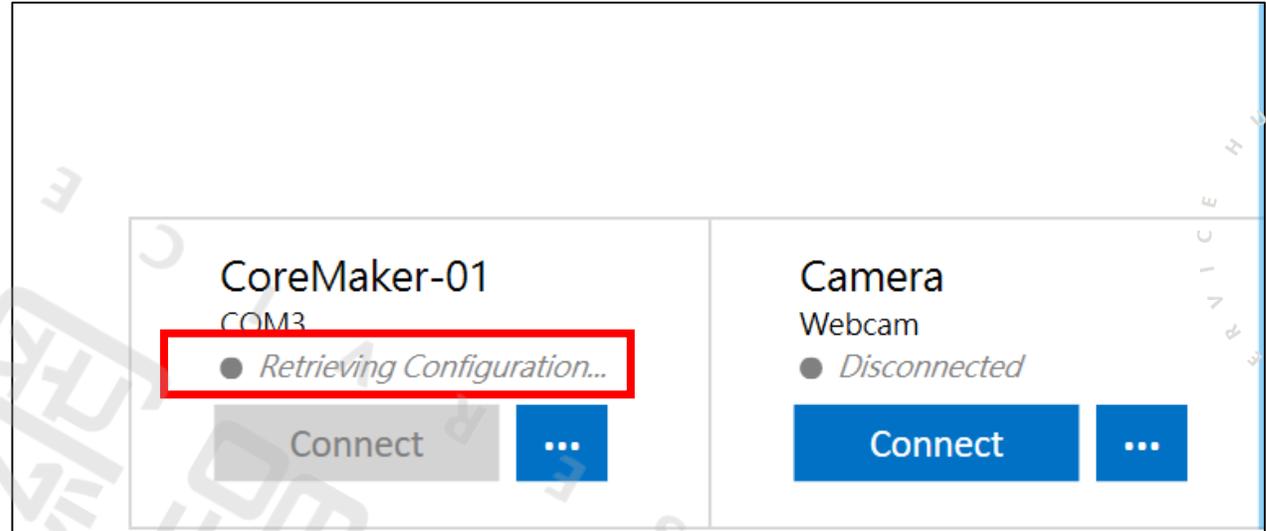
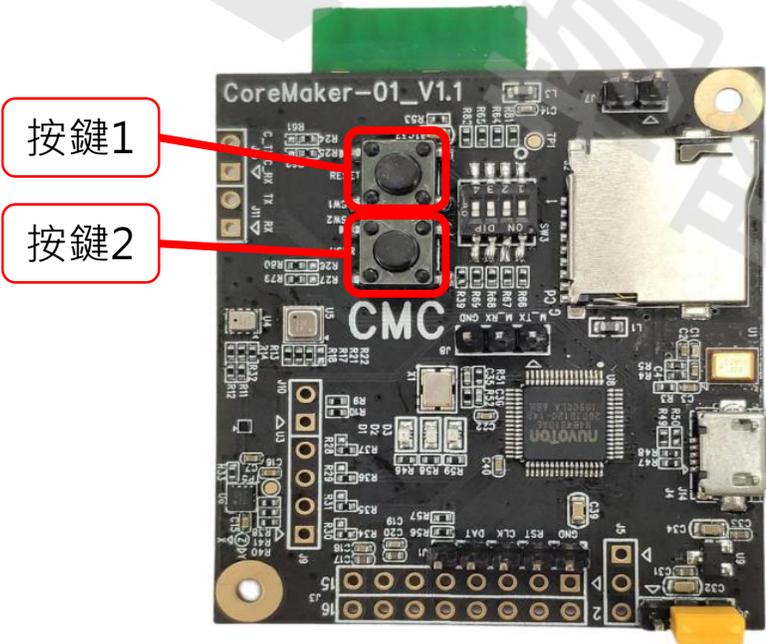


Sensor 設定完成後會自動跳回主畫面，繼續點選下方 Connect 設定連接埠。



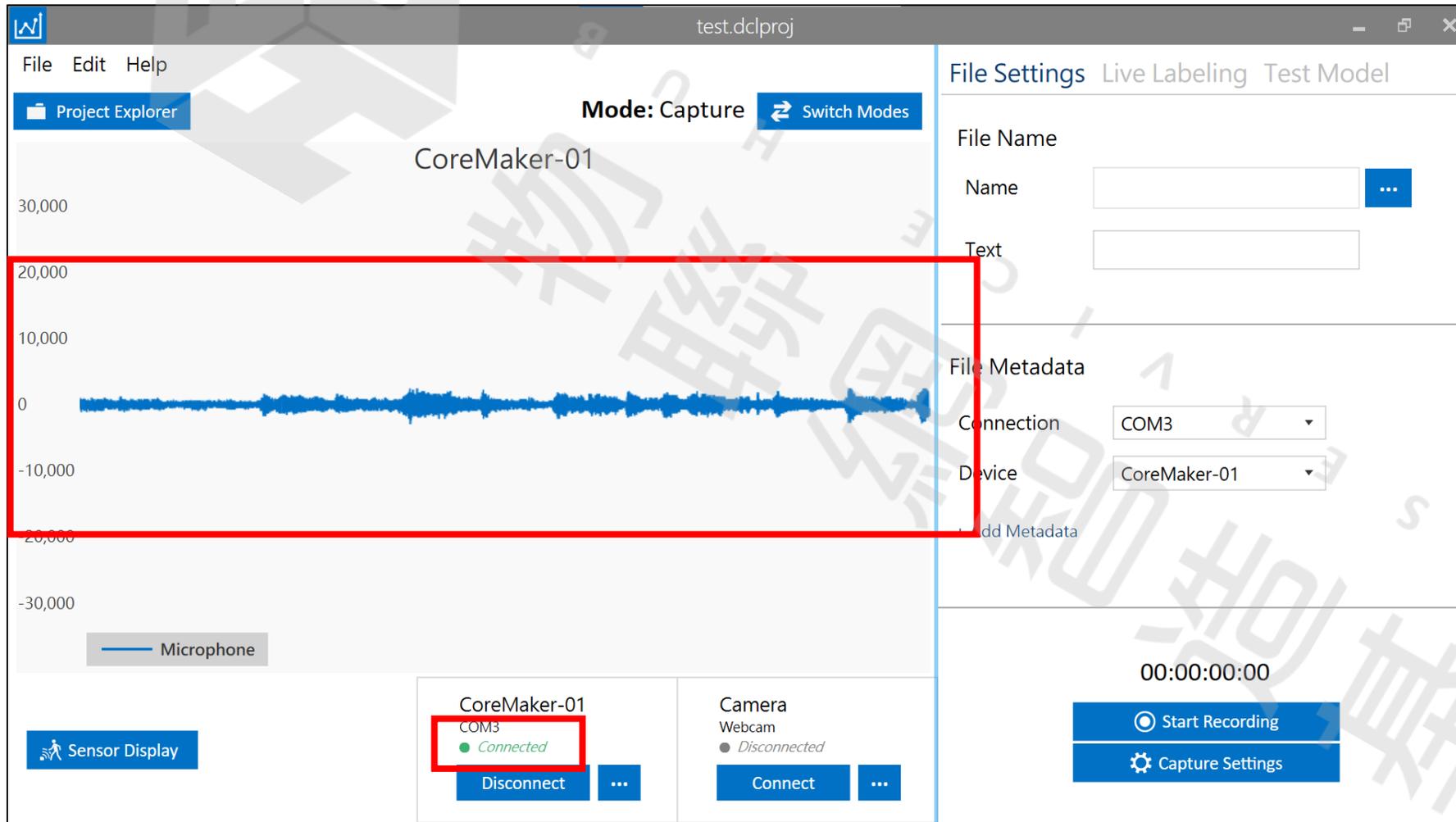
將CoreMaker-01以Micro-USB連結電腦後，再點擊 Scan，選擇板子所連接的Serial Port，點選 Done 完成。

3-4 聲音資料收集



連接埠設定完成後，再次點擊Connect，會出現 *Retrieving Configuration...*，此時按壓 CoreMaker-01 上的 按鍵2即可完成連線。

3-4 聲音資料收集

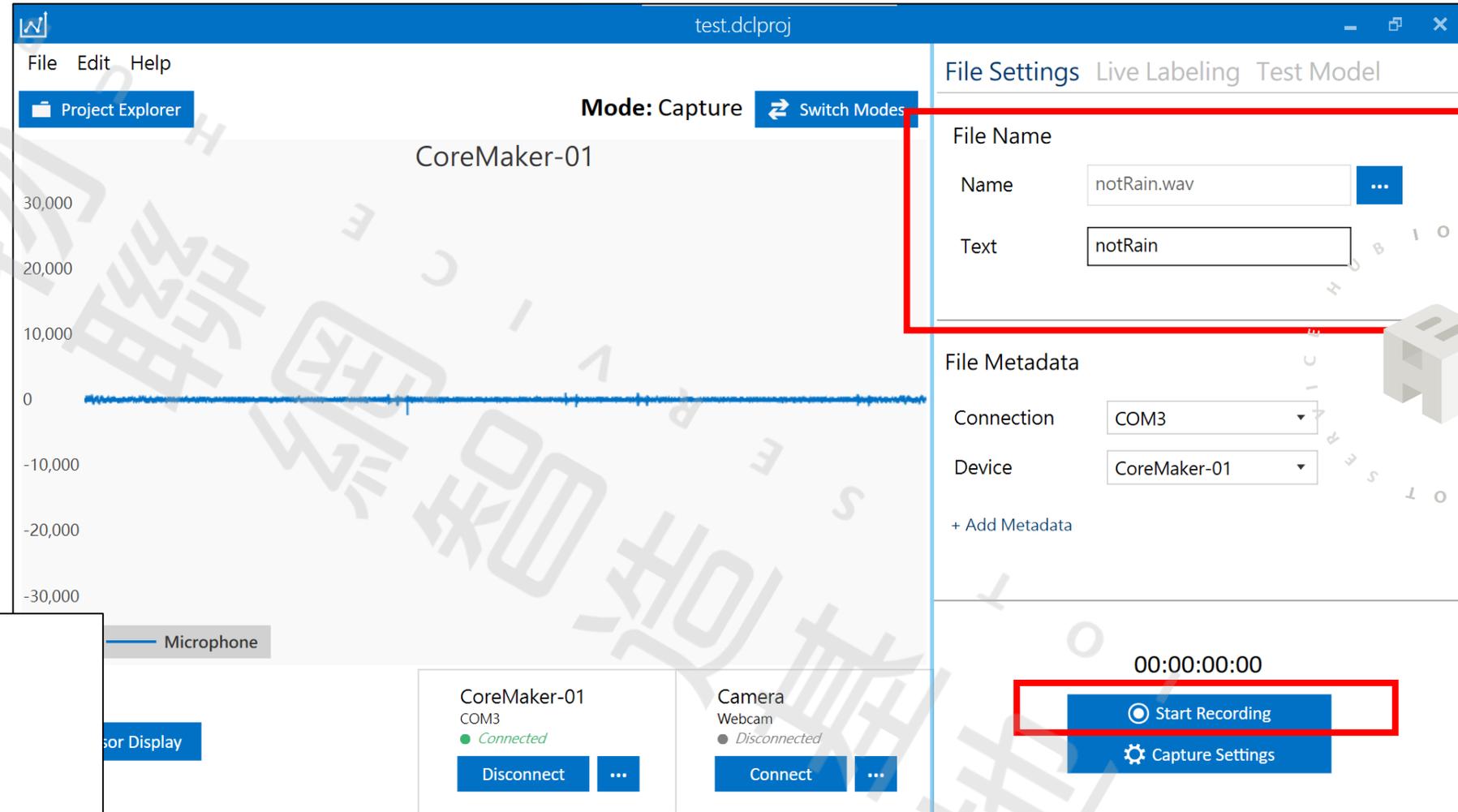


The screenshot displays the 'test.dclproj' application window. The main area shows a waveform for 'CoreMaker-01' with a red box highlighting it. The right sidebar contains 'File Settings', 'File Metadata', and recording controls. The bottom status bar shows 'CoreMaker-01 COM3' as 'Connected' with a green dot and a red box around it.

連接成功後
可看見下方出現綠色**Connected**，
以及主畫面中
我們聲音的波型

3-4 聲音資料收集

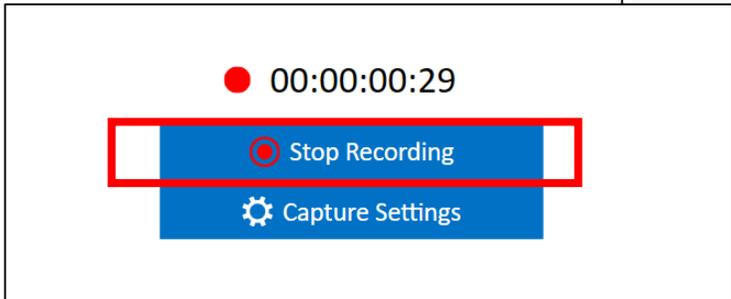
在右上方輸入即將要錄製的音檔檔名，輸入完成後，於右下方Start Recording 點擊開始錄製聲音，錄製完畢後按下 Stop Recording 停止錄音



The screenshot shows the CoreMaker-01 software interface. The main window displays a waveform for audio recording. On the right side, there is a settings panel with the following sections:

- File Settings:** Includes a 'File Name' section with a 'Name' field containing 'notRain.wav' and a 'Text' field containing 'notRain'.
- File Metadata:** Includes a 'Connection' dropdown set to 'COM3' and a 'Device' dropdown set to 'CoreMaker-01'.
- Recording Controls:** A 'Start Recording' button is highlighted with a red box, and a 'Stop Recording' button is also highlighted with a red box.

At the bottom of the interface, there are status indicators for 'Microphone', 'CoreMaker-01 COM3' (Connected), and 'Camera Webcam' (Disconnected). A timer at the top right shows '00:00:00:00'.



This close-up shows the recording controls. A red dot indicates the recording is in progress, with a timer showing '00:00:00:29'. Below the timer are two buttons: 'Stop Recording' and 'Capture Settings'.

3-4 聲音資料收集

Save Confirmation

File Name

Name ...

Text

File Metadata

Connection

Device

[+ Add Metadata](#)

Auto-Upload

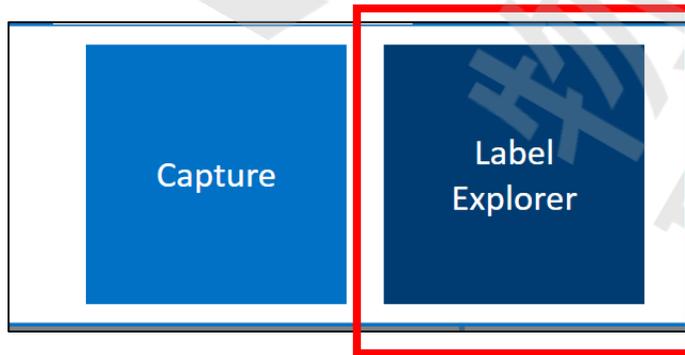
Do you want to save the sensor data?

錄製完畢後會跳出關於這次錄製音檔的訊息視窗，確認無誤後要勾選 **Auto-Upload** 上傳至 **SensiML** 的雲端，再點選 **Save** 儲存錄製的音檔。

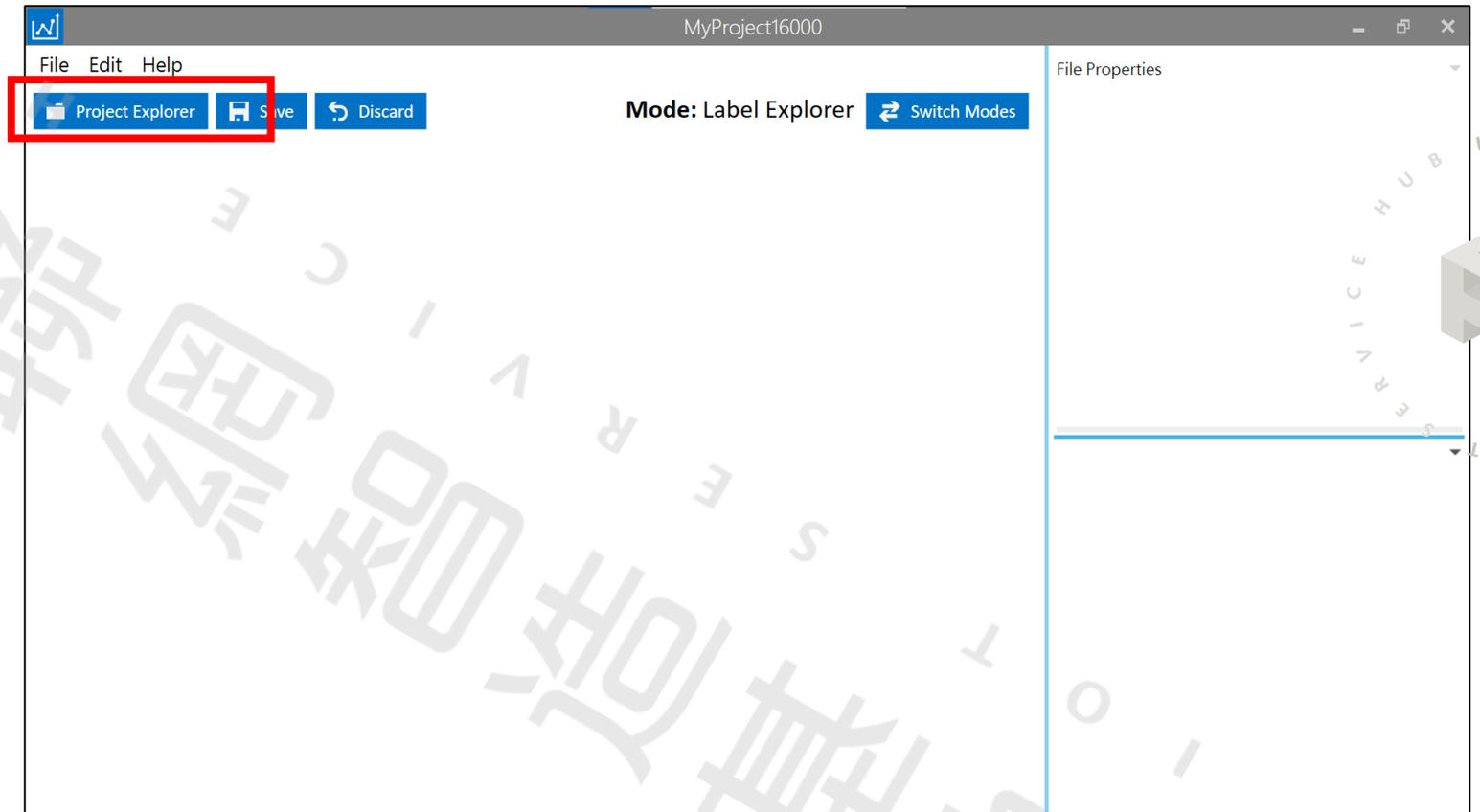
因為此範例只有是否為下雨聲，因此僅需錄製2個音檔，而為求達到更好的辨識效果，此範例每個錄音時間長達30分鐘，使用者可根據環境音的複雜程度或是開發時間來增減音檔的錄製長度

3-5 資料標註

接下來要標記我們錄製的音檔

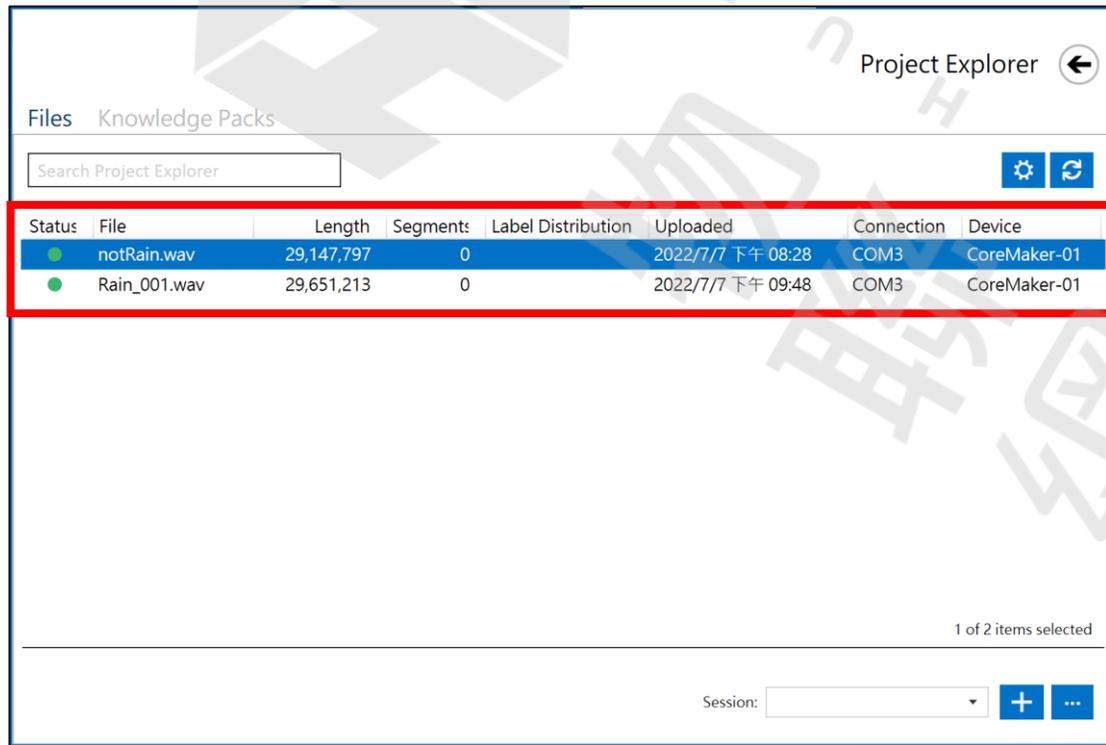


點選 Switch Modes，
選擇 Label Explorer。



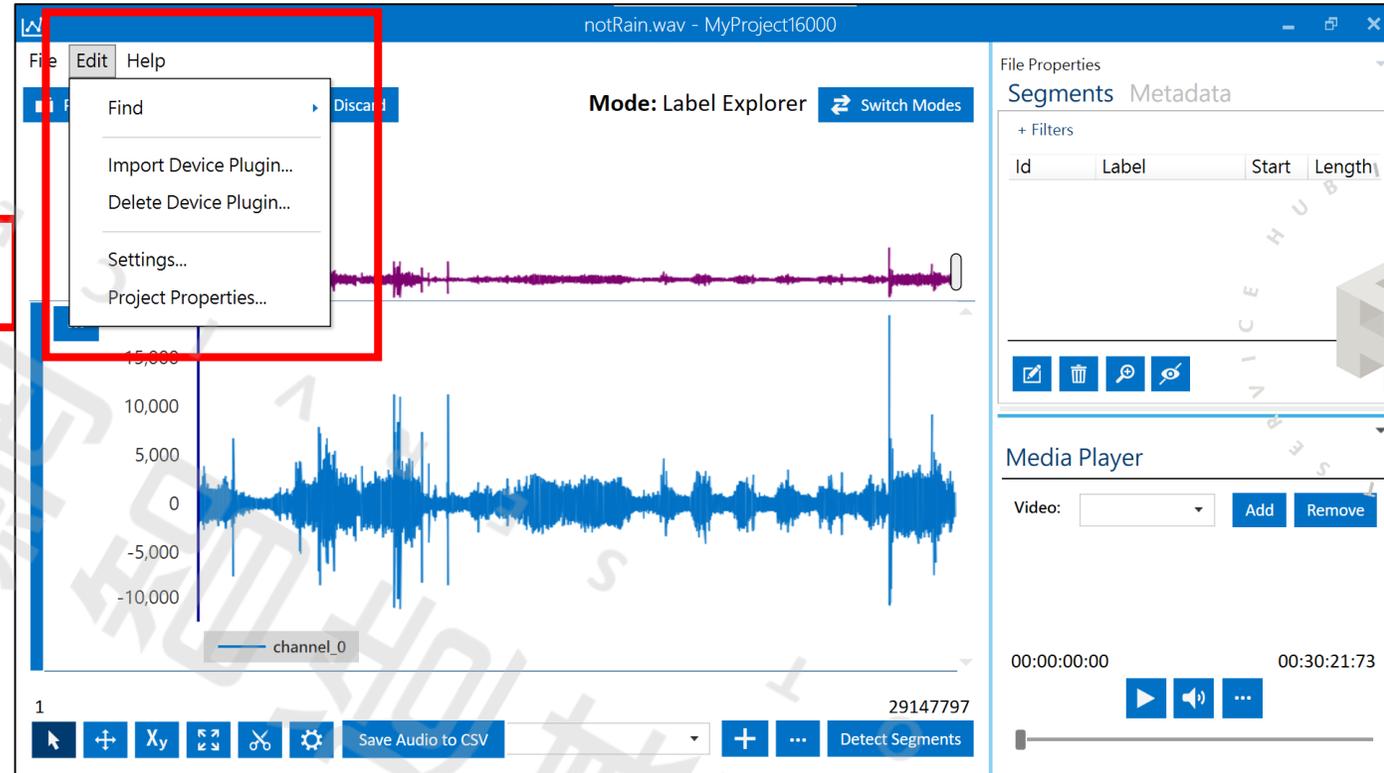
出現標記音檔模式的主畫面，點選 Project Explorer來開啟我們的音檔

3-5 資料標註



| Status | File | Length | Segments | Label Distribution | Uploaded | Connection | Device |
|--------|--------------|------------|----------|--------------------|-------------------|------------|--------------|
| ● | notRain.wav | 29,147,797 | 0 | | 2022/7/7 下午 08:28 | COM3 | CoreMaker-01 |
| ● | Rain_001.wav | 29,651,213 | 0 | | 2022/7/7 下午 09:48 | COM3 | CoreMaker-01 |

選擇我們要標記的音檔



notRain.wav - MyProject16000

Mode: Label Explorer

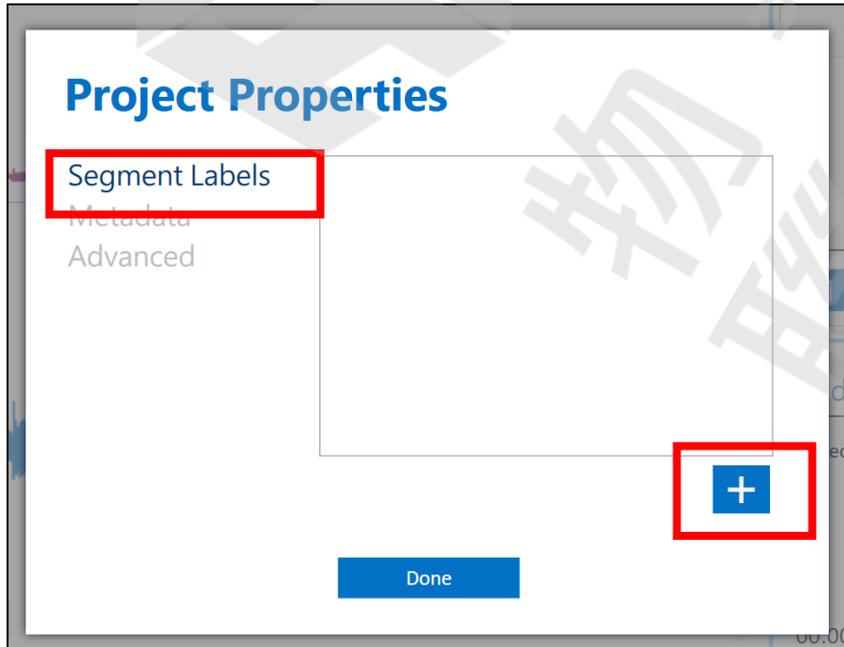
File Properties: Segments Metadata

Media Player: 00:00:00:00 / 00:30:21:73

先建立label名稱，點左上Edit，選擇 project properties

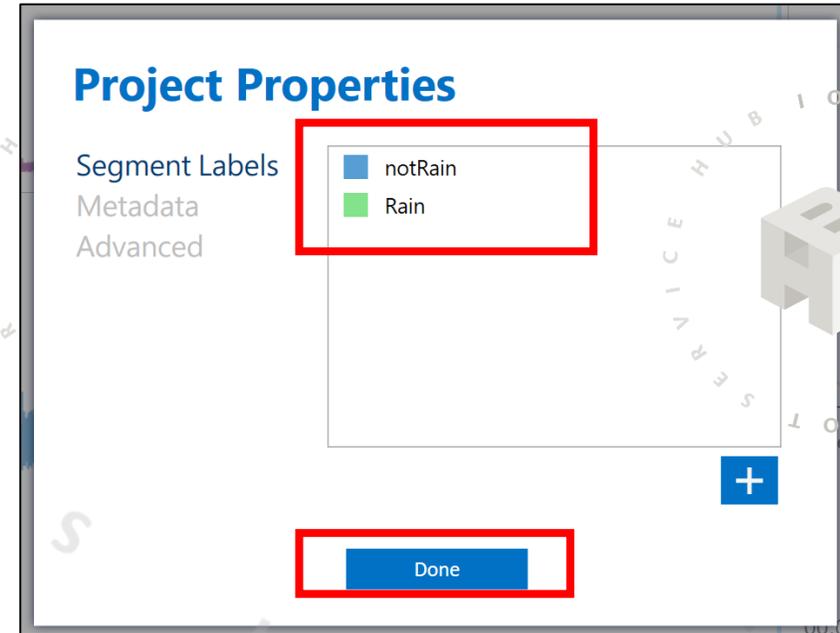
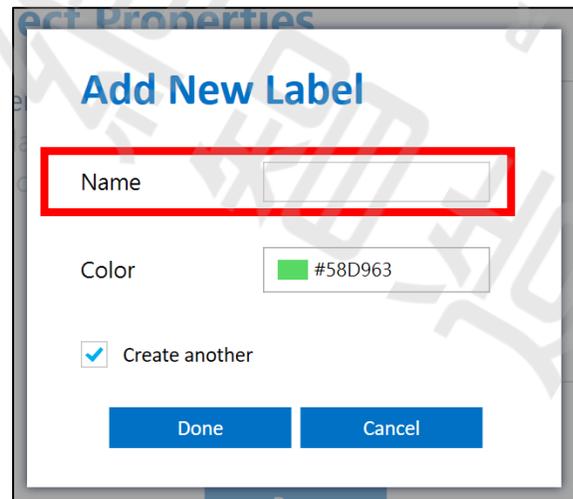
3-5 資料標註

此次範例要建立兩種類別，分別是雨聲以及非雨聲



選擇Segment labels，再點選 +號。

建立Label名稱，勾選Create another建立另外一個類別。



確認建立無誤後，點選Done完成設置。

3-5 資料標註



跳回標記音檔模式的主畫面
 點擊下方 session 的+號。

New Labeling Session

A labeling session separates events into their own group. This allows you to work on multiple use cases using the same dataset

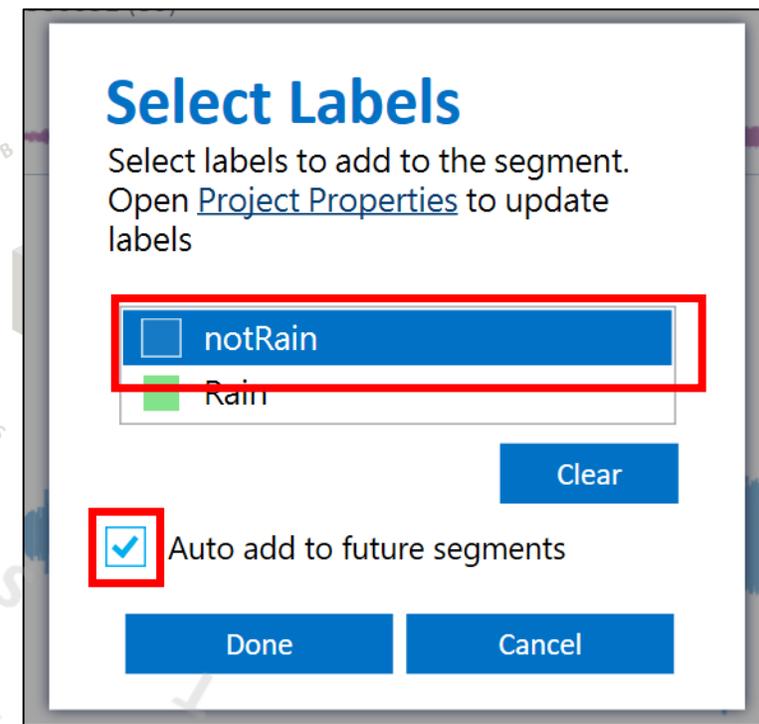
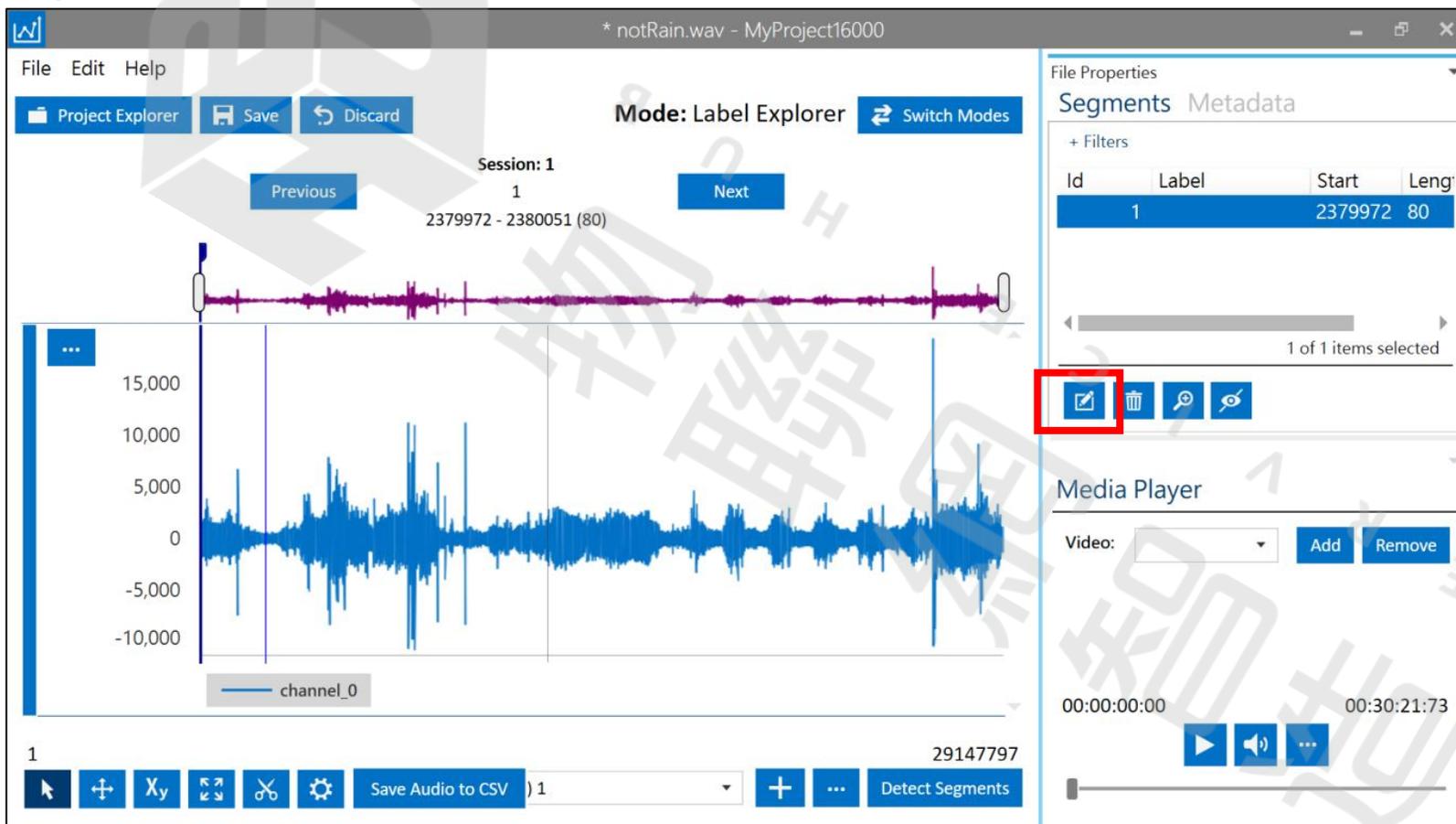
Name

Type
 Manual Auto

Manual:
 Create event labels by right-clicking on the graph. Segments can be edited by adjusting the location manually on the graph

自訂session 的名稱，
 Type點選Manual，
 下雨聲和非下雨聲的檔案標記要在
 相同的Session之中。

3-5 資料標註



在波型畫面中任意地方點擊右鍵，就會出現新的類別片段，於右方的segments資訊框中，在其左下角點擊edit來選擇我們在此音檔所使用的標籤。

選擇音檔所屬的Label
再勾選auto來自動標記相同的類別在接下來的片段中

3-5 資料標註

| Id | Label | Start | Length |
|----|---------|---------|--------|
| 1 | notRain | 2379972 | 90 |

Edit Location

Start

End

Lock Length

Done Cancel

在新設類別片段按右鍵，點選edit location，來設定擷取片段的位置以及長度。

輸入起始以及終止的時間位置，因為我們頻率設定16000赫茲，每段片段若以25秒來擷取，則總長度是400000，使用者可依案例情況調整擷取長度。

3-5 資料標註



File Edit Help

Project Explorer Save Discard

Mode: Label Explorer Switch Modes

Session: 1
1 - Rain
0 - 400000 (400001)

Previous Next

6,000
4,000
2,000
0
-2,000
-4,000
-6,000
-8,000

channel_0

1 29651213

Save Audio to CSV 1 Detect Segments

File Properties

Segments Metadata

+ Filters

| Id | Label | Start | Length |
|----|-------|---------|--------|
| 1 | Rain | 0 | 400000 |
| 2 | Rain | 500000 | 400000 |
| 3 | Rain | 1000000 | 400000 |
| 4 | Rain | 1500000 | 400000 |

1 of 59 items selected

Media Player

Video: Add Remove

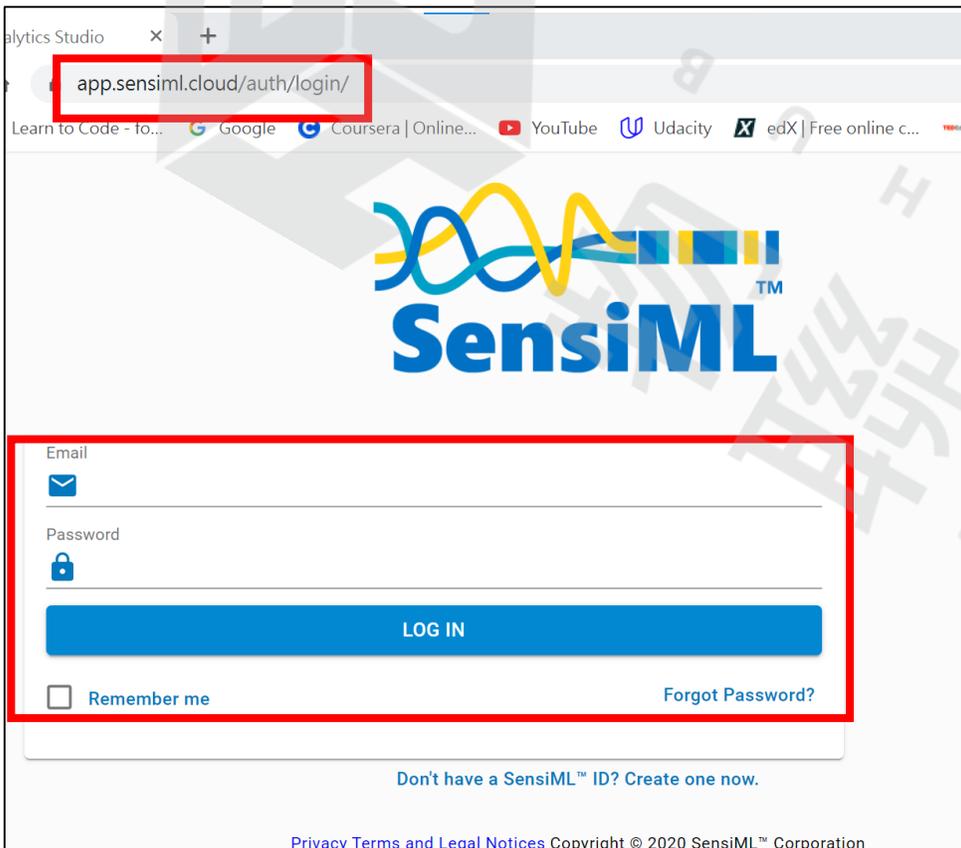
00:00:00:00 00:30:53:20

重複擷取步驟

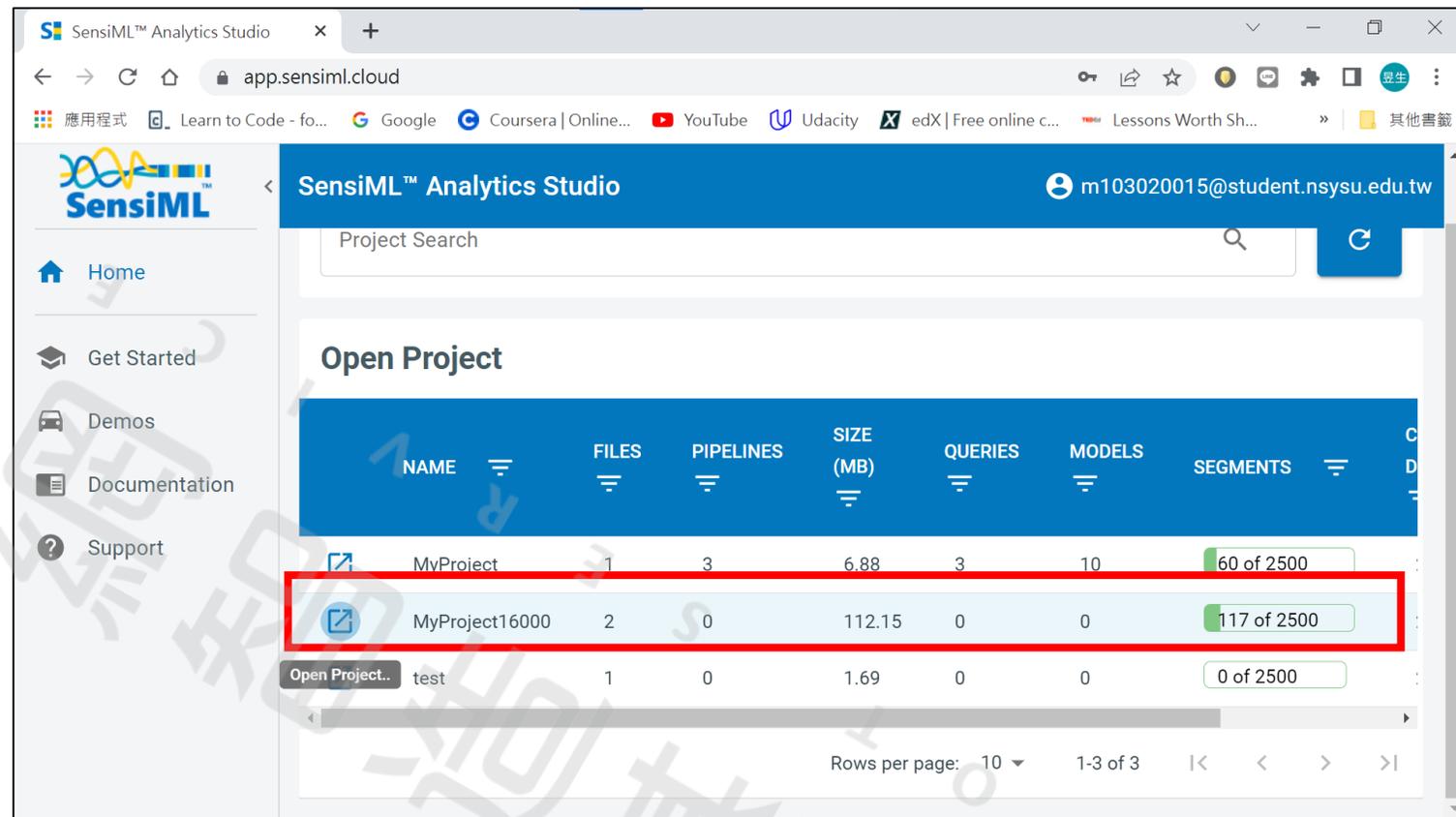
此次範例的片段時間為25秒，於30分鐘的音檔長度可以擷取約55到60個片段來供模型訓練。使用者也可自行增加音檔的錄音時間來增加類別片段，以提升我們模型的精準度。

按下右上方**Save**來儲存擷取片段資訊上傳至SensiML的雲端雲端

3-6 模型建立與下載

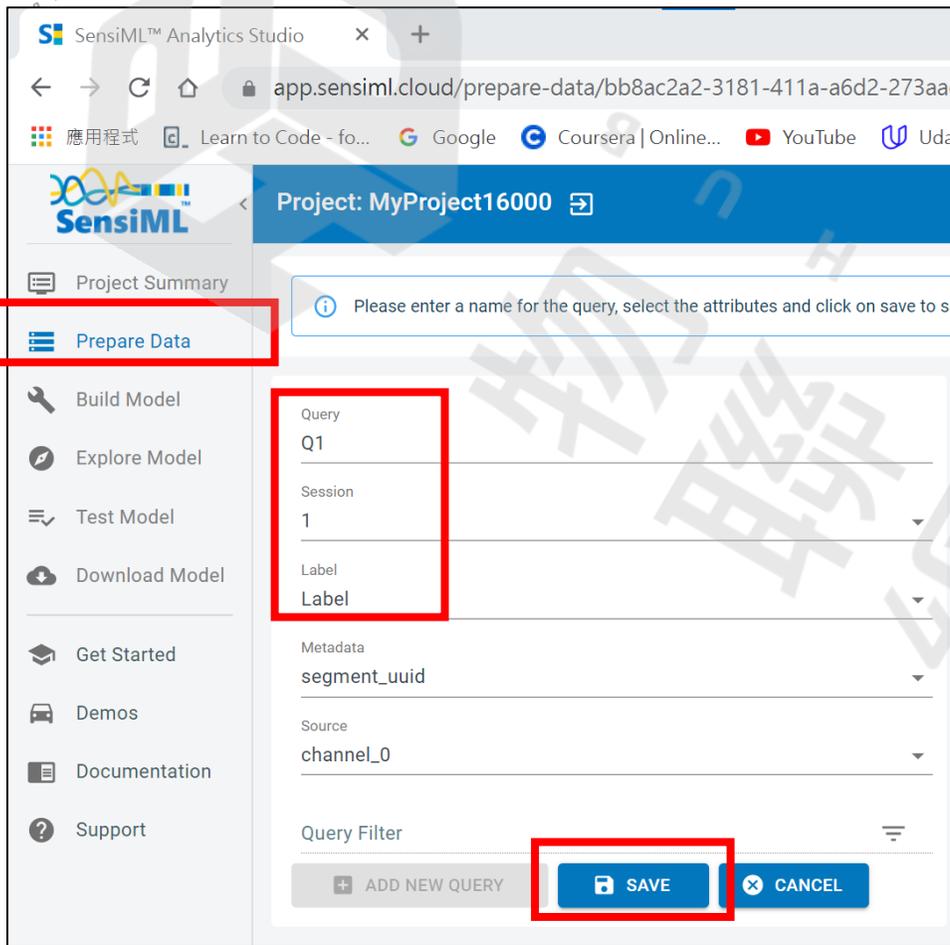


開啟 SensiML Analytics Studio 網頁：
<https://app.sensiml.cloud/auth/login/>
 登入註冊帳號。

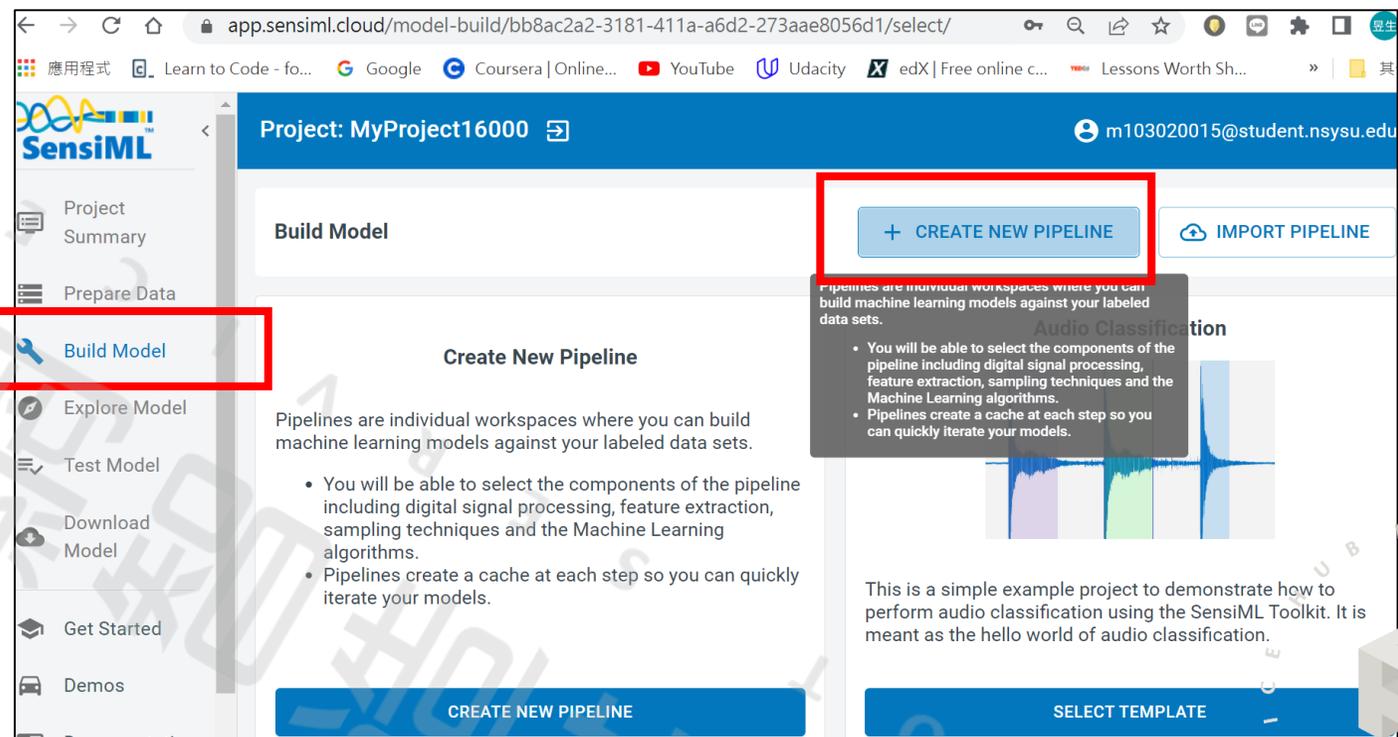


網頁主畫面，選擇我們要訓練模型的Project。

3-6 模型建立與下載

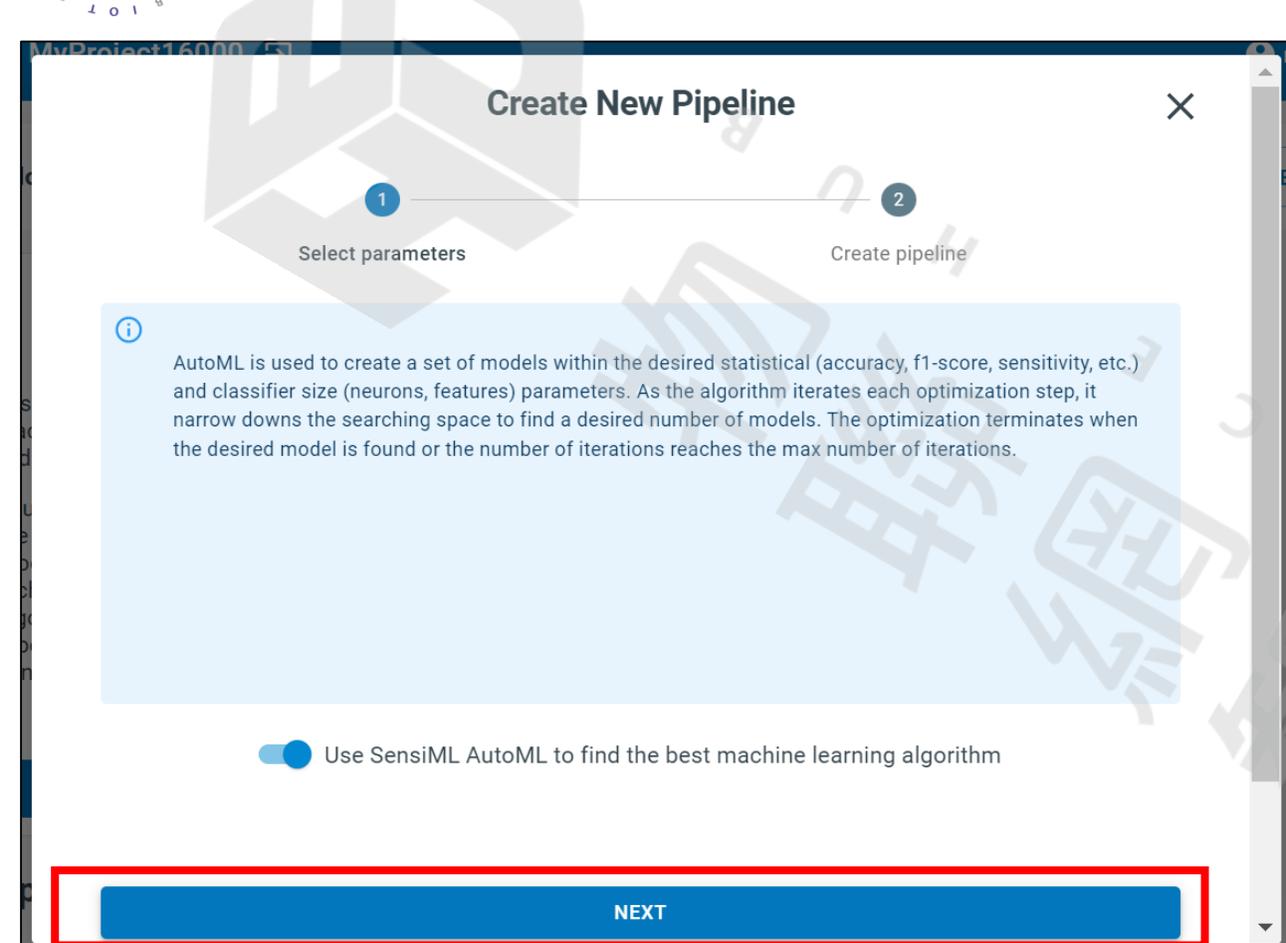


在左方目錄欄點選Prepare Data，
輸入Query 名稱(自訂)，選擇Session和Label，
完成後點擊Save。

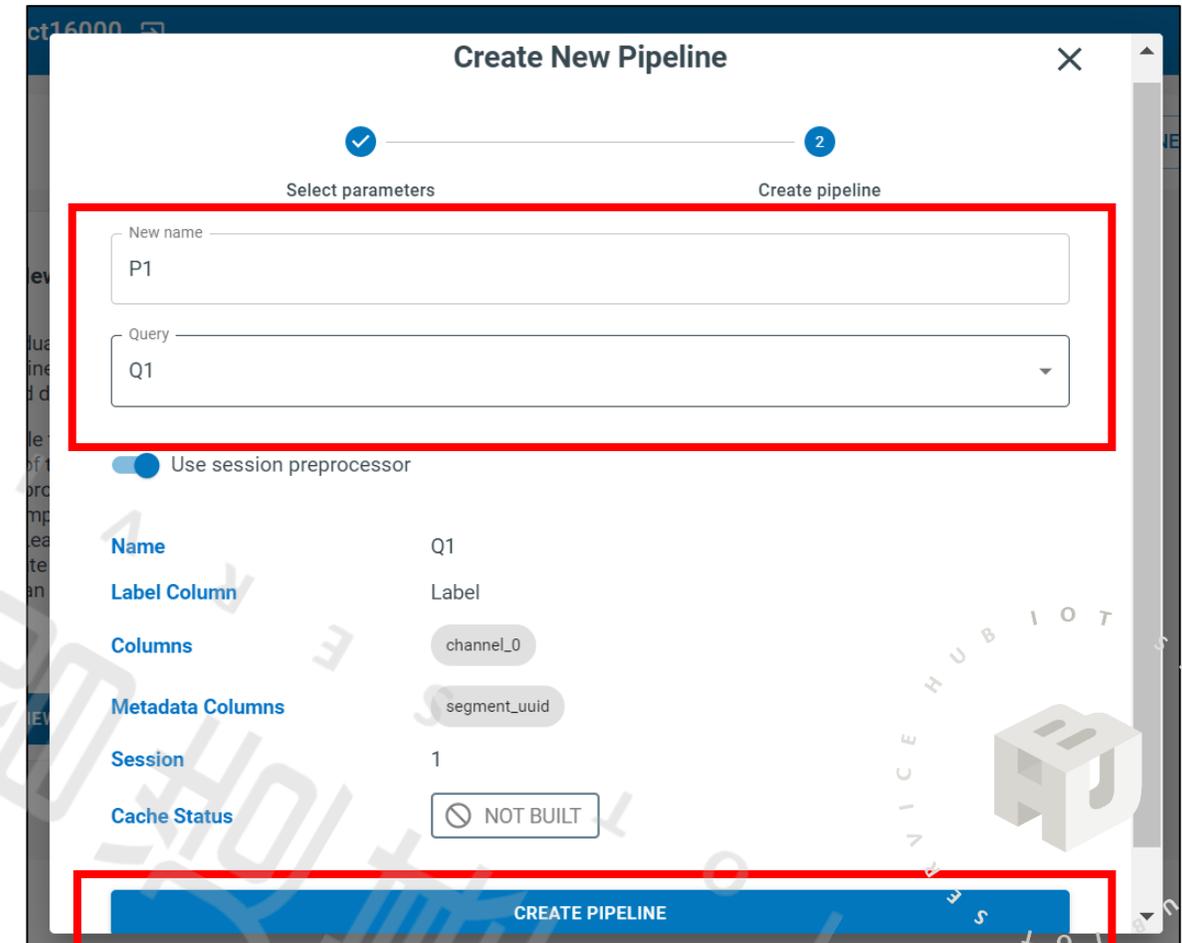


於左方目錄欄點選Build Model，
點選建立新的Pipeline。

3-6 模型建立與下載



跳出資訊欄，按Next



輸入pipeline的名稱(自訂)，選擇Query，完成後點選Create pipeline。

3-6 模型建立與下載

於頁面右方跳出 Pipeline 設定，點選Next。

Pipeline Created

The pipeline has been created. There are some parameters that we suggest reviewing to help you build the best model for your application. Click the next button to review/update the following parameters.

Name: [Windowing](#)

Type: [Segmenter](#)

Parameters: [Window Size](#), [Delta](#)

NEXT

Segmenter

Segmenter

Windowing has been set as the default Segmenter. The default window size is set to 1 second of data for your project. Please review/update the parameters and then click the save button to confirm.

Segmenter
Windowing

Window Size 16000
1 16384

Slide 16000
1 16384

CANCEL **SAVE**

設定辨識長度以及辨識的間隔時間，其中Window Size的數值要小於我們在音檔標記中擷取片段的最小長度(此次範例擷取長度達400000，因此兩者皆要小於此數)

3-6 模型建立與下載

Project: MyProject16000

Name: Q1
Type: Input Query

Name: Windowing
Type: Segmenter

Type: AutoML Parameters

| | |
|-----------------|-------|
| Classifier SRAM | 32000 |
| f1-score | 100 |

▶ OPTIMIZE

設定完成後點選OPTIMIZE開始建構模型

Project: MyProject16000

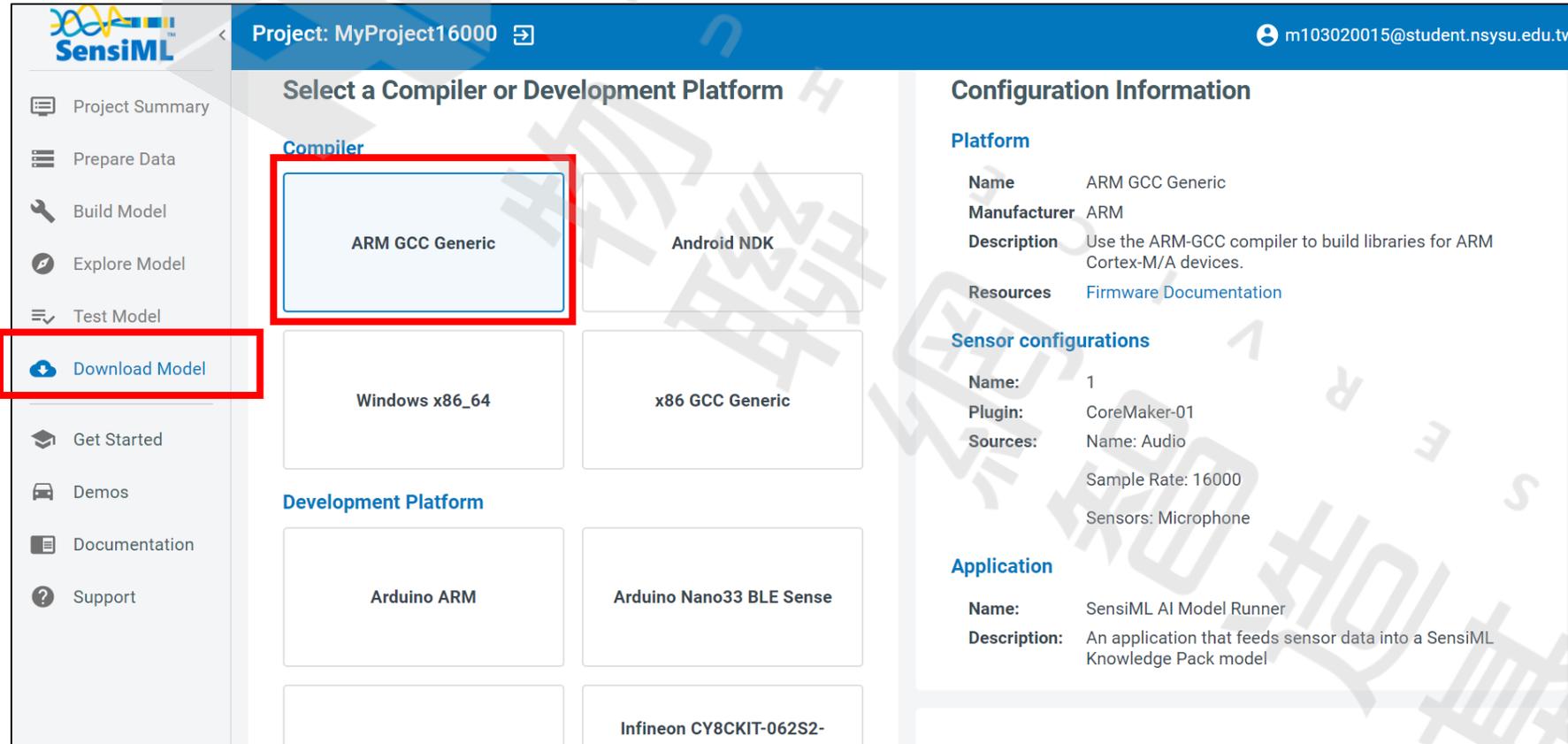
RESULT LOGS TRAINING SUMMARY

AutoML Results

| MODEL NAME | ACCURACY | CLASSIFIER SIZE(B) | NUM. FEATURES | SENSITIVITY | F1-SCORE |
|------------|----------|--------------------|---------------|-------------|----------|
| P1_rank_0 | 99 | 13360 | 8 | 99 | 99 |
| P1_rank_1 | 99 | 7220 | 13 | 99 | 99 |
| P1_rank_2 | 98 | 5780 | 12 | 98 | 98 |
| P1_rank_3 | 97 | 10030 | 8 | 97 | 97 |
| P1_rank_4 | 98 | 5466 | 12 | 98 | 98 |

運行完成後會顯示五個不同的模型，有各自不同的精確度和分類器大小等特徵供使用者挑選適合的模型，

3-6 模型建立與下載



SensiML Project: MyProject16000 m103020015@student.nsysu.edu.tw

Select a Compiler or Development Platform

Compiler

- ARM GCC Generic** (Selected)
- Android NDK

Development Platform

- Windows x86_64
- x86 GCC Generic
- Arduino ARM
- Arduino Nano33 BLE Sense
- Infineon CY8CKIT-062S2-

Configuration Information

Platform

- Name:** ARM GCC Generic
- Manufacturer:** ARM
- Description:** Use the ARM-GCC compiler to build libraries for ARM Cortex-M/A devices.
- Resources:** [Firmware Documentation](#)

Sensor configurations

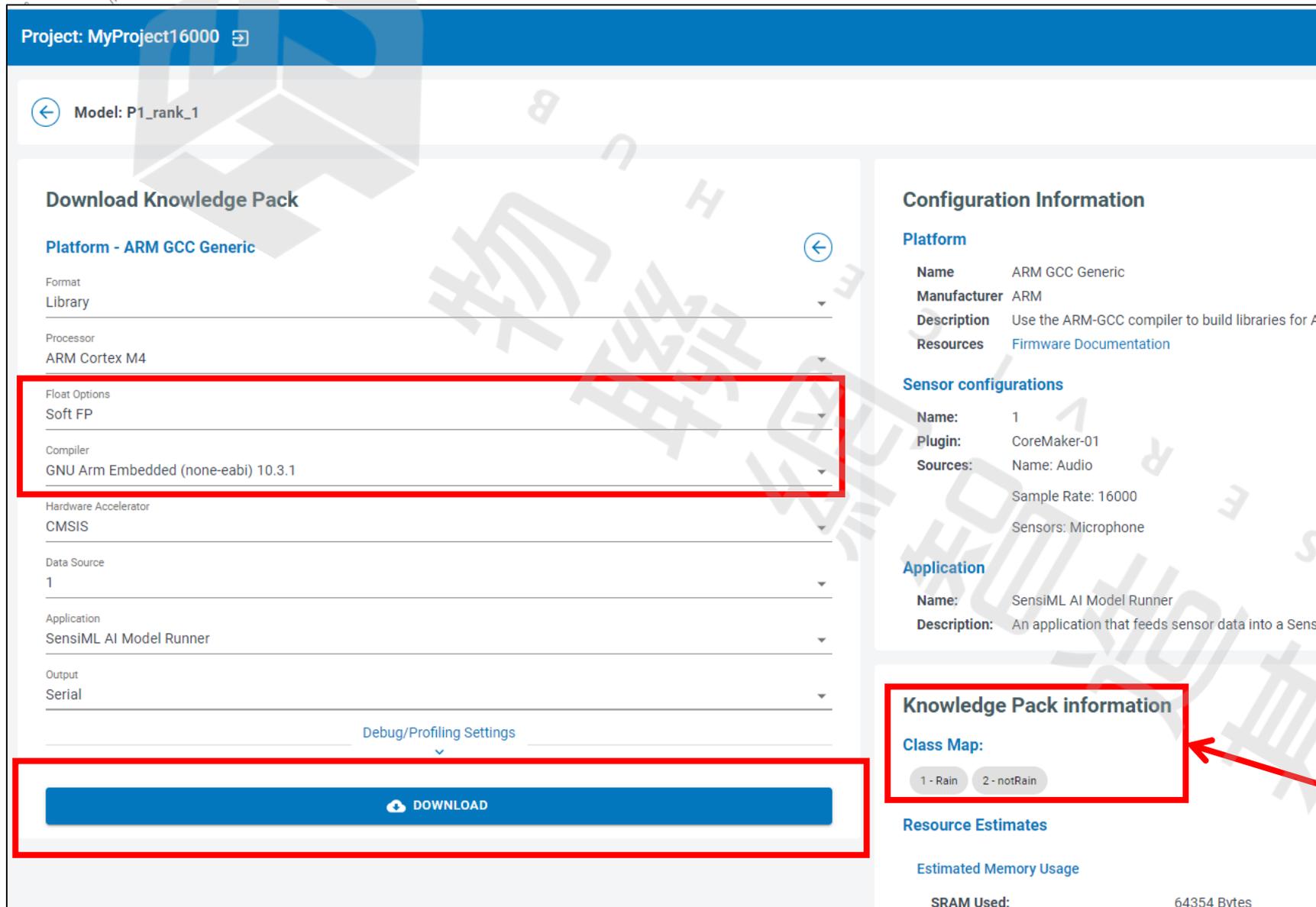
- Name:** 1
- Plugin:** CoreMaker-01
- Sources:** Name: Audio
Sample Rate: 16000
Sensors: Microphone

Application

- Name:** SensiML AI Model Runner
- Description:** An application that feeds sensor data into a SensiML Knowledge Pack model

在左方目錄欄點選 **Download Model**，選擇要下載的模型，**Compiler** 點選 **ARM GCC Generic** 後，按 **Select platform** 繼續下一頁設定。

3-6 模型建立與下載



Project: MyProject16000

Model: P1_rank_1

Download Knowledge Pack

Platform - ARM GCC Generic

Format: Library

Processor: ARM Cortex M4

Float Options: Soft FP

Compiler: GNU Arm Embedded (none-eabi) 10.3.1

Hardware Accelerator: CMSIS

Data Source: 1

Application: SensiML AI Model Runner

Output: Serial

Debug/Profiling Settings

DOWNLOAD

Configuration Information

Platform

Name: ARM GCC Generic
 Manufacturer: ARM
 Description: Use the ARM-GCC compiler to build libraries for A
 Resources: [Firmware Documentation](#)

Sensor configurations

Name: 1
 Plugin: CoreMaker-01
 Sources: Name: Audio
 Sample Rate: 16000
 Sensors: Microphone

Application

Name: SensiML AI Model Runner
 Description: An application that feeds sensor data into a Sensi

Knowledge Pack information

Class Map:

1 - Rain 2 - notRain

Resource Estimates

Estimated Memory Usage

SRAM Used: 64354 Bytes

更改部分:

Float options → Soft FP
 Compiler → GNU 10.3.1

維持預設:

Format → Library
 Processor → ARM Cortex M4

按下Download
 下載模型的壓縮檔。

模型代表數字
 此模型輸出1代表下雨聲
 輸出2代表非下雨聲

3-6 模型建立與下載

把下載下來的檔案解壓縮後，
將裡面libsensiml 資料夾
替換掉CoreMaker-01
原本的libsensiml 資料夾

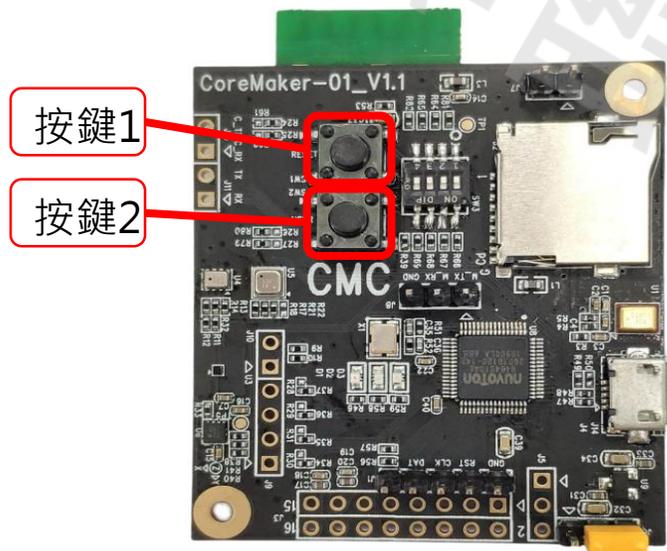
本機磁碟 (C:) > 使用者 > 88692 > CoreMaker-01

搜尋 CoreMaker-01

| 名稱 | 修改日期 | 類型 | 大小 |
|-------------------|-------------------|---------------|-------|
| .git | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| CMC_ISP | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| docs | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| libsensiml | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| mbed-os | 2022/8/2 下午 04:13 | 檔案資料夾 | |
| nvt_rs485 | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| sd_card | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| SensiML | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| sensors | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| TARGET_AIOT2101 | 2022/8/2 下午 04:05 | 檔案資料夾 | |
| .gitignore | 2022/8/2 下午 04:05 | 文字文件 | 1 KB |
| .gitmodules | 2022/8/2 下午 04:05 | 文字文件 | 1 KB |
| CMakeLists | 2022/8/2 下午 04:05 | 文字文件 | 2 KB |
| custom_targets | 2022/8/2 下午 04:05 | JSON 來源檔案 | 1 KB |
| DebounceIn | 2022/8/2 下午 04:05 | C++ 來源檔案 | 3 KB |
| DebounceIn | 2022/8/2 下午 04:05 | C Header 來源檔案 | 2 KB |
| global | 2022/8/2 下午 04:05 | C Header 來源檔案 | 2 KB |
| jRead | 2022/8/2 下午 04:05 | C 來源檔案 | 23 KB |

3-6 模型建立與下載

再進行一次編譯以及燒錄



燒錄完成後，以USB轉TTL序列傳輸線連接電腦以及 CoreMaker-01，開啟連線程式(如Tera Term)，連線設定完成後，按下 CoreMaker-01上的 SW2 鍵，可在螢幕視窗上看到 AI 辨識的分類結果。

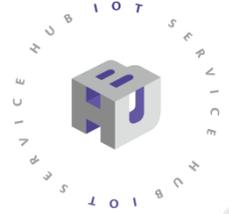
3-7 Arduino程式下載和介紹

```
CoreMaker-01
1 #include <Servo.h>
2 |
3 Servo myservo;
4 String cmd, str;
5
6 void setup() {
7   // put your setup code here, to run once:
8   Serial.begin(115200);
9   Serial.println("Arduino Ready!");
10  myservo.attach(9);
11  pinMode(LED_BUILTIN, OUTPUT);
12 }
13
```

```
14 void loop() {
15   if(Serial.available()){
16     //Serial.write(Serial.read());
17     cmd = Serial.readString();
18     str = cmd.substring(26,27);
19     Serial.println(cmd);
20     Serial.println(str);
21     if(cmd.substring(26,27) == "1"){
22       Serial.println('3');
23     }else{
24       myservo.attach(9);
25       myservo.write(30); //旋轉到0度，就是一般所說的歸零
26       delay(1000);
27       myservo.write(150); //旋轉到180度
28       delay(1000);
29       myservo.detach();
30     }
31
32   }
33 }
```

4. 成果示意圖





Thank you

