



物聯網智造基地

I O T S E R V I C E H U B

國產IC開發套件- CoreMaker-01

開源示範案例- 智慧語音電梯按鈕

指導單位:  經濟部工業局

INDUSTRIAL DEVELOPMENT BUREAU
MINISTRY OF ECONOMIC AFFAIRS
經濟部工業局

作者: 徐瑞茂



資訊工業策進會 Institute for Information Industry

大綱

- 一、應用情境概述
- 二、系統介紹
- 三、開發環境設定
- 四、韌體設定與修改
- 五、範本錄製與取樣
- 六、AI模型製作
- 七、Arduino程式編寫
- 八、成果展示



一、應用情境概述

- COVID-19疫情流行期間，需要以電梯來進出的大樓住戶，即可透過CoreMaker-01所提供的語音辨識功能來控制電梯按鈕(本例以HMI觸控螢幕模擬)，如此便可降低住戶因觸摸電梯按鈕而遭到感染的風險。



****注意****

本案例僅示範一種聲音模型訓練，聲音模型跟訓練者個人聲紋有關，故訓練完成後若非訓練者本人使用可能會影響辨識效果。若要建立通用模型，需多取樣不同人的聲音以提高辨識率。

二、系統介紹 – 運作流程

語音輸入樓層



CoreMaker-01

回傳辨識結果



Arduino
Mega-2560

根據辨識結果
確定/取消樓層

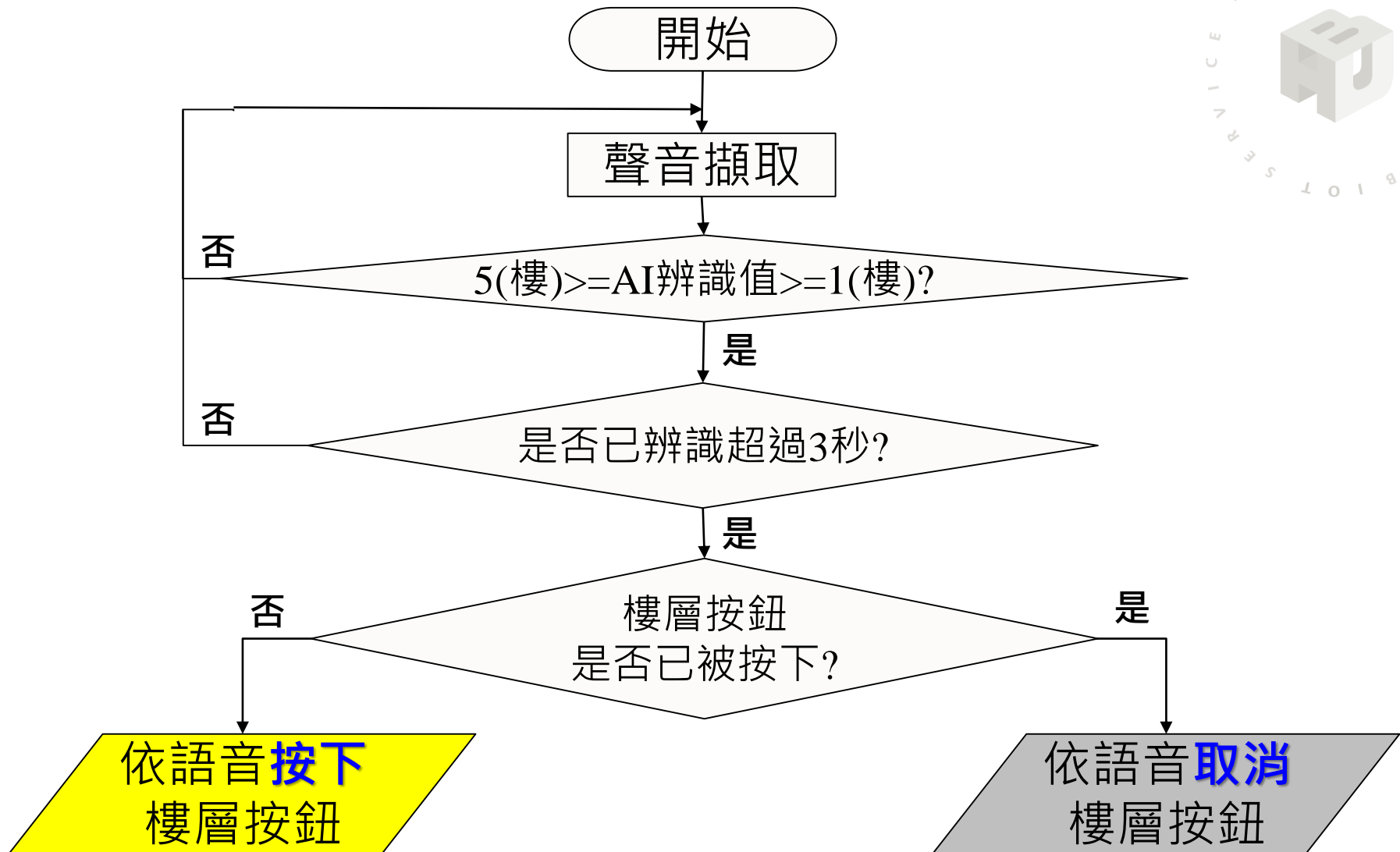


HMI
觸控面板

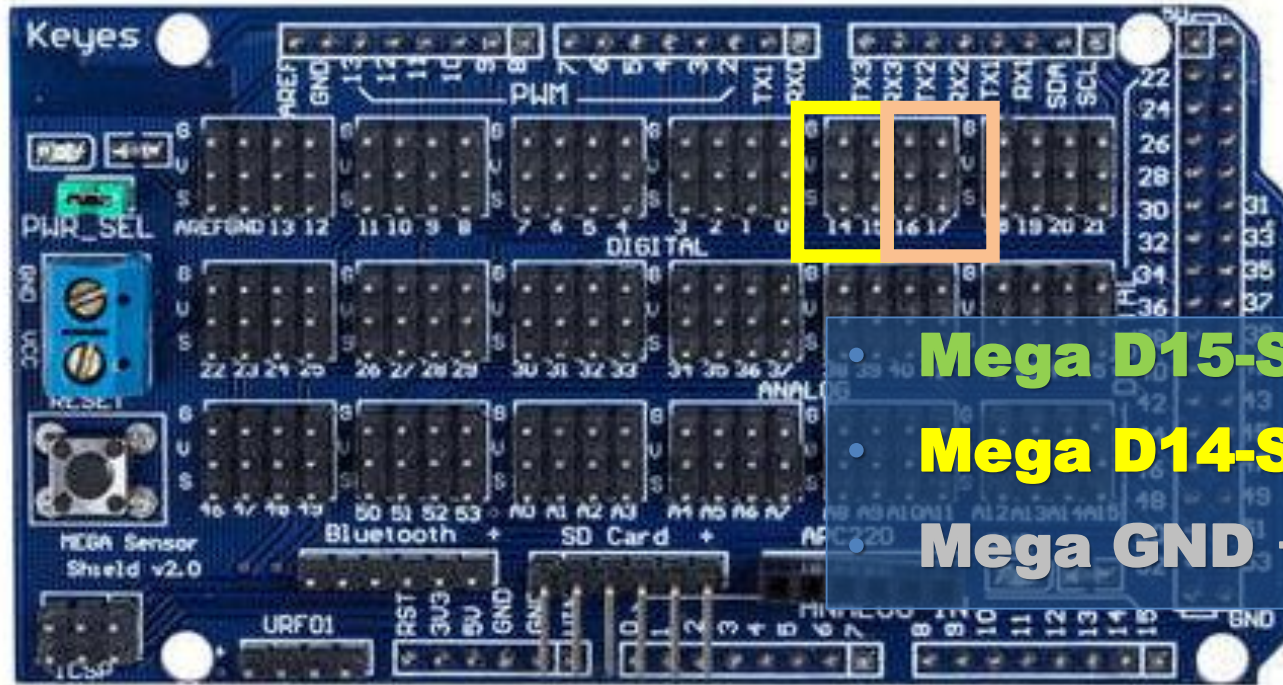


根據控制板命令來
點亮/熄滅樓層按鈕

二、系統介紹 – 系統動作流程圖

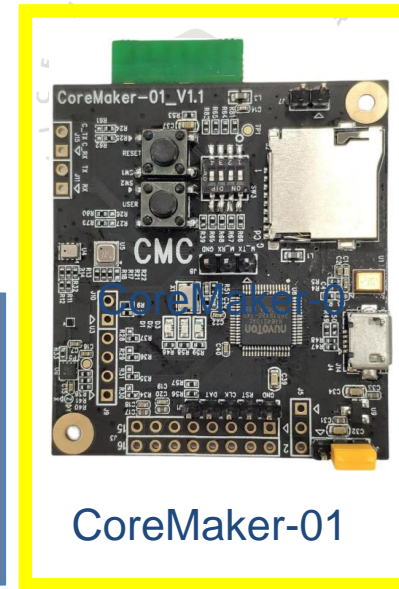


二、系統介紹 – 硬體接線



Arduino Mega-2560

- **Mega D15-S → W_TX**
- **Mega D14-S → W_RX**
- **Mega GND → GND**

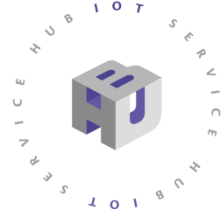


CoreMaker-01

- **Mega VCC → HMI 5V**
- **Mega D17-S → HMI TX**
- **Mega D16-S → HMI RX**
- **Mega GND → HMI GND**

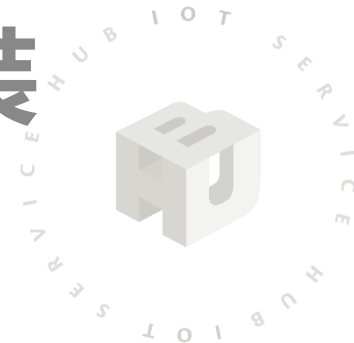


HMI
觸控面板



三、開發環境設定 I

作業環境要求與套件安裝



- 作業系統

- ◆ Windows 10

- 須安裝套件

- ◆ Git

(下載韌體原始檔使用)

- ◆ Python

(韌體編譯使用)

- ◆ Cmake

(韌體編譯使用)

- ◆ Ninja

(韌體編譯使用)

- ◆ Mbed CLI 2

(韌體編譯使用)

- ◆ GNU Arm Embedded Toolchain

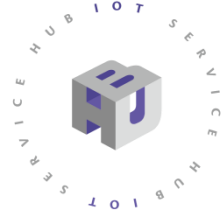
(韌體編譯使用)

****安裝步驟：**

→ <https://github.com/CoretronicMEMS/CoreMaker-01/>

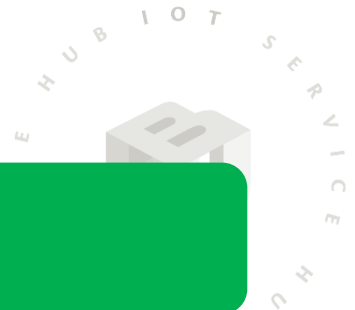
→ docs資料匣

→ coremaker操作指南_V1.1.pdf



三、開發環境設定 II

下載韌體原始程式



在”命令提示字元”下，輸入”**git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git**”

```
命令提示字元
Microsoft Windows [版本 10.0.18362.267]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\kc.hu> git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git
Cloning into 'CoreMaker-01'...
remote: Enumerating objects: 438, done.
remote: Counting objects: 100% (438/438), done.
remote: Compressing objects: 100% (184/184), done.
remote: Total 438 (delta 267), reused 420 (delta 249), pack-reused 0 eceiving objects: 88% (386/438), 5.83 M
Receiving objects: 100% (438/438), 6.89 MiB | 1.75 MiB/s, done.
Resolving deltas: 100% (267/267), done.
Submodule 'mbed-os' (https://github.com/CoretronicMEMS/mbed-os.git) registered for path 'mbed-os'
Cloning into 'C:/Users/kc.hu/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), 465.20 MiB | 3.00 MiB/s, done.
Resolving deltas: 100% (293263/293263), done.
Submodule path 'mbed-os': checked out '875ce5bfca68fd74d22ffec266e542d0ebadc0b6'


C:\Users\kc.hu>
```




三、開發環境設定 III

安裝資料擷取工具



[Products](#) [Solutions](#) [Plans](#) [Resources](#) [Support](#) [Company](#) [Developers](#) [Sign In](#) [GET PROTO EDITION](#)

SensiML Downloads

Get the Latest Application Releases For Building Intelligent IoT Endpoints

Don't Have A SensiML Account Yet?


SensiML applications require an active account to function.

[Get For Free](#)

<https://sensiml.com/download/>
下載 SensiML Data Capture Lab → 安裝

SensiML Data Capture Lab


Capture, cleanse, label, and annotate your custom IoT sensor datasets.



Windows 10
Version: 2022.4.0
[Release Notes](#)

SensiML Analytics Studio

AutoML application to easily generate device-optimized IoT firmware from your labeled datasets.

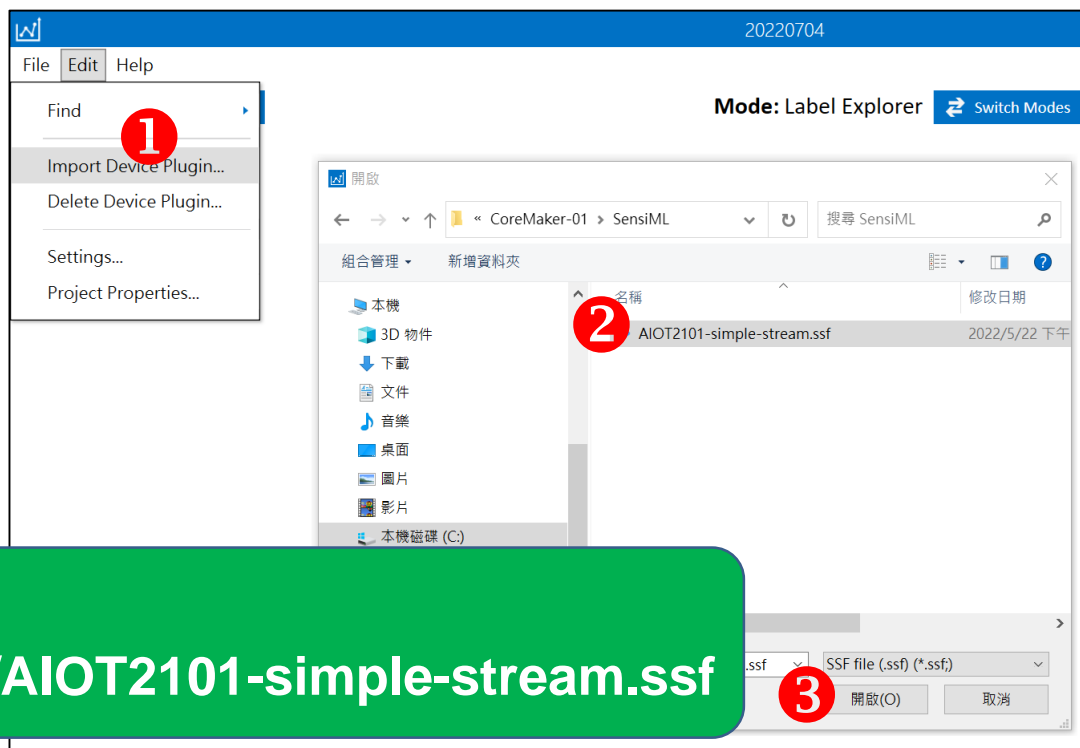
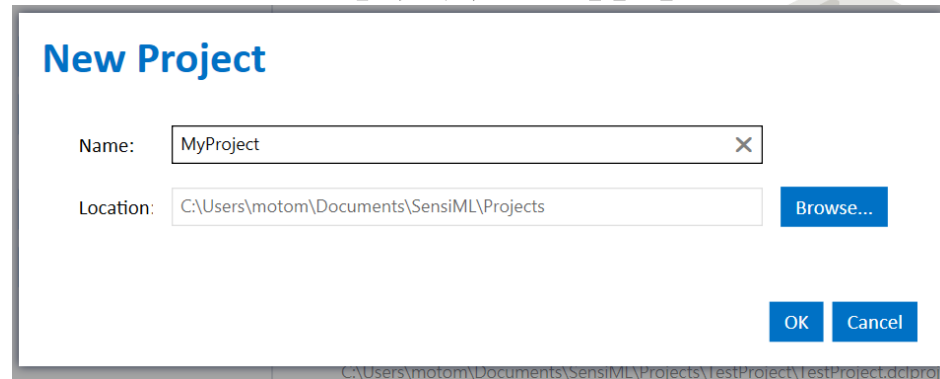


Web Application
No download required.



四、韌體設定與修改！ 匯入CoreMaker設定檔

- 開啟 SensiML Data Capture Lab
- 登入帳號
- 建立專案



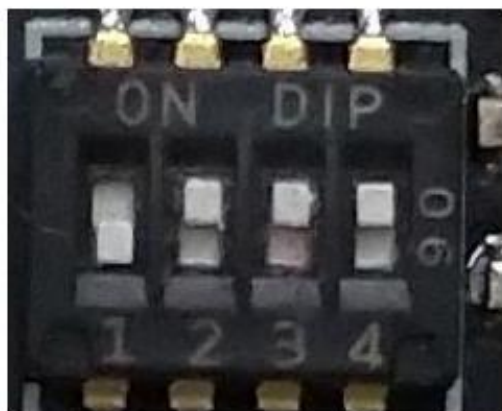
- 匯入 CoreMaker 設定檔
檔案位置在 CoreMaker-01/SensiML/AIOT2101-simple-stream.ssf

四、韌體設定與修改 II

韌體程式修改1

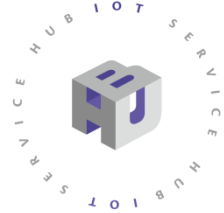
- 使用編輯器開啟 CoreMaker-01/sensors/SensorHub.cpp
- 將第 56 行麥克風的取樣率(sample rate)設定為16000 (取樣率越大聲音越清楚)。
- acoustic_node : 聲音，可設定值為 [100, 200, 400, 1000, 2000, 4000, 8000, 16000]

•DIP Switch調成聲音模式



聲音

```
typedef int (*RUN_MODEL)(SENSOR_DATA_T *, int , int );
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);
```



四、韌體設定與修改 III

韌體程式修改2



- 繼續修改CoreMaker-01/sensors/SensorHub.cpp
- 將語音識別模型回傳的數值直接從UART傳送出來(不需夾雜其他文字)

```
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(short), 0);
258             if (ret > -1)
259             {
260                 //printf("AI classification result: %d\n", ret);
261                 printf("%d\n", ret);
262                 kb_reset_model(0); // Reset running model to initial state.
263             }
264             else if (ret == -2)
265                 printf("This segment has been filtered.\n");
266             else if (ret < -2)
267                 printf("AI error: %d\n", ret);
268         }
```

四、韌體設定與修改 IV

韌體重新編譯

- 進入電腦的”命令提示字元”模式，並輸入“**cd CoreMaker-01**”進入韌體程式所在的資料夾
- 輸入”**mbed-tools compile -m AIOT2101 -t GCC_ARM**”開始編譯，初次編譯需要較長的時間
- 編譯完成會在 **CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM** 內，產生 **AIOT_2101.bin**

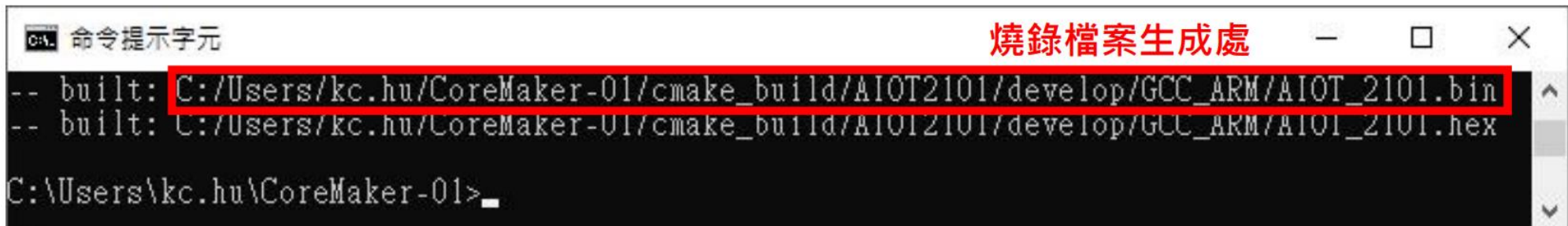


```

Microsoft Windows [版本 10.0.18362.267]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\kc.hu>cd CoreMaker-01
C:\Users\kc.hu\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
  
```

編譯指令



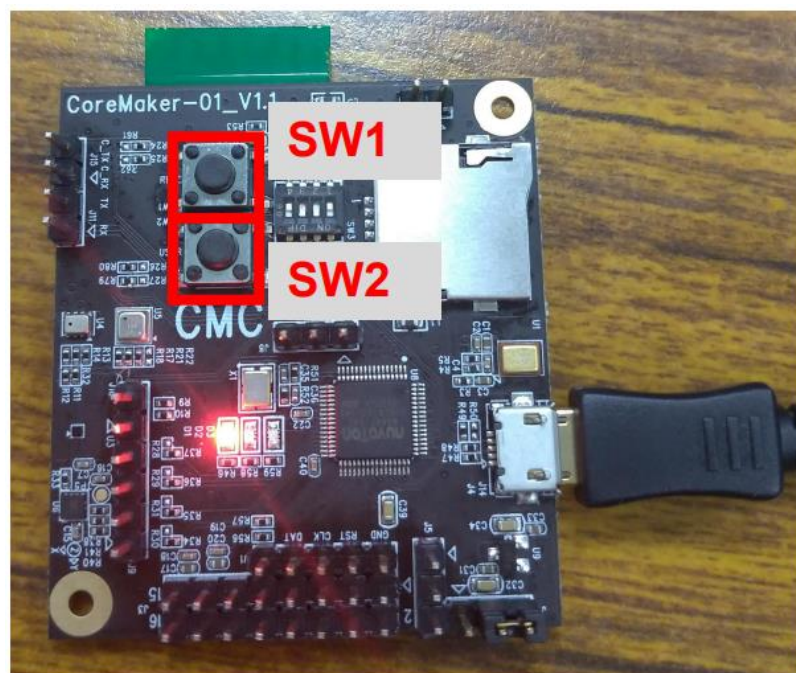
```

-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex

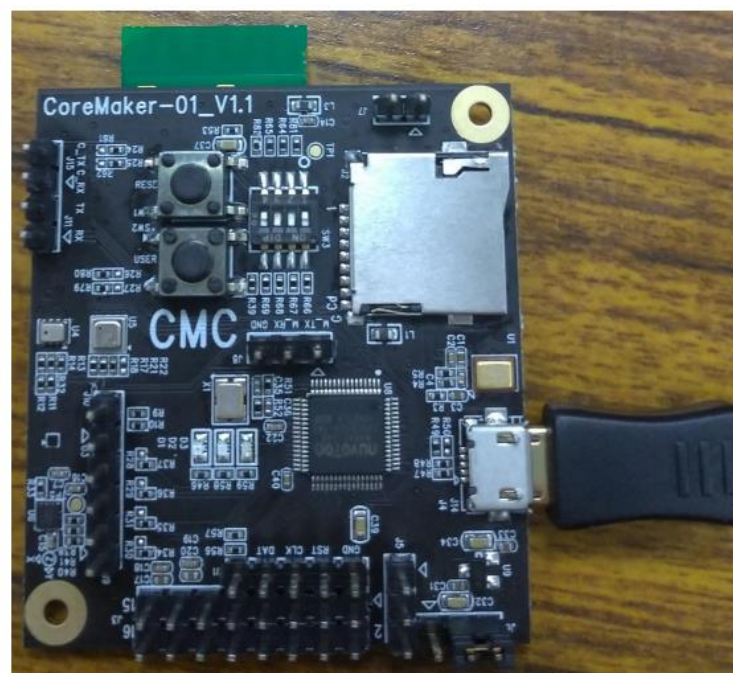
C:\Users\kc.hu\CoreMaker-01>
  
```


四、韌體設定與修改 V 進入燒錄模式

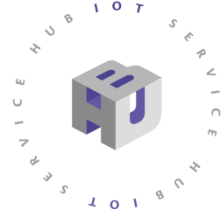
1. 使用 MicroUSB to USB 轉接線，連接 CoreMaker 與 PC
2. 同時按下 SW1 及 SW2
3. 先放開 SW1，再放開 SW2
4. 當 CoreMaker 上的紅色 LED 不再閃爍時，進入燒錄模式。



一般模式



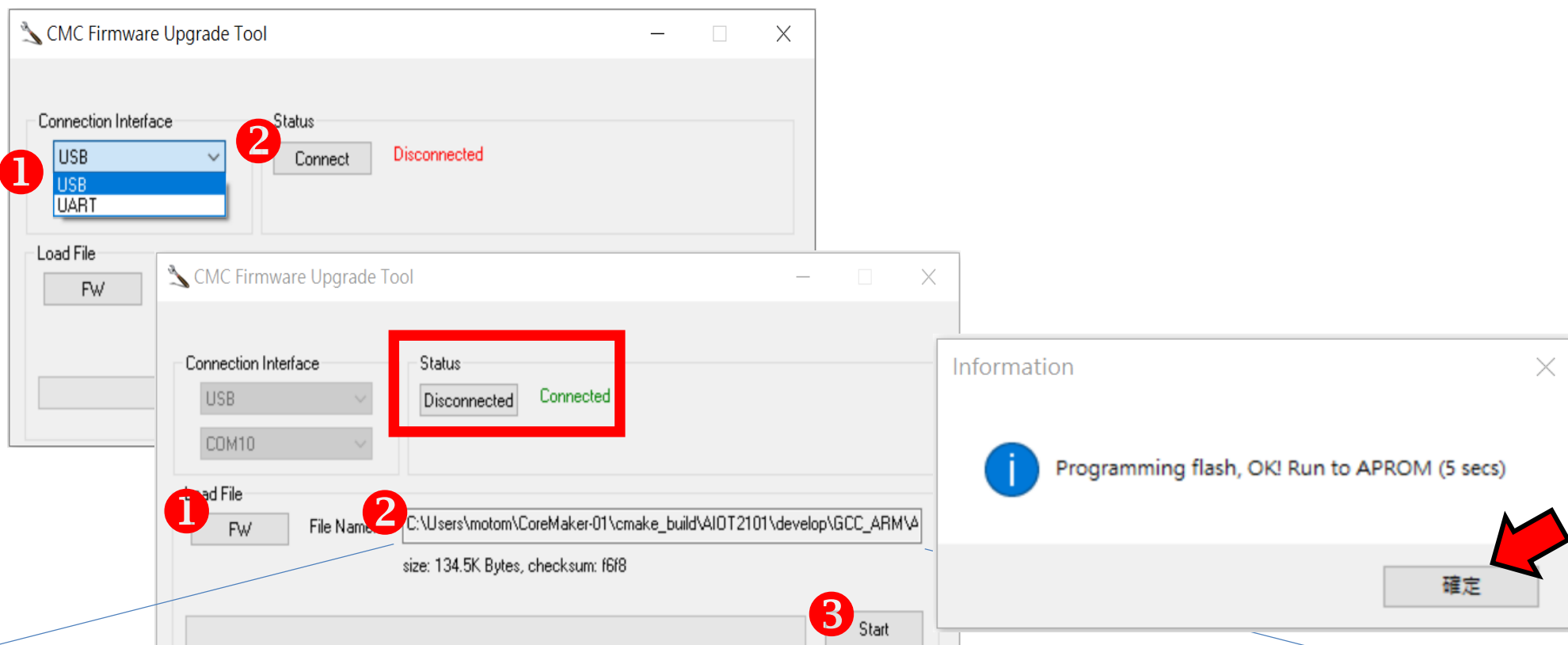
燒錄模式



四、韌體設定與修改 VI 韌體燒錄

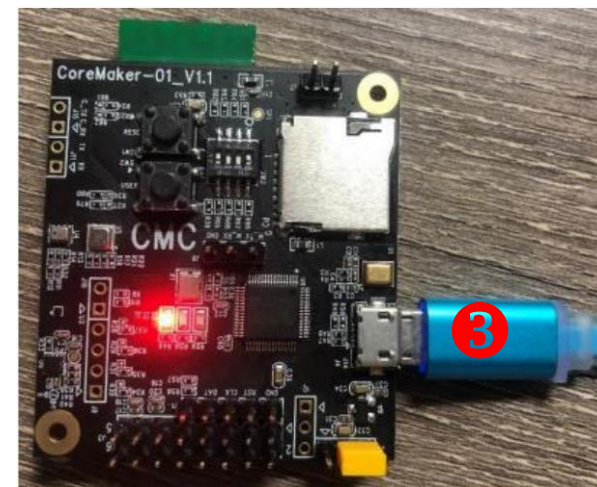
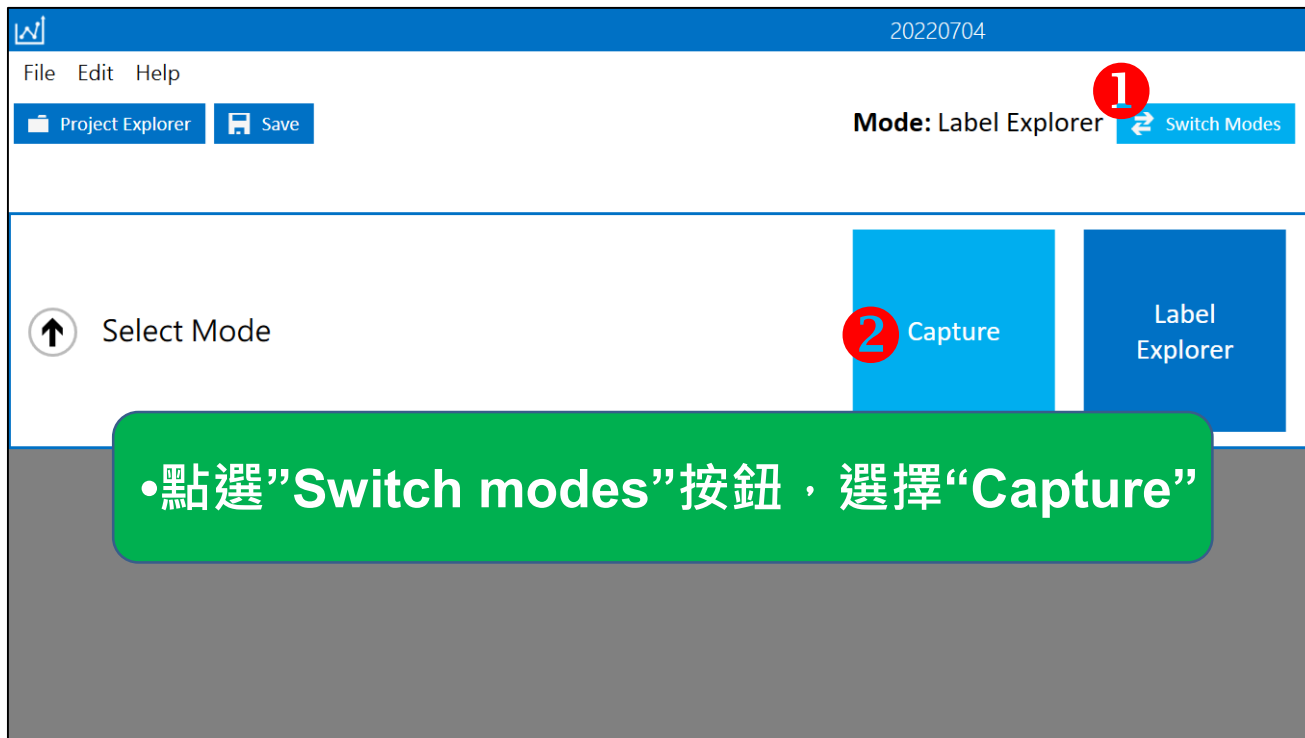


• 執行解壓縮後的 CMC_ISP.exe 並依下圖的步驟開始燒錄韌體



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

五、範本錄製與取樣 I 連接 CoreMaker I



•使用 MicroUSB to USB 的線，
連接 PC 與 CoreMaker



五、範本錄製與取樣 II

連接CoreMaker II



File Edit Help

Project Explorer

Mode: Capture Switch Modes

File Settings Live Labeling Test Model

File Name

Name

Text

File Metadata

+ Add Metadata

00:00:00:00

Start Recording

Capture Settings

Sensor Webcam Disconnected Connect

Camera Webcam Disconnected Connect

Sensor Display

•點選頁面下方 CoreMaker-01 的“Connect”按鈕

五、範本錄製與取樣 III

建立 Sensor Configuration I

- 第一次使用需要做 Device Plugin 的設定
- 跳出提示視窗後，點擊”Next”，進入選擇畫面
- 選擇名稱為”CoreMaker-01”的 device，點擊”Next”進入下一步

Sensor Configuration

Select a sensor configuration profile to use during data collection.
This will be saved as metadata to each file.

Cancel

Next

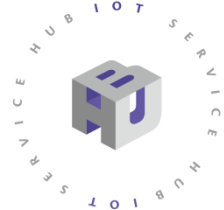
Select a Device Plugin

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
2	CoretronicMEMS	CoreMaker-01	CoretronicMEMS	Simple Streaming	Microphone, Temperature, Pressure, Humidity, Gas resistance, Accelerometer, Magnet
	Arduino	Nano33 BLE Sense	SensiML	Simple Streaming	Accelerometer, Gyroscope, Microphone
	Arduino	Nicla Sense ME	SensiML	Simple Streaming	Accelerometer, Gyroscope
	Infineon	PSoC 6 Wi-Fi BT Pioneer Kit	SensiML	Simple Streaming	Accelerometer, Gyroscope, Microphone
	Microchip Technology	SAMD21 ML Eval Kit	Microchip Technology	Simple Streaming	Accelerometer, Gyroscope
	Nordic Semiconductor	Thingy:52	SensiML	Custom	Accelerometer, Gyroscope, Audio
	onsemi	RSL10 Sense	onsemi	Simple Streaming	Accelerometer, Gyroscope
	QuickLogic	Chilkat	QuickLogic	Custom	Accelerometer, Gyroscope
	QuickLogic	QuickAI	QuickLogic	Custom	Accelerometer, Gyroscope, Microphone, Channel 1, Channel 2, Channel 3, Channel 4
	QuickLogic	QuickAI	QuickLogic	MQTT-SN	Accelerometer, Gyroscope, Microphone, Mayhew ADC
	QuickLogic	QuickFeather	QuickLogic	MQTT-SN	Accelerometer, Microphone
	QuickLogic	QuickFeather	QuickLogic	Simple Streaming	Accelerometer, Microphone
	Silicon Labs	Thunderboard Sense 2	Silicon Labs	Simple Streaming	Accelerometer, Gyroscope, Microphone

Cancel

Next



五、範本錄製與取樣 IV

建立 Sensor Configuration II

- 因本例使用麥克風辨識，Capture Source 選擇”Audio”，Sample Rate 選擇”16000”，Selected Sensors勾選”Microphone”。
- **Sample Rate 需選擇與 CoreMaker 韌體相同的設定。**

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
Name: Motion
Sensors: Accelerometer
Sample Rates: 12800, 6400, 3200, 1600, 800, 400, 200, 100, 50, 25

Back

Next

Sensor Properties

Device Plugin

CoreMaker-01

Capture Source

1 Audio

Sample Rate

2 16000

Selected Sensors

☒ Microphone

Back

3 Next

Save Sensor Configuration

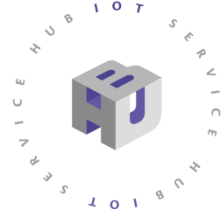
Enter a profile name for your sensor configuration. This will be saved as metadata to any files that are collected using the selected configuration.

Enter profile name

Back

Save

- 輸入自訂的設定檔名稱來儲存



五、範本錄製與取樣 V

連線Data Capture Lab

20220704.dclproj

File Edit Help

Project Explorer

Mode: Capture Switch Modes

Connection Settings

Capture Method
Live Stream Capture

Connection Method
Serial Port

COM10

File Settings Live Labeling Test Model

File Name
Name
Text

File Metadata
+ Add Metadata

- 進入 **Connection Settings** 頁面，點擊“Scan”後，會出現偵測到有連接裝置的連接埠，選擇連接埠後點擊“Done”完成設定。
- 再次點擊“Connect”，當狀態列出現“Retrieving Configuration”時，按壓 **CoreMaker** 上的 **SW2** 按鍵即可完成連線。

1 Scan Done 3 Cancel

Sensor Display

Connect ...

Disconnected

Connect ...

00:00:00:00

Start Recording

Capture Settings

五、範本錄製與取樣 VI

錄製聲音資料



The screenshot shows the CoreMaker-01 software interface. The main window displays a waveform of recorded audio. On the left, there is a small inset image of a person speaking into a microphone, with a red circle and the number '3' next to it. The waveform itself has a red circle and the number '3' next to it. The right sidebar contains settings for the recording, including File Name, Text, File Metadata, and a 'Start Recording' button. The bottom status bar shows the recording status and a 'Stop Recording' button.

20220704.dclproj

File Edit Help

Project Explorer

Mode: Capture Switch Modes

CoreMaker-01

File Settings Live Labeling Test Model

File Name

Name Audio-1F.wav

Text 1 Audio-1F

File Metadata

Connection COM10

Device CoreMaker-01

+ Add Metadata

Microphone

Sensor Display

CoreMaker-01 COM10 Recording Disconnect

Camera Webcam Disconnected Connect

00:00:00:21

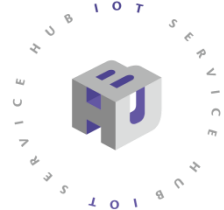
2 Stop Recording

Capture Settings

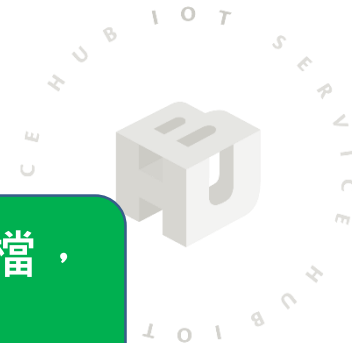
• 點擊視窗右下角的“Start Recording”按鈕開始錄製聲音。

注意

- 1.錄製聲音時請盡量靠近CoreMaker-01電路板
- 2.錄音檔時間依作者經驗建議：“至少”須超過5分30秒或取樣200次以上，如此可增加訓練模型時的參考數據，並提升模型的辨識能力



五、範本錄製與取樣 VII 上傳聲音資料



- 因為範例僅提供5層電梯按鈕，因此需錄製6個聲音檔，包括1、2、3、4、5，以及一個安靜的錄音檔。

Save Confirmation

File Name

Name Audio-1F.wav

Text Audio-1F

File Metadata

Connection COM10

Device CoreMaker-01

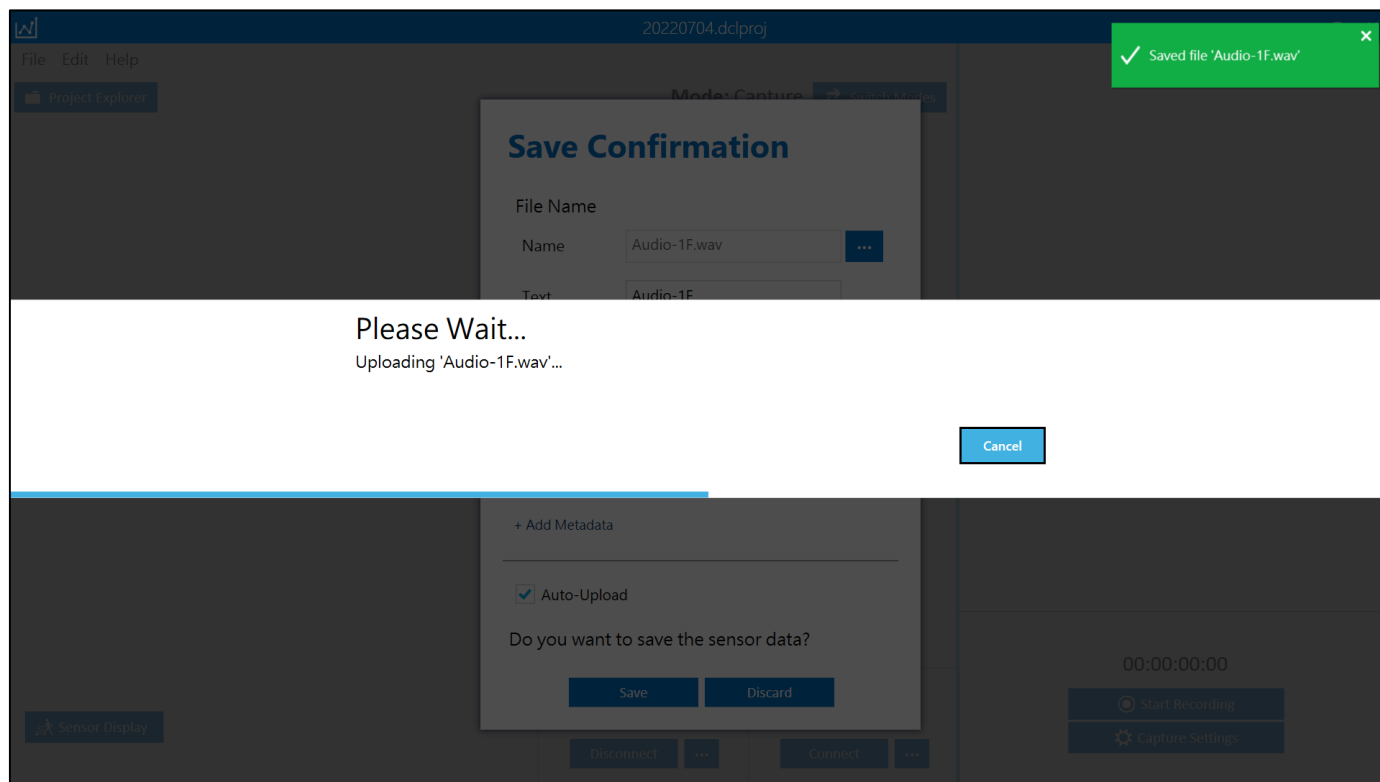
+ Add Metadata

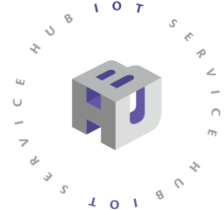
☒ Auto-Upload

Do you want to save the sensor data?

Save

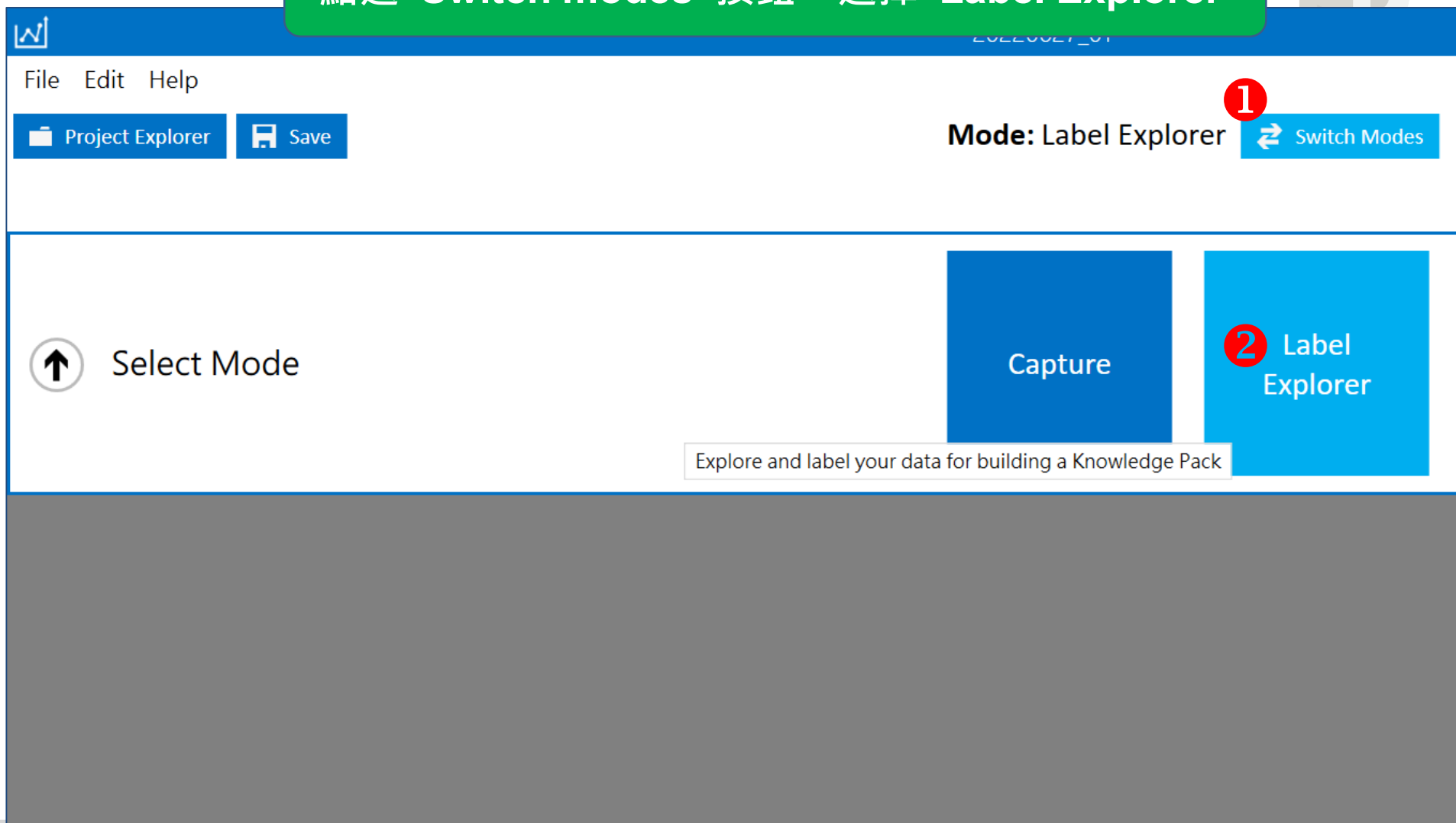
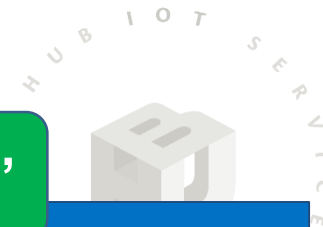
Discard



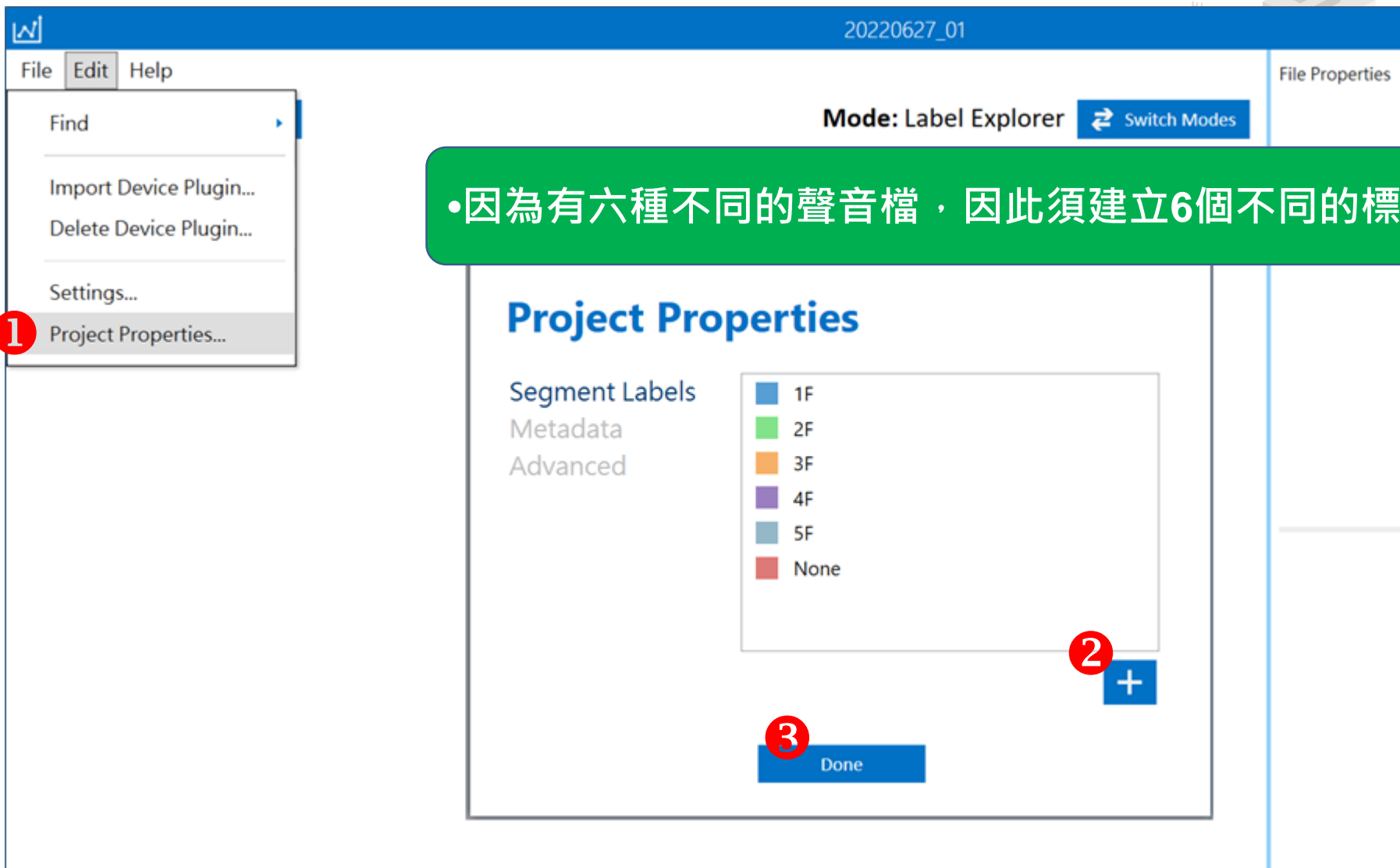


五、範本錄製與取樣 VIII 切換成標記模式

• 點選“Switch modes”按鈕，選擇“Label Explorer”



五、範本錄製與取樣 IX 建立標籤



20220627_01

File Edit Help

Find

Import Device Plugin...

Delete Device Plugin...

Settings...

1 Project Properties...

Mode: Label Explorer Switch Modes

File Properties

因為有六種不同的聲音檔，因此須建立6個不同的標籤

Project Properties

Segment Labels

Metadata

Advanced

1F

2F

3F

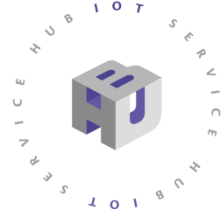
4F

5F

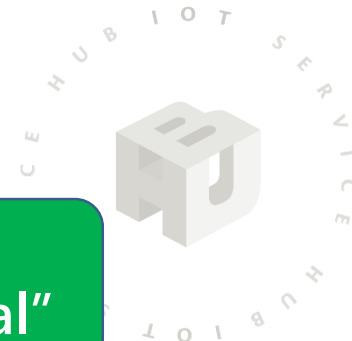
None

2 +

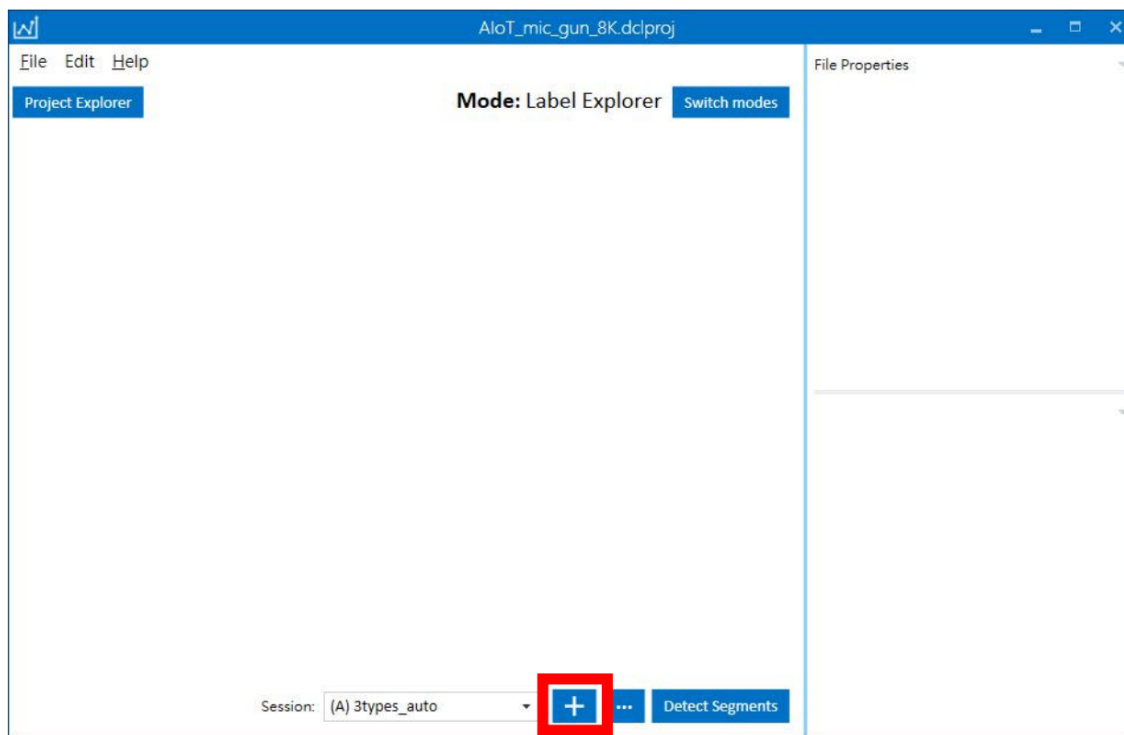
3 Done



五、範本錄製與取樣 X 建立手動標記 Session



- 點擊視窗下方的 “+” 新增 session
- 在 Name 欄位輸入 session 名稱，Type 選擇 “Manual”



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

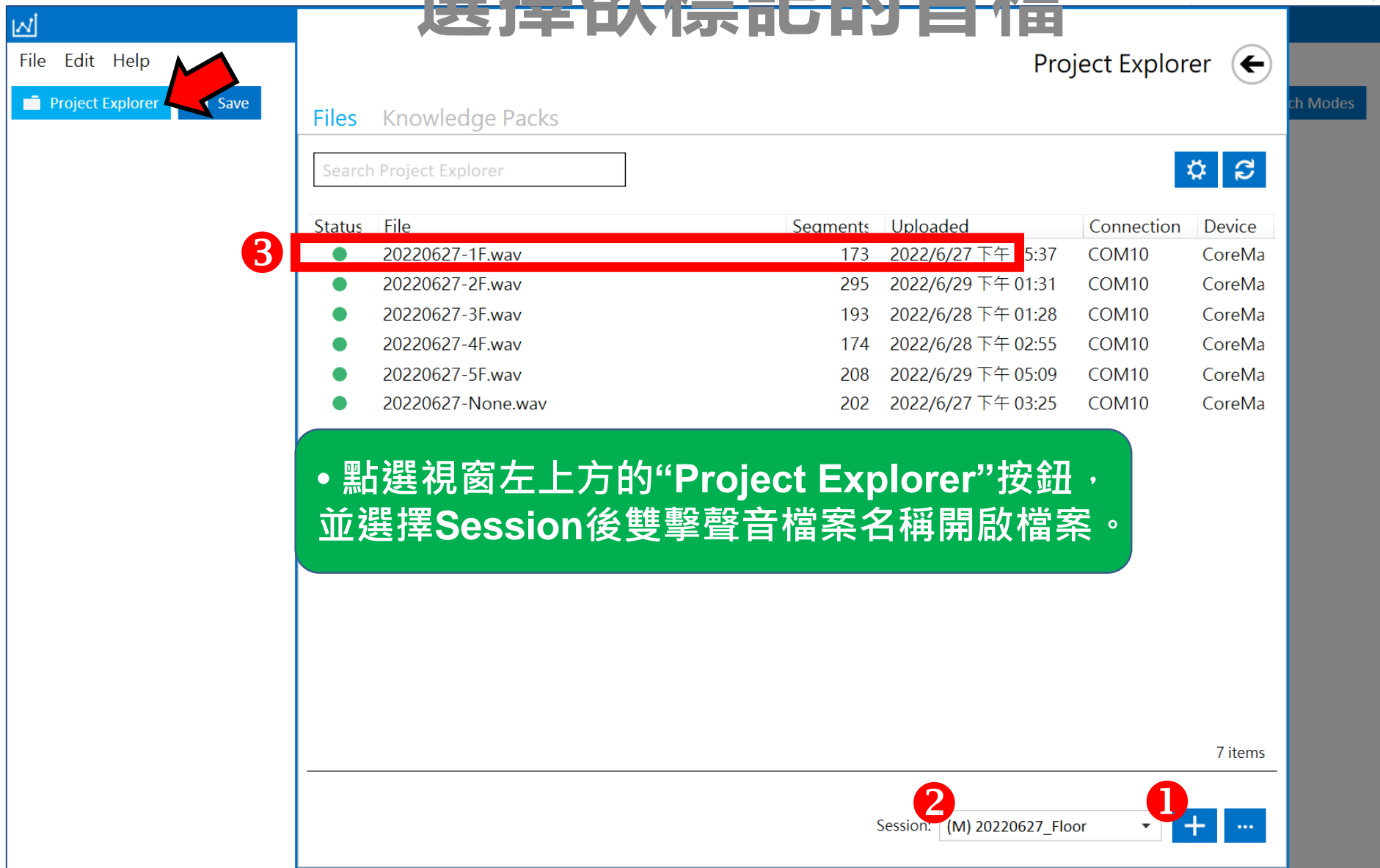
Type

☒ Manual ☐ Auto

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

五、範本錄製與取樣 XI

選擇欲標記的音檔



File Edit Help

Project Explorer Save

Project Explorer

Files Knowledge Packs

Search Project Explorer

Status	File	Segments	Uploaded	Connection	Device
●	20220627-1F.wav	173	2022/6/27 下午 5:37	COM10	CoreMa
●	20220627-2F.wav	295	2022/6/29 下午 01:31	COM10	CoreMa
●	20220627-3F.wav	193	2022/6/28 下午 01:28	COM10	CoreMa
●	20220627-4F.wav	174	2022/6/28 下午 02:55	COM10	CoreMa
●	20220627-5F.wav	208	2022/6/29 下午 05:09	COM10	CoreMa
●	20220627-None.wav	202	2022/6/27 下午 03:25	COM10	CoreMa

7 items

Session: (M) 20220627_Floor

1 2

• 點選視窗左上方的“Project Explorer”按鈕，並選擇Session後雙擊聲音檔案名稱開啟檔案。

五、範本錄製與取樣 XII

標記資料

- 在波形圖上點擊**滑鼠右鍵**，此時會出現兩條重疊的藍色長線，以**滑鼠左鍵**拖動這兩條線來設定標記範圍
- 點選**編輯按鈕**編輯標記
- 標記完成後，點選”Save Changes”按鈕儲存



File Edit **3** Save (Ctrl + S) Save

Project Explorer Mode: Label Explorer Switch Modes

Session: 20220627_Floor 1 - 1F 35168 - 47863 (12696)

Previous Next

1

2

Id	Label	Start	Length	St
1	1F	35168	12696	
2	1F	67343	13787	
3	1F	100682	13873	
4	1F	133055	15524	
5	1F	167099	14048	
6	1F	198840	15210	

1 of 173 items selected

Media Player

Video: Add Remove

00:00:00:00 00:05:34:36

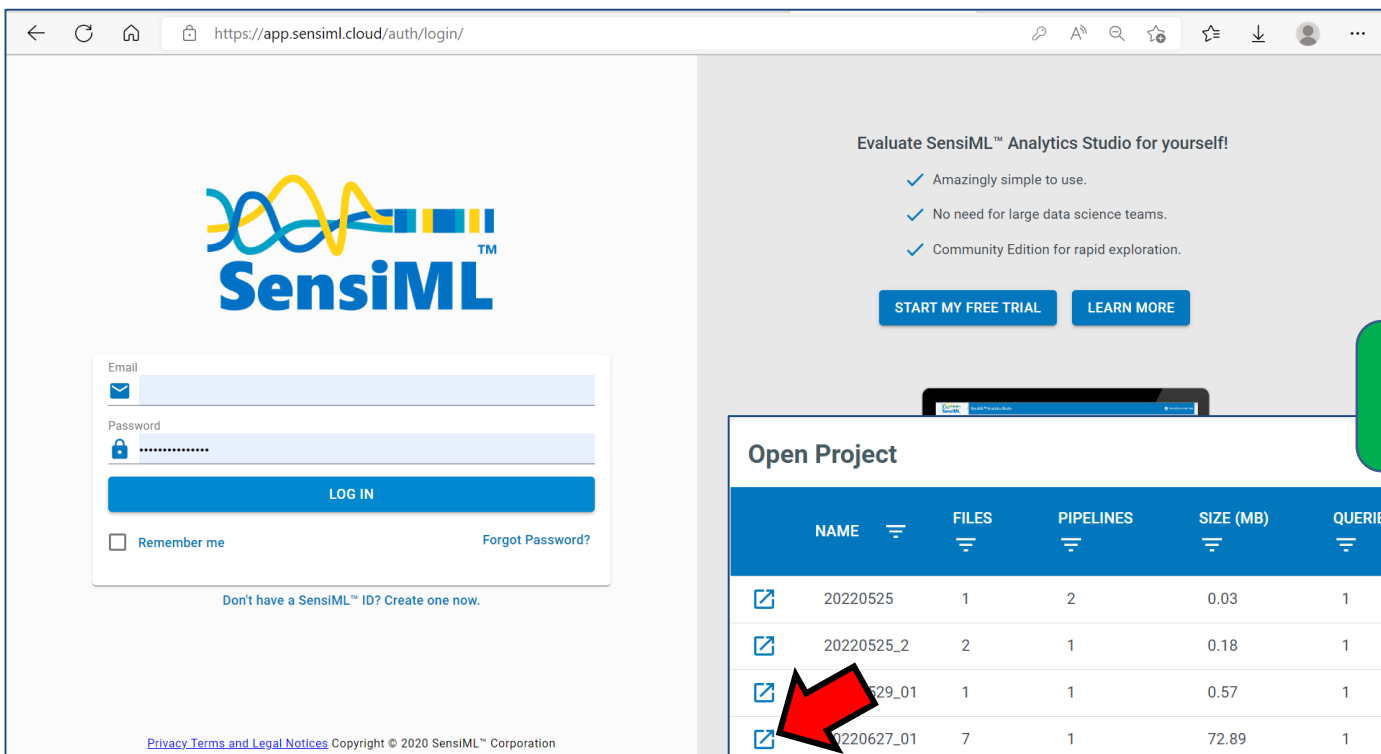
Session: (M) 20220627_Floor + ... Detect Segments

Save Audio to CSV

- 六個音檔須設定成同一個**Session**並標記各自的資料

六、AI模型製作 I 登入訓練網頁

- 進入 SensiML Analytics Studio 網頁 (<https://app.sensiml.cloud/auth/login/>)
- 登入帳號



https://app.sensiml.cloud/auth/login/

Evaluate SensiML™ Analytics Studio for yourself!

- ✓ Amazingly simple to use.
- ✓ No need for large data science teams.
- ✓ Community Edition for rapid exploration.

[START MY FREE TRIAL](#) [LEARN MORE](#)

Email:

Password:

[LOG IN](#)

☐ Remember me [Forgot Password?](#)

[Don't have a SensiML™ ID? Create one now.](#)

[Privacy Terms and Legal Notices](#) Copyright © 2020 SensiML™ Corporation

- 選擇要訓練的模型專案

Open Project

	NAME	FILES	PIPELINES	SIZE (MB)	QUERIES	MODELS	SEGMENTS	CREATED DATE	DELETE
<input checked="" type="checkbox"/>	20220525	1	2	0.03	1	5	3 of 2500	2022/5/25	
<input checked="" type="checkbox"/>	20220525_2	2	1	0.18	1	5	10 of 2500	2022/5/25	
<input checked="" type="checkbox"/>	20220529_01	1	1	0.57	1	5	9 of 2500	2022/5/29	
<input checked="" type="checkbox"/>	20220627_01	7	1	72.89	1	5	1462 of 2500	2022/6/27	
<input checked="" type="checkbox"/>	20220704	1	0	1.29	0	0	0 of 2500	2022/7/4	
<input checked="" type="checkbox"/>	TestProject	0	0	0	0	0	0 of 2500	2022/5/24	

Rows per page: 10 1-6 of 6



六、AI模型製作 II

建立模型 1



- 在 Prepare Data 頁面中建立新的 Query
 - 輸入 Query 名稱
 - 選擇使用的 Session
 - 選擇使用的 Label
 - 選擇完成點擊”Save”儲存

1 Prepare Data

Project: 20220627_01

Please enter a name for the new query

2

Query
20220627_Query

Session
20220627_Floor

Label

Metadata
segment_uuid

Source
channel_0

Query Filter

3 SAVE CANCEL

ADD NEW QUERY

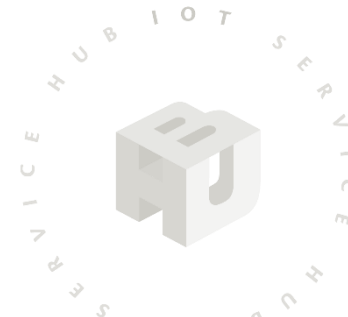
No Data to Display

No Data to Display



六、AI模型製作 III

建立模型 2



- 在 Build Model 頁面中建立新的 Pipeline

The screenshot displays the SensiML web interface. On the left is a sidebar with navigation links: Project Summary, Prepare Data, Build Model (highlighted with a red box and a red circle with the number 1), Explore Model, Test Model, Download Model, Get Started, Demos, Documentation, and Support. The main content area is titled 'Build Model' and shows a 'Create New Pipeline' section. This section includes a description of pipelines, a list of capabilities (selecting components and creating caches), and a blue button labeled 'CREATE NEW PIPELINE' (pointed to by a red arrow and a red circle with the number 2). A modal window titled 'Create New Pipeline' is open, showing a two-step process: 1. Select parameters and 2. Create pipeline. It contains an information box about AutoML and a toggle switch for 'Use SensiML AutoML to find the best machine learning algorithm'. At the bottom of the modal is a blue button labeled 'NEXT' (marked with a red circle and the number 3).


1 Build Model


2 CREATE NEW PIPELINE

3 NEXT

六、AI模型製作 IV

建立模型 3





Select parameters

Create pipeline

New name

20220627_Pipe

Query

20220627_Query

☒ Use session preprocessor

Name

20220627_Query

Label Column

Label

Columns

channel_0


Metadata Columns

segment_uuid

Session

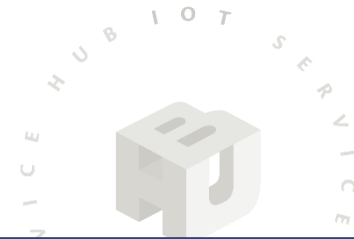
20220627_Floor

Cache Status

 NOT BUILT

CREATE PIPELINE

- 建立新的 Pipeline
 - 輸入 Pipeline 名稱
 - 選擇使用的 Query
 - 選擇完成點擊”CREATE PIPELINE”



2022 © 資訊工業策進會 Institute for Information Industry



六、AI模型製作 VI

建立模型 5



- Pipeline 運行完成後，會建立 5 個模型
- 差別在於使用的分類器空間大小及特徵數量，使用者可根據需求選擇下載合適的模型
- 不同的分類器大小及特徵數量會影響模型的準確度

Project: 20220627_01

Build Model

Explore Model

Test Model

Download Model

Get Started

Demos

Documentation

Support

Name: 20220627_Query
Type: Input Query

Name: Windowing
Type: Segmenter

Type: AutoML Parameters

Classifier SRAM 32000
f1-score 100

OPTIMIZE

RUNNING

✓ Optimizing Query...

✓ Query is optimized...

✓ Submitting Automation Pipeline Run.


✓ Status: Running, Time: 0.06, STEP: 0/3, NAME: 20220627_Query, TYPE: query,

✓ Status: Running, Time: 0.11, STEP: 0/3, NAME: 20220627_Query, TYPE: query,

✓ Status: Running, Time: 0.16, STEP: 0/3, NAME: 20220627_Query, TYPE: query,

✓ Status: Running, Time: 0.21, STEP: 0/3, NAME: 20220627_Query, TYPE: query,

RESULT

LOGS 

TRAINING SUMMARY

AutoML Results

MODEL NAME	ACCURACY	CLASSIFIER SIZE(B)	NUM. FEATURES	SENSITIVITY
20220627_Pipe_rank_0	98	11900	15	98
20220627_Pipe_rank_1	98	13424	17	98
20220627_Pipe_rank_2	97	11444	16	97
20220627_Pipe_rank_3	98	4268	28	97
20220627_Pipe_rank_4	96	1674	16	95

- 開始訓練，訓練完成會出現如右圖所示結果



六、AI模型製作 VII 模型測試



- 訓練完成後可先在網頁上進行辨識度測試 (非必要步驟)

Project: 20220627

Session
20220627_Floor

Project Summary

Prepare Data

Build Model

Explore Model

Test Model

Download Model

Get Started

Demos

Documentation

Support

Evaluate your model by selecting one or more capture files for classification.

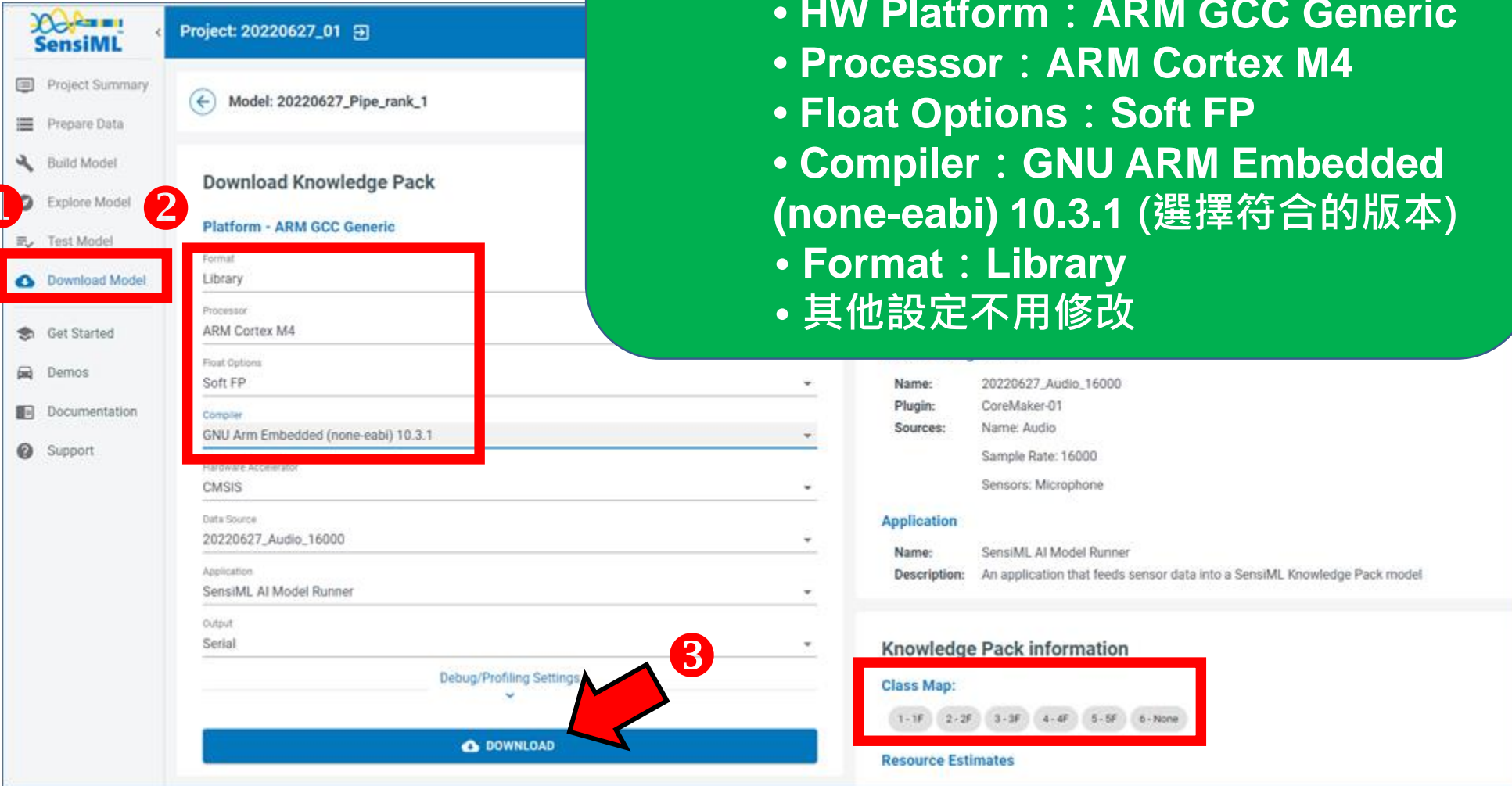
Captures

...	CAPTURE NAME	ACCURACY	TOTAL EVENT COUNT	CAPTURE UUID	RESULTS	SIZE (MB)	CONNECTION	DEVICE
<input checked="" type="checkbox"/>	20220627-1F.wav	89.5	173	af336d7c-0f82-4d4c-873b-c016ab7a3e45	RESULTS	10.20	COM10	CoreMaker-01
<input checked="" type="checkbox"/>	20220627-2F.wav	56.2	295	83981681-ead6-4a38-8e77-9fafc39d76f7	RESULTS	10.80	COM10	CoreMaker-01
<input checked="" type="checkbox"/>	20220627-4F.wav	83.9	174	ee7cbdb2-8975-4628-aa44-0a4f125f35a3	RESULTS	10.72	COM10	CoreMaker-01
<input checked="" type="checkbox"/>	20220627-None.wav	100	202	11429d09-63b2-402e-b073-abbc17ad013c	RESULTS	10.17	COM10	CoreMaker-01
<input checked="" type="checkbox"/>	20220627-5F.wav	94.8	208	261234fd-c8e9-4e8a-b45e-fab96a5f4046	RESULTS	10.23	COM10	CoreMaker-01
<input checked="" type="checkbox"/>	20220627-3F.wav	79.7	193	f3dda026-89ef-4651-ba88-cb88ad404f36	RESULTS	10.22	COM10	CoreMaker-01

六、AI模型製作 VIII

模型下載

- 在 Download Model 頁面中，選擇以下設定
 - HW Platform : ARM GCC Generic
 - Processor : ARM Cortex M4
 - Float Options : Soft FP
 - Compiler : GNU ARM Embedded (none-eabi) 10.3.1 (選擇符合的版本)
 - Format : Library
 - 其他設定不用修改



1 Download Model

2 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

3 DOWNLOAD

Knowledge Pack information

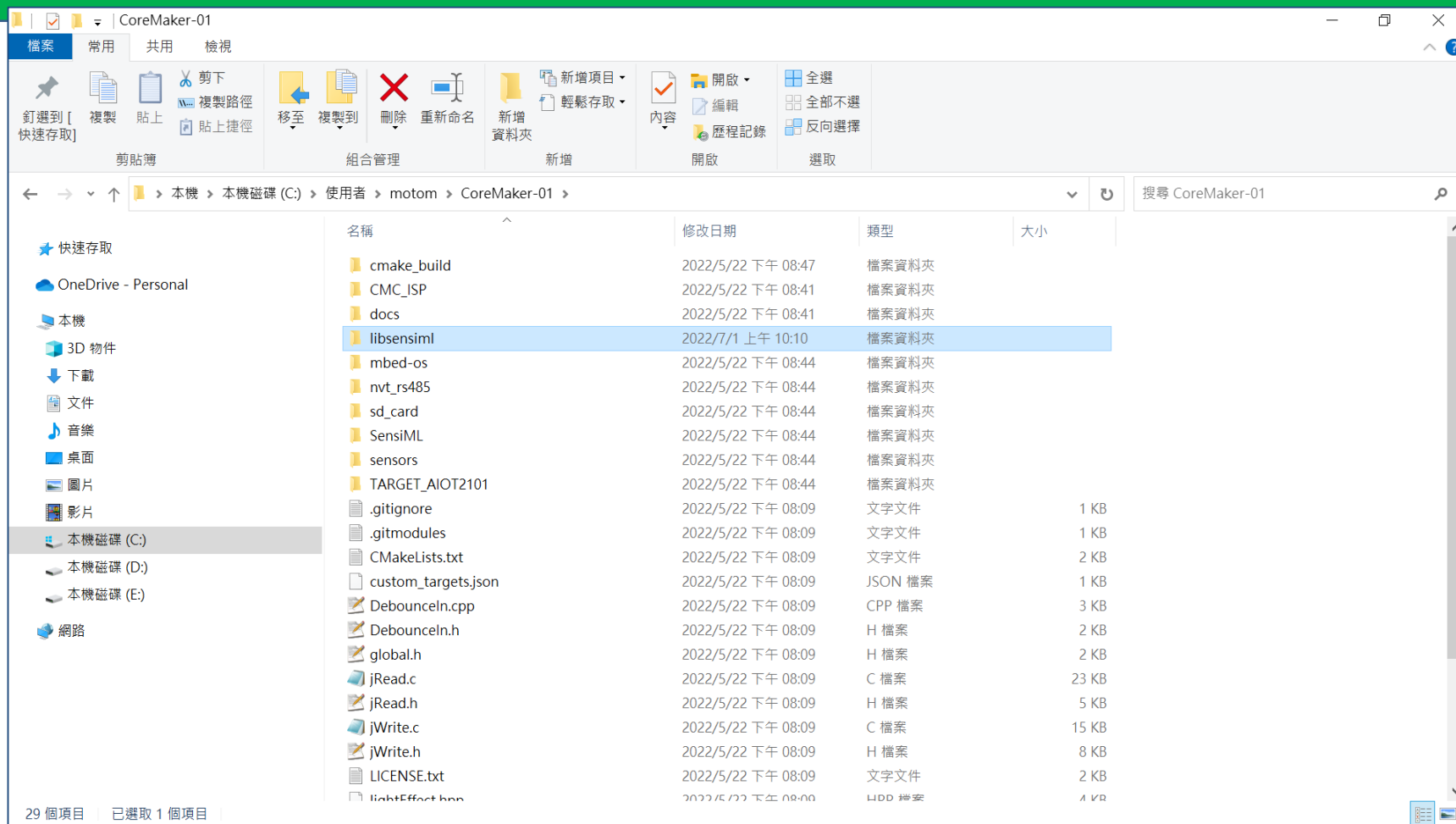
Class Map:

1 - 1F 2 - 2F 3 - 3F 4 - 4F 5 - 5F 6 - None

六、AI模型製作 IX

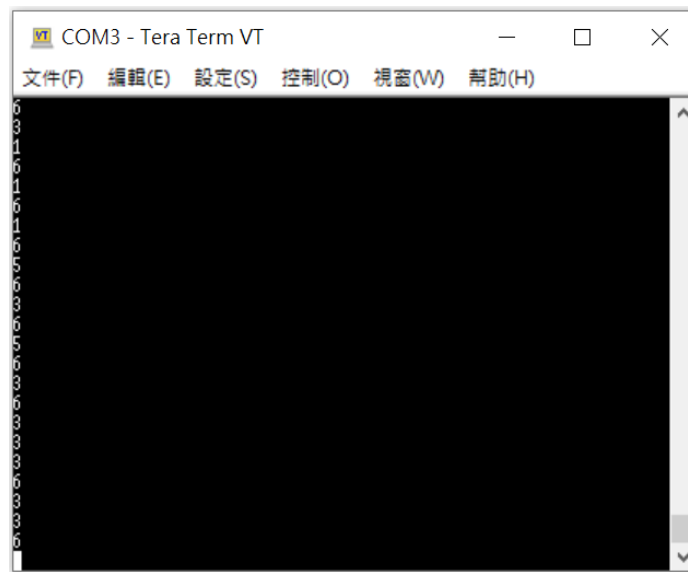
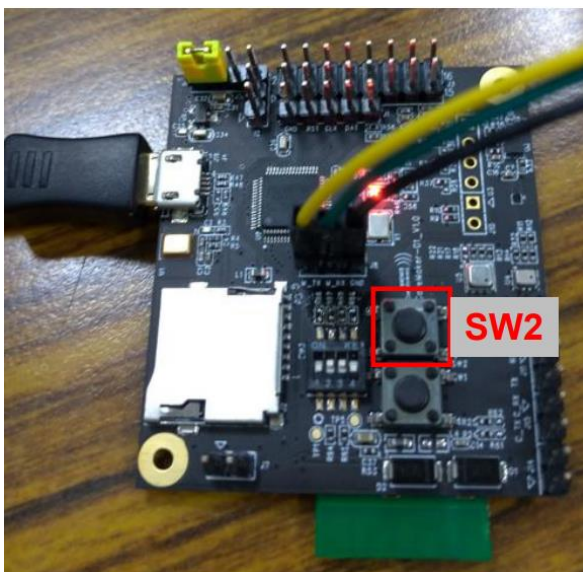
模型編譯與燒錄

- 解壓縮下載的檔案，用裡面的 libsensiml 資料夾覆蓋 CoreMaker-01 下的 libsensiml 資料夾
- 進行重新編譯韌體及燒錄



六、AI模型製作 X 模型驗證

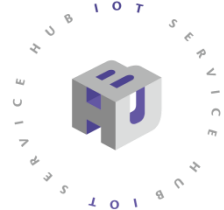
- 燒錄完成後，按下 CoreMaker 上的 SW2 鍵，可以透過 UART 看到 AI 辨識的分類結果，分類結果的數字代表意義，可在模型下載頁面中查看



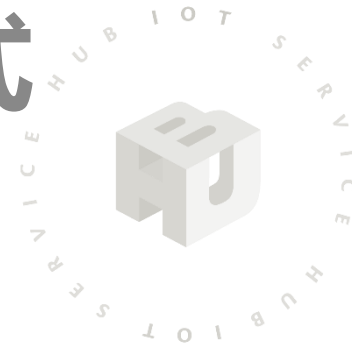
Knowledge Pack information		
Class Map:		
1 - 1F	2 - 2F	3 - 3F
4 - 4F	5 - 5F	6 - None
Resource Estimates		
Estimated Memory Usage		
SRAM Used:	32358 Bytes	?
Stack Size:	1200 Bytes	?
Flash Used:	29016 Bytes	?
Estimated Latency		
Feature Extraction Latency:	517.100 ms (41368000)	?
Total Latency:	517.100 ms (41368000)	?

****注意****

模型跟訓練者聲紋有關，故訓練完成後若非訓練者本人使用可能會影響辨識效果。若要建立通用模型，需多取樣不同人的聲音以提高辨識率。



七、Arduino程式編寫 I 變數宣告與Setup()函式



```
1 // 與HMI觸控面板溝通的變數
2 #define HMI_DATA_BUFF_SIZE 64
3 char szHMIData[HMI_DATA_BUFF_SIZE];
4 int nHMIDataLen = 0;
5
6 const byte LED_PIN = 13; // 定義LED腳位
7 int nLastFloorCall = 0; // 最後呼叫的樓層
8 long lCallTimer = 0; // 呼叫樓層的時間
9
10 // 各樓層按鈕的狀態(是否已被按下)
11 #define FLOOR_COUNT 5
12 bool bBtnStatus[FLOOR_COUNT];
```

```
14 void setup()
15 {
16     int i=0;
17
18     Serial.begin(38400); //顯示除錯訊息的串列埠口
19     Serial2.begin(115200); //與DSI-6484溝通的串列埠口
20     Serial3.begin(115200); //與HMI觸控面板溝通的串列埠口
21     Serial.println("HMI LCD 初始化");
22
23     pinMode(LED_PIN, OUTPUT);
24     memset(szHMIData, 0, sizeof(char)*HMI_DATA_BUFF_SIZE);
25     //初始化各樓層按鈕的狀態
26     for (i = 0; i < FLOOR_COUNT; i++)
27         bBtnStatus[i]=false;
28
29     for (i = 0; i < 3; i++) { //整體初始化動作完成，LED閃爍三次
30         digitalWrite(LED_PIN, HIGH);
31         delay(300);
32         digitalWrite(LED_PIN, LOW);
33         delay(200);
34     }
35 }
```

七、Arduino程式編寫 II

Loop()函式

```

37 void loop()
38 {
39     int nHMIDataLen=0;
40
41     if (Serial3.available()) {
42         //取得來自DSI-6484回覆的辨識結果
43         char cData = Serial3.read();
44
45         //當呼叫樓層三秒後，自動幫使用者按下或取消電梯按鈕
46         if(lCallTimer>0 && millis()-lCallTimer>3000)
47         {
48             char tempBuff[10];
49
50             setHmiTxt("t0", "--");
51             sprintf(tempBuff, "bt%d", nLastFloorCall);
52             bBtnStatus[nLastFloorCall-1]=!bBtnStatus[nLastFloorCall-1];
53             fnSetHmiVal(tempBuff, bBtnStatus[nLastFloorCall-1]);
54             lCallTimer=0;
55             nLastFloorCall=0;
56
57             return;
58         }

```

```

59     } else if(lCallTimer>0 && millis()-lCallTimer<500)
60     {
61         return;
62     }
63
64     //當呼叫樓層數字大於1、小於5時，先將所辨識到的數字顯示在電梯螢幕上
65     if(cData >='1' && cData <= '5')
66     {
67         nLastFloorCall=atoi((const char*)&cData);
68         lCallTimer=millis();
69         Serial.println(nLastFloorCall);
70         setHmiTxt("t0", &cData);
71     }
72 }

```

```

74     //取得來自HMI觸控面板的串列埠封包
75     nHMIDataLen=handleHmiSerial(&Serial2);
76     if(nHMIDataLen>0)
77     {
78         Serial.println("Get HMI Data!");
79         if (szHMIData[0] == 0x65 && szHMIData[1] == 0x00)
80         {
81             bBtnStatus[szHMIData[2]-1]=!bBtnStatus[szHMIData[2]-1];
82         }
83     }
84 }

```

七、Arduino程式編寫 III

HMI觸控面板 1

```

87 // 接收來自 HMI 的資料
88 //
89 int handleHmiSerial(Stream* ser)
90 {
91     nHMIDataLen=0;
92     memset(szHMIData, 0, sizeof(char)*HMI_DATA_BUFF_SIZE);
93     if (ser->available()) {
94         while (ser->available()) {
95             char cc = ser->read();
96
97             if (nHMIDataLen >= HMI_DATA_BUFF_SIZE) {
98                 Serial.println("Error : out of range");
99                 nHMIDataLen = 0;
100                 return 0;
101             }
102             szHMIData[nHMIDataLen++] = cc;
103
104             if(nHMIDataLen>3)
105             {
106                 if (checkEnd(szHMIData, nHMIDataLen))
107                     return nHMIDataLen;
108             }
109             delay(200);
110         }
111     }
112     return 0;
113 }

```

```

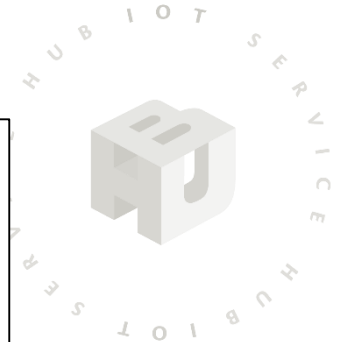
115 // 檢查來自HMI觸控面板的封包是否已經結束
116 bool checkEnd(char data[], int len)
117 {
118     if (len < 3)
119         return false;
120
121     int ei = len - 1;
122     //Serial.printf("data[%d]=%02x\n", ei, data[ei]);
123     if (data[ei--] != 0xFFFFFFFF)
124         return false;
125
126     //Serial.printf("data[%d]=%02x\n", ei, data[ei]);
127     if (data[ei--] != 0xFFFFFFFF)
128         return false;
129
130     //Serial.printf("data[%d]=%02x\n", ei, data[ei]);
131     if (data[ei] != 0xFFFFFFFF)
132         return false;
133
134     return true;
135 }

```




七、Arduino程式編寫 IV

HMI觸控面板 2

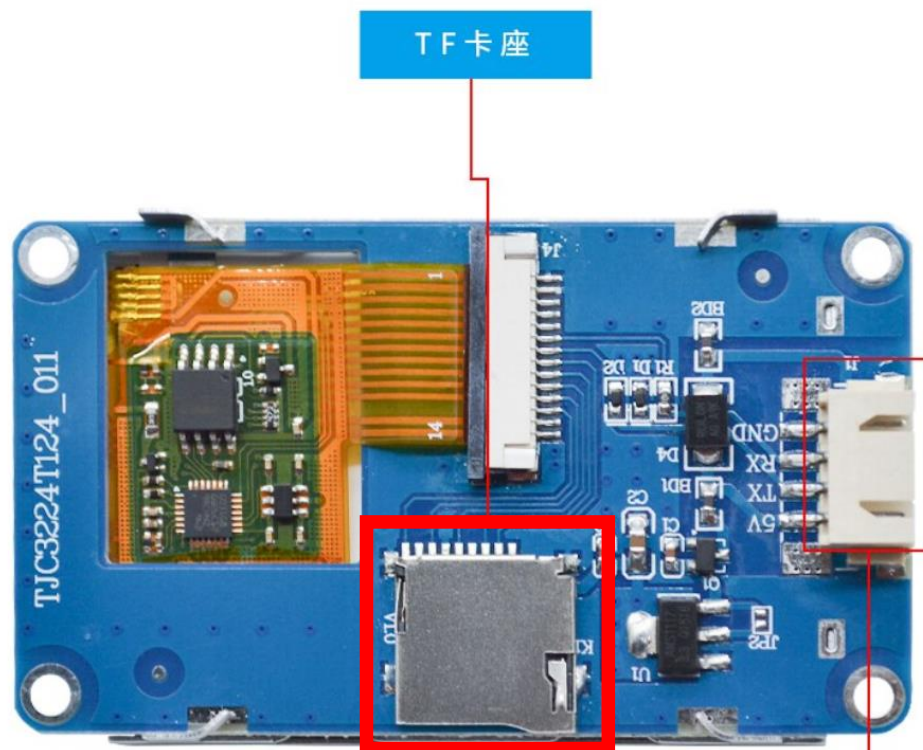


```
137 void fnSetHmiVal(const char* ctl, int val)
138 {
139     char buff[32];
140     sprintf(buff, "%s.val=%d", ctl, val);
141     Serial2.print(buff);
142     Serial2.write(0xff);
143     Serial2.write(0xff);
144     Serial2.write(0xff);
145 }
146
147 void setHmiTxt(const char* ctl, char* txt)
148 {
149     char buff[32];
150     sprintf(buff, "%s.txt=\"%s\"", ctl, txt);
151     Serial2.print(buff);
152     Serial2.write(0xff);
153     Serial2.write(0xff);
154     Serial2.write(0xff);
155 }
```

七、Arduino程式編寫 V

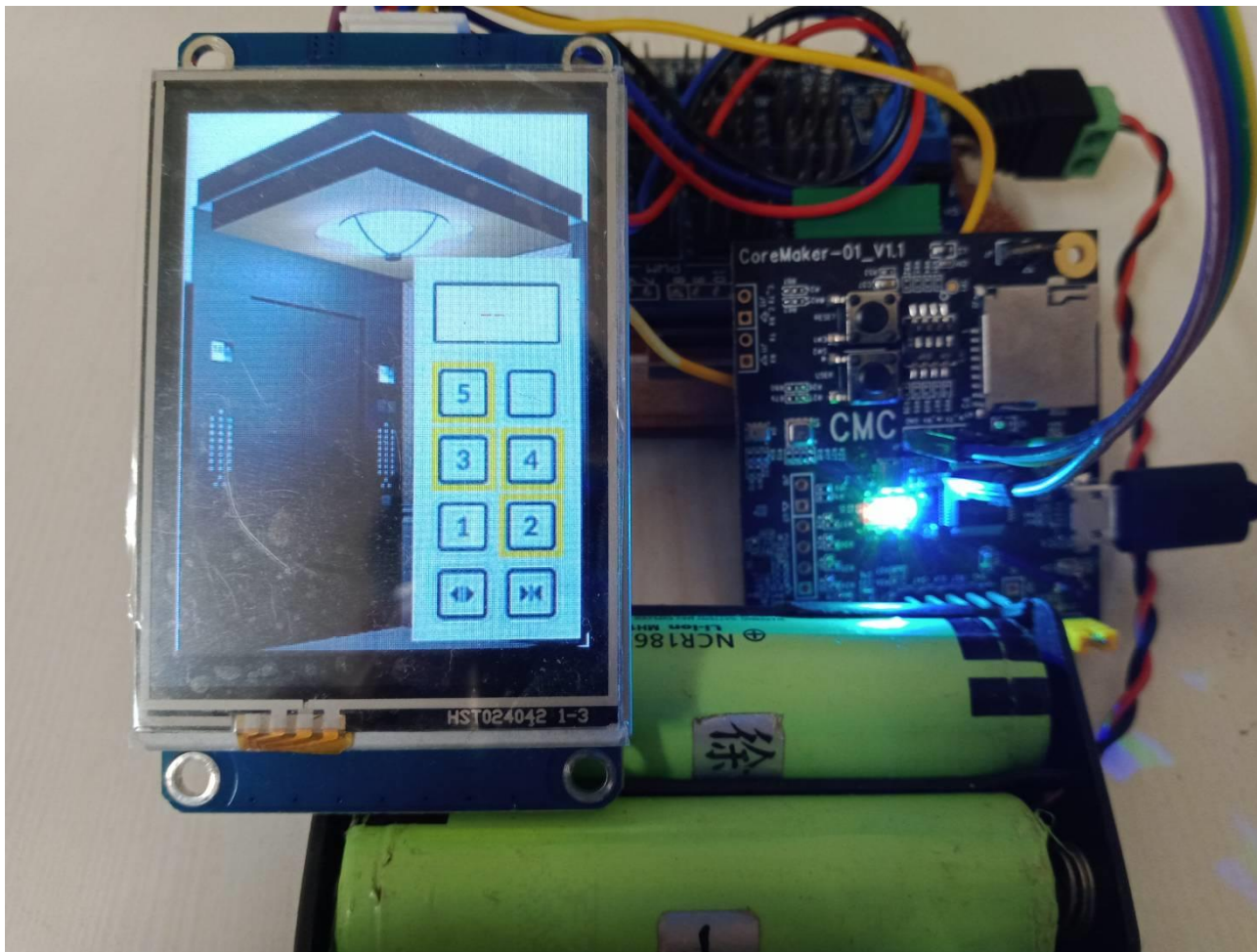
HMI觸控面板 3

- 觸控面板型號：HMI~T1系列
TJC3224T124_011R

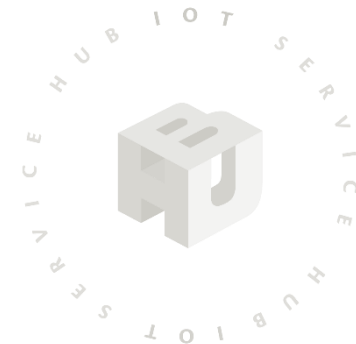
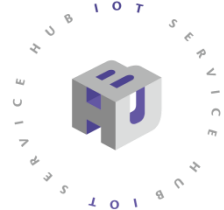


****將設計好的HMI觸控面板畫面程式(tft檔)複製到MicroSD卡中，在觸控面板通電的狀態下插入該SD卡，畫面程式便會自動寫入觸控面板中。
(此動作只需做一次即可，程式寫入完成後便可移除SD卡)**

八、成果展示



參考影片 <https://youtu.be/Eky6IRX41QQ>



Thank you

