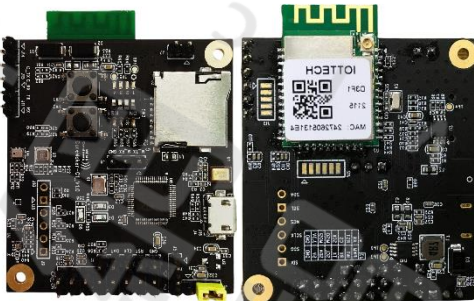


CoreMaker-01



人體姿態智能辨識

指導單位：

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

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
IIOT SERVICE HUB

合作單位：設計者：Dereck

案例簡介

- 本專案使用國產IC開發套件：CoreMaker-01開發板實現人體姿態辨識，根據板子上的三軸加速度器，偵測人不同動作下結果；其延伸可應用在各類穿戴式產品，紀錄人體的運動狀態，如跑步、爬山、游泳或慢走等等動作，提供使用者各種數據分析
- 可辨識類別為：靜止、預備姿勢、直拳和右鉤拳
- 使用邊緣運算AI優勢：
 1. 數據隱私性以及資料安全性
 2. 可離線工作, 反應即時
 3. 低功耗, 持久性佳
 4. 小且容易安裝, 成本低

開發流程圖

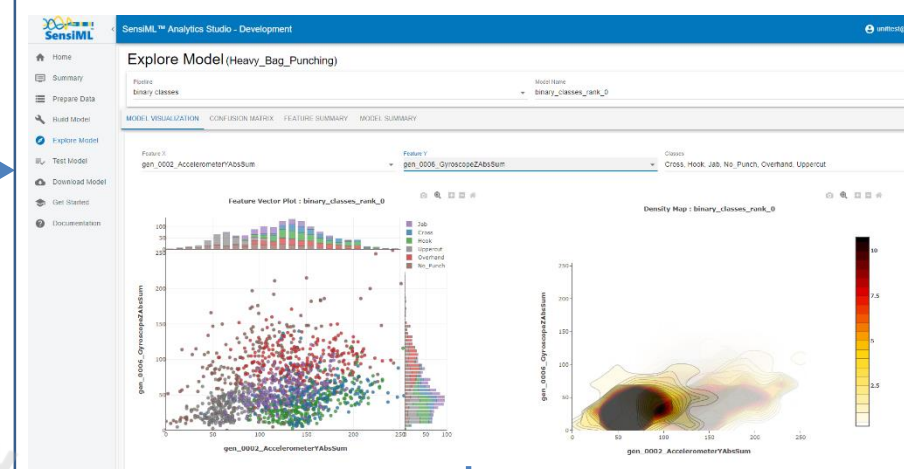
開發

手部姿態

蒐集與標註資料(Data Capture Lab)



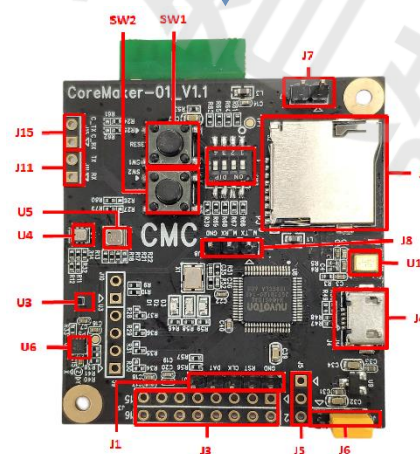
訓練與編譯模型(Analytics Studio)



應用

手部姿態

下載燒錄



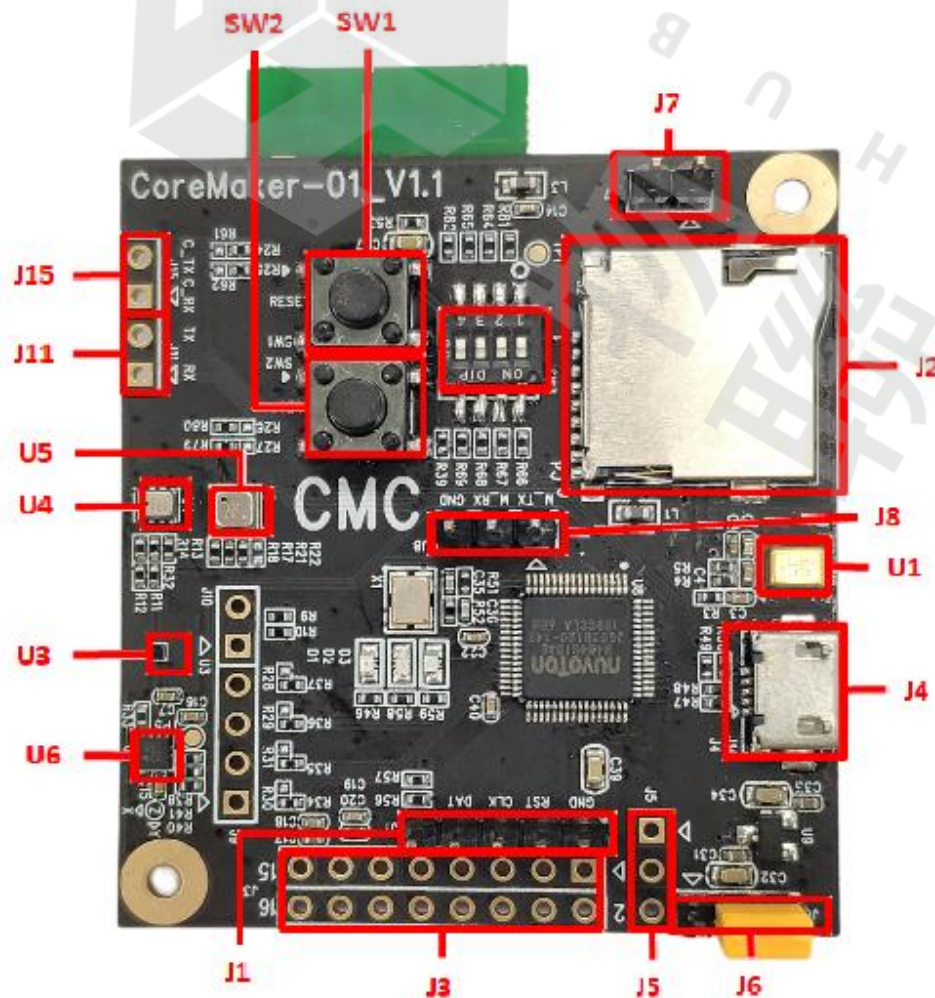
直拳

右鉤拳

預備動作

手放於大腿旁

開發環境介紹---硬體



● 主控制器M484SIDAE

- ◆ 工作主頻高達 192 MHz，512KB Flash/160 KB SRAM，內建 FPU和DSP可加速算法，1.8V到3.6V操作電壓，以及 - 40 °C 到 105 °C 的工業級工作溫度

● 板上元件

- ◆ 磁力計(GMC306A)
- ◆ 氣壓計(GMP102)
- ◆ 環境偵測(BME680)
- ◆ 加速度計(KX122-1037)
- ◆ 麥克風
- ◆ WIFI/BLE無線模組

● 支援介面

- ◆ mikro bus



開發環境建立

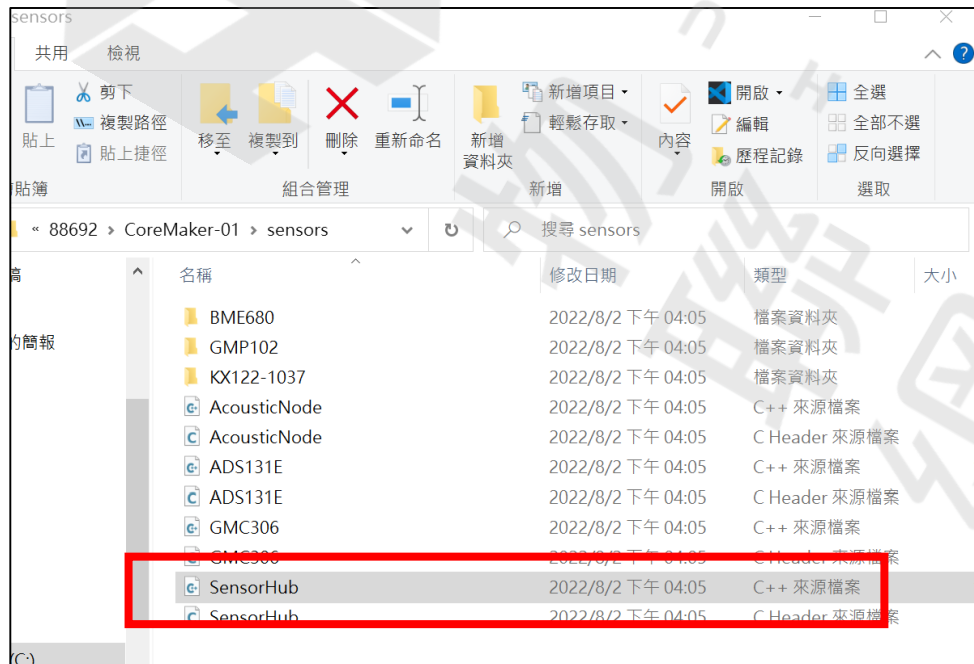
- [CoreMaker-01_V1.1 操作指南:環境設定、下載及編譯原始碼、加入AI](#)
- 範例程式
 - [WiFi 運行指南](#)
 - [SD Card 運行指南](#)
 - [LED 燈效控制範例](#)
- Sensor Datasheet
 - [KX122-103 加速度感知器](#)
 - [BME680 環境感知器](#)
 - [GMP102 氣壓感知器](#)
 - [SPU0410LR5H-QB 聲音感知器](#)
- CoreMaker-01電路圖
 - [CoreMaker-01_V1.1](#)
- 應用指南
 - [CoreMaker-01_V1.1電源供應模式](#)
 - [CoreMaker-01_V1.1 PWM控制LED和風扇](#)



資料蒐集相關設定

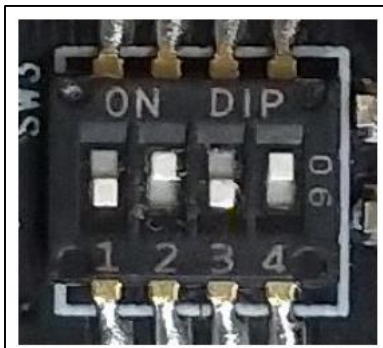
檔案位置:

CoreMaker-01/sensors/SensorHub.cpp



```

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, 2000);
57     BME680 bme680(0x76 << 1, &i2c1);
58     GMP102 gmp102(&i2c1, 256);
59     GMC306 gmc306(&i2c1, 10);
60     KX122 kx122(&spi0, PA_10, 100);
61
62
    
```

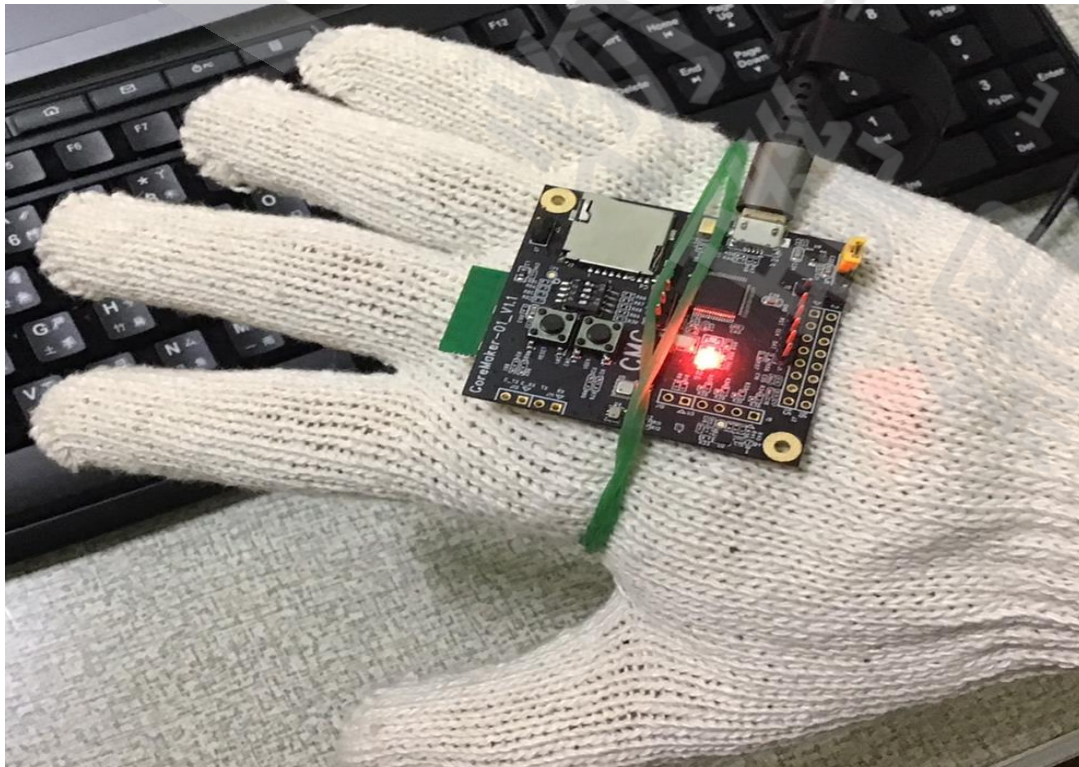


把SW3 撥鈕調整至加速
度蒐集模式

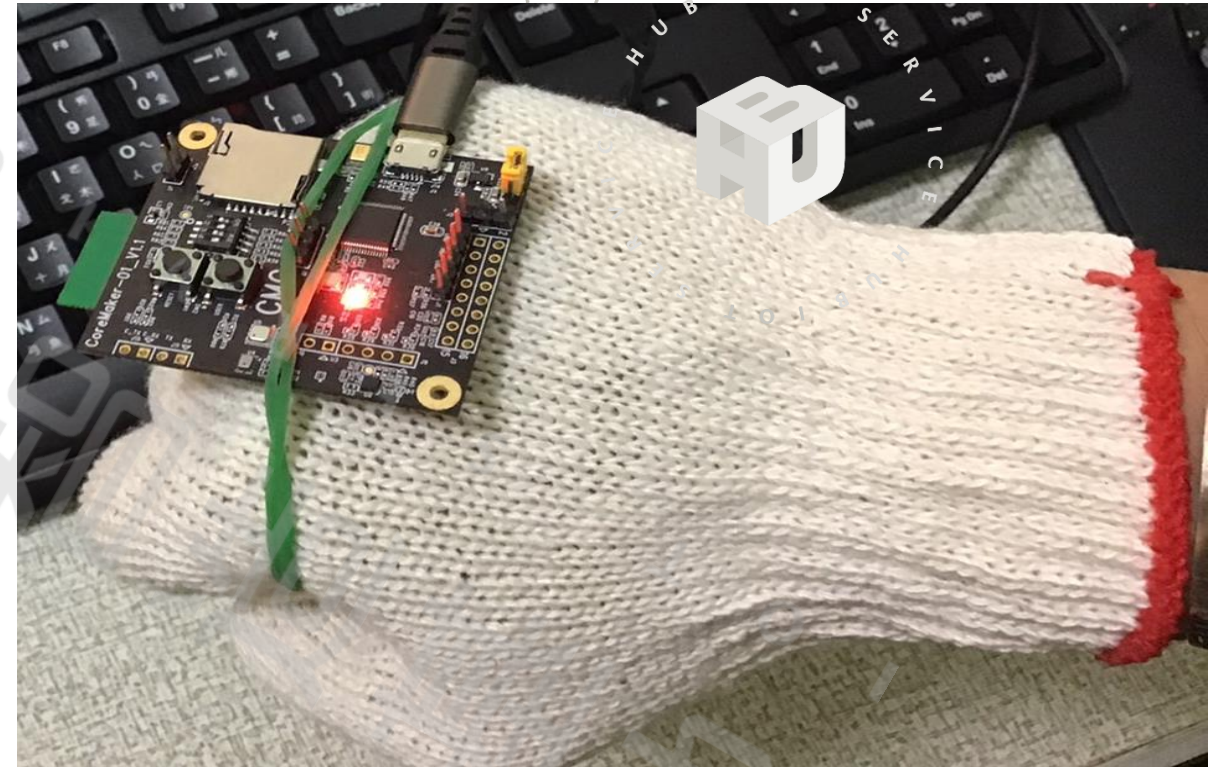
在SensorHub.cpp的60行
將聲音取樣類別調整至100Hz當作採樣頻率

板子安裝位置示意圖

CoreMaker-01放於手背，並以板上的wifi模組與中指切齊



手張開

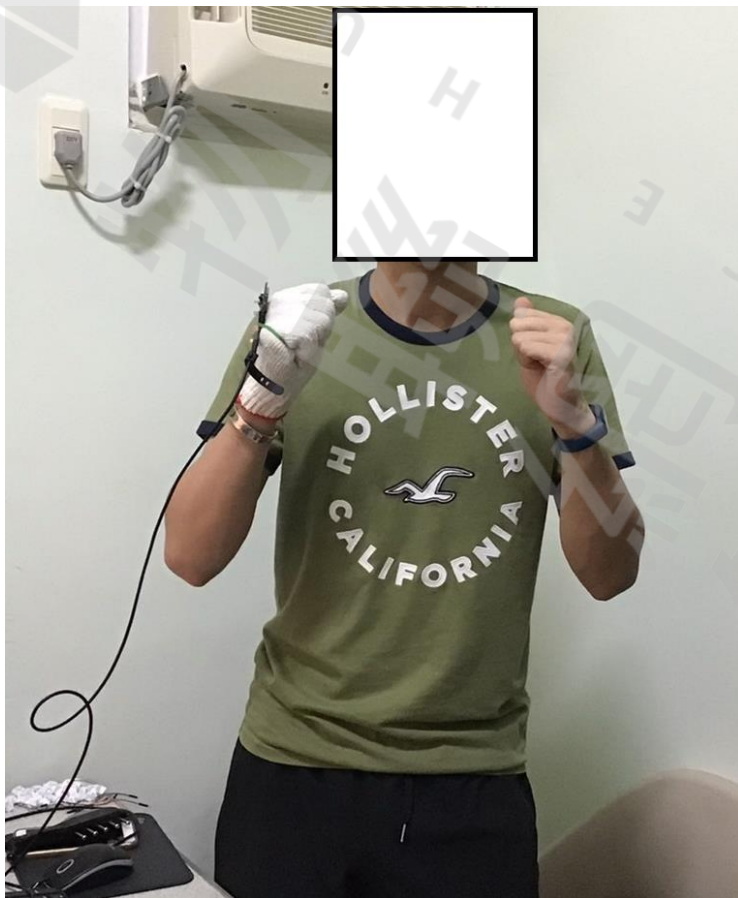


握拳

動作演示

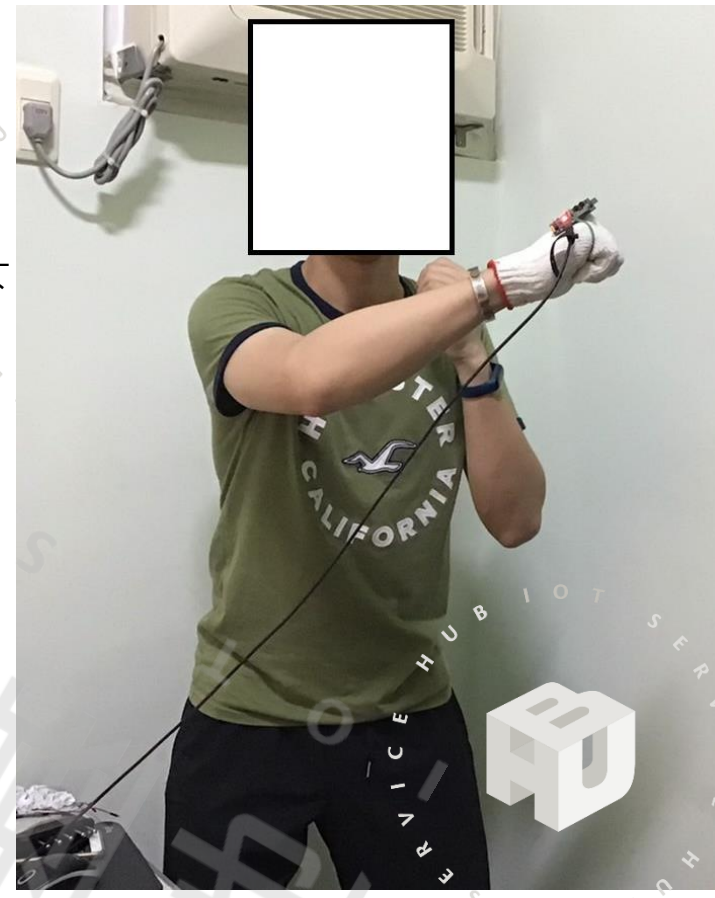
預備動作

拳頭放於兩頰以下



右鉤拳

拳頭放於兩頰以下

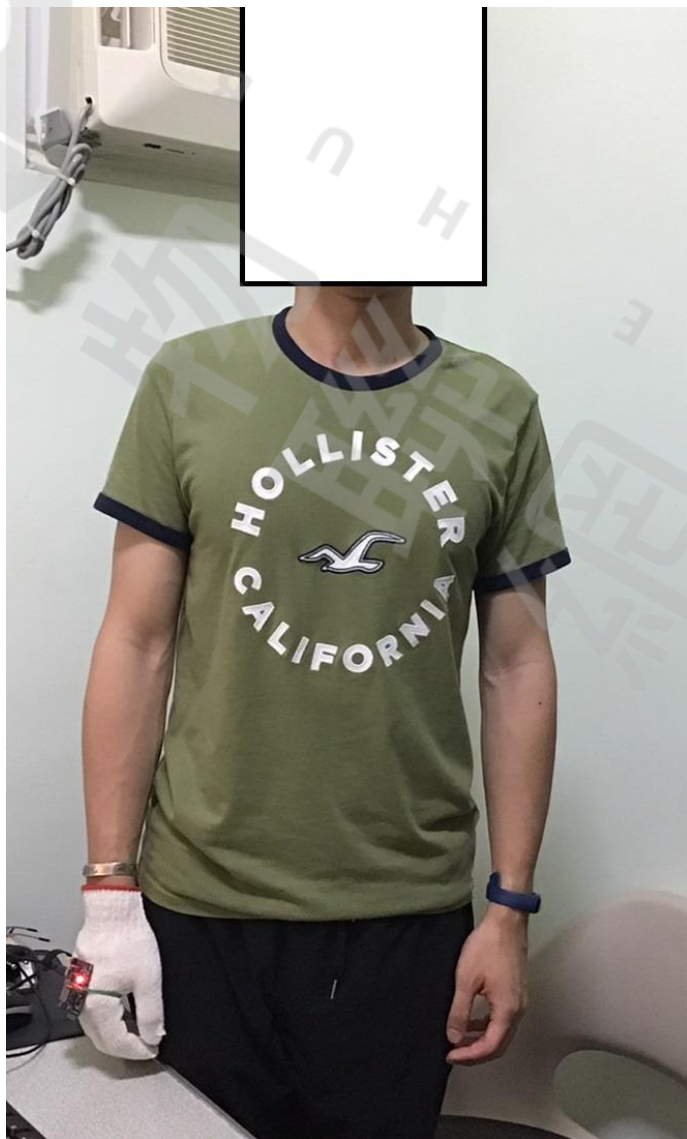


姿勢參考<https://www.funSPORT.com.tw/Article/Detail/37720?lang=zh-TW>

動作演示

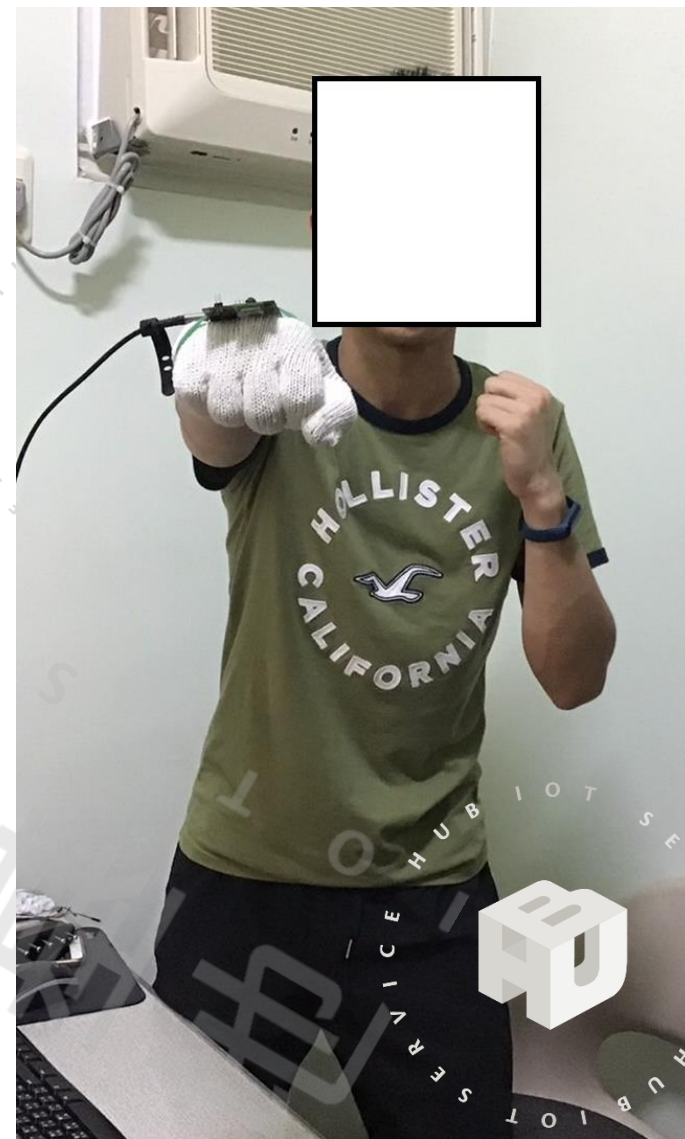
站立姿勢

手放於大腿旁



直拳

揮擊前動作
參考上頁的
右勾拳



蒐集數據設置

設定感測器來源為motion
其取樣頻率為100Hz

Sensor Properties

Device Plugin
CoreMaker-01

Capture Source
Motion

Sample Rate
100

Selected Sensors
☒ Accelerometer

Next Cancel

輸入錄製數據的檔名

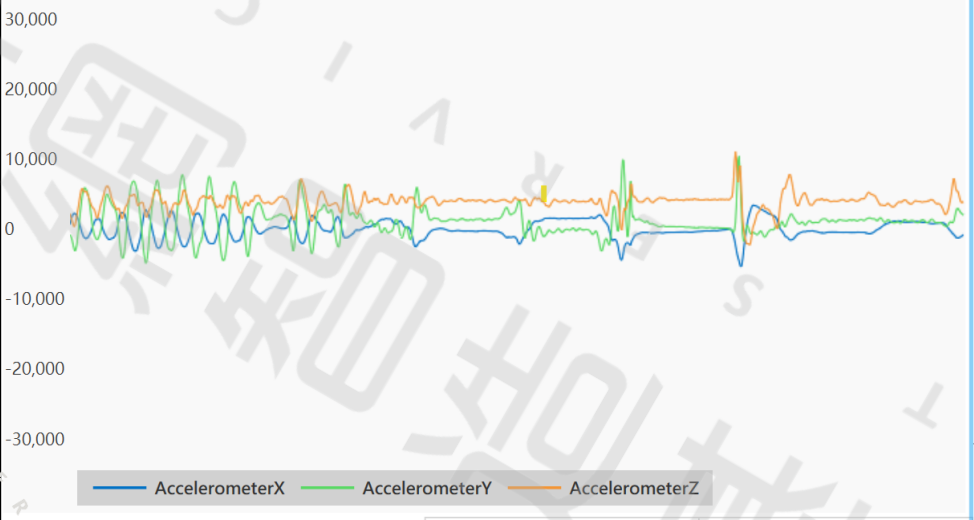
punch 1

File Edit Help

Project Explorer

Mode: Capture Switch Modes

CoreMaker-01



AccelerometerX AccelerometerY AccelerometerZ

Sensor Display

CoreMaker-01
COM3
Connected
Disconnect

Camera
Webcam
Disconnected
Connect

File Settings Live Labeling Test Model

File Name

Name lay down_001.csv

Text lay down

Label

File Metadata

Connection COM3

Device CoreMaker-01

+ Add Metadata

00:00:00:00

Start Recording

Capture Settings

開始錄製數據

標記數據

匯入數據:

選擇要標記的感測器檔案

Project Explorer

Files Knowledge Packs

Search Project Explorer

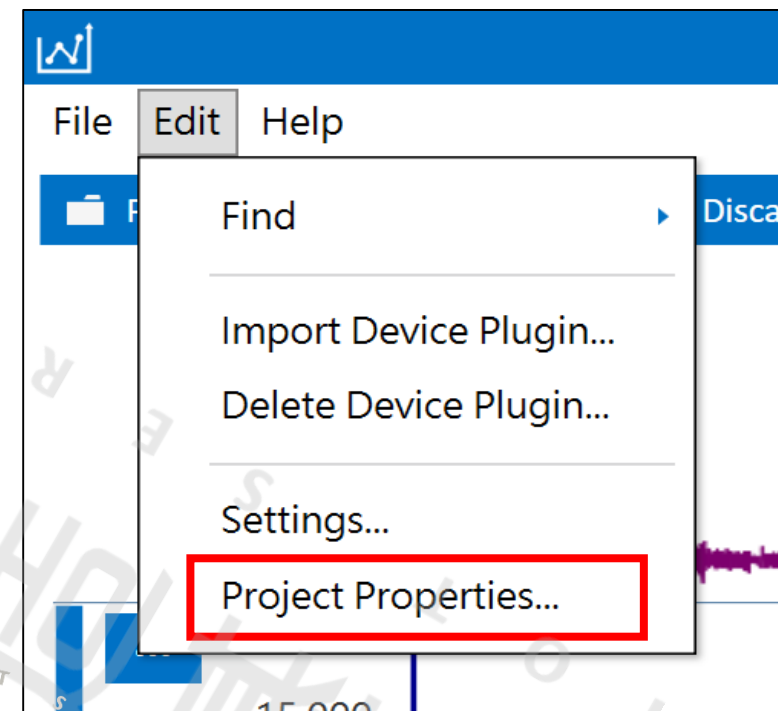
Status	File	Time	Length	Segments	Label Distribution	Uploaded
●	hook.csv	2:26	14,625	59		2022/9/1 下午 12:12
●	lay down.csv	2:18	13,893	16		2022/9/1 下午 12:15
●	straight and hold.csv	2:03	12,375	59		2022/9/1 下午 12:09

1 of 3 items selected

Session: (M) 1

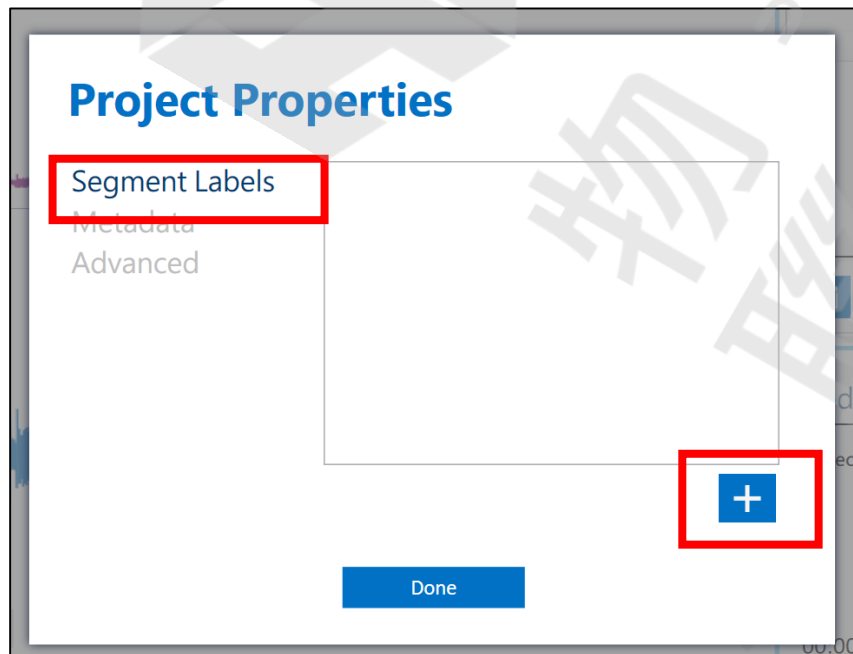
建立label:

點左上Edit，選擇 project properties



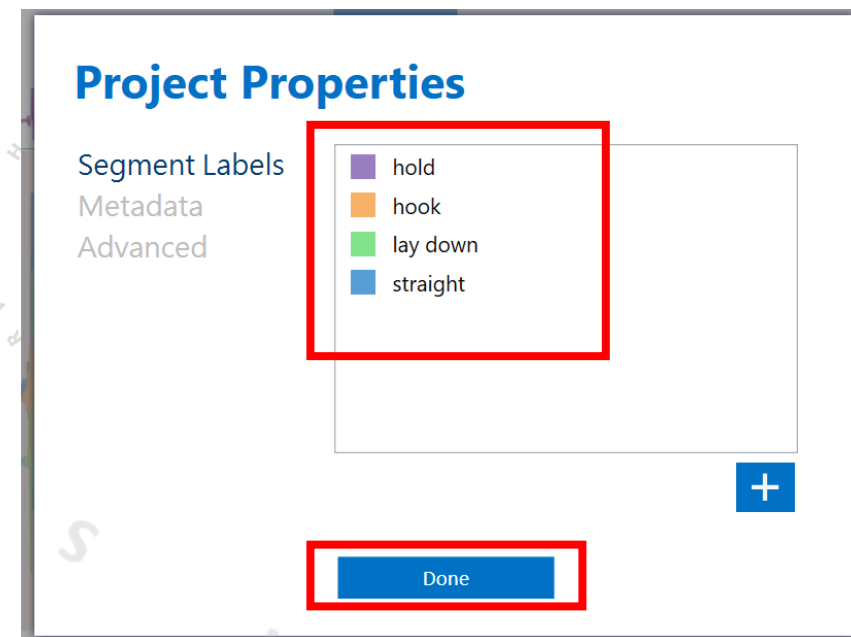
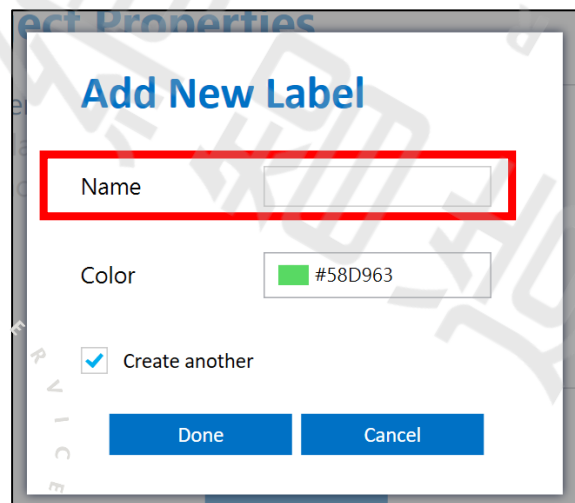
標記數據

此次範例要建立四種類別，分別是預備動作、右鉤拳、站立姿勢以及直拳



選擇Segment labels，
再點選 + 號。

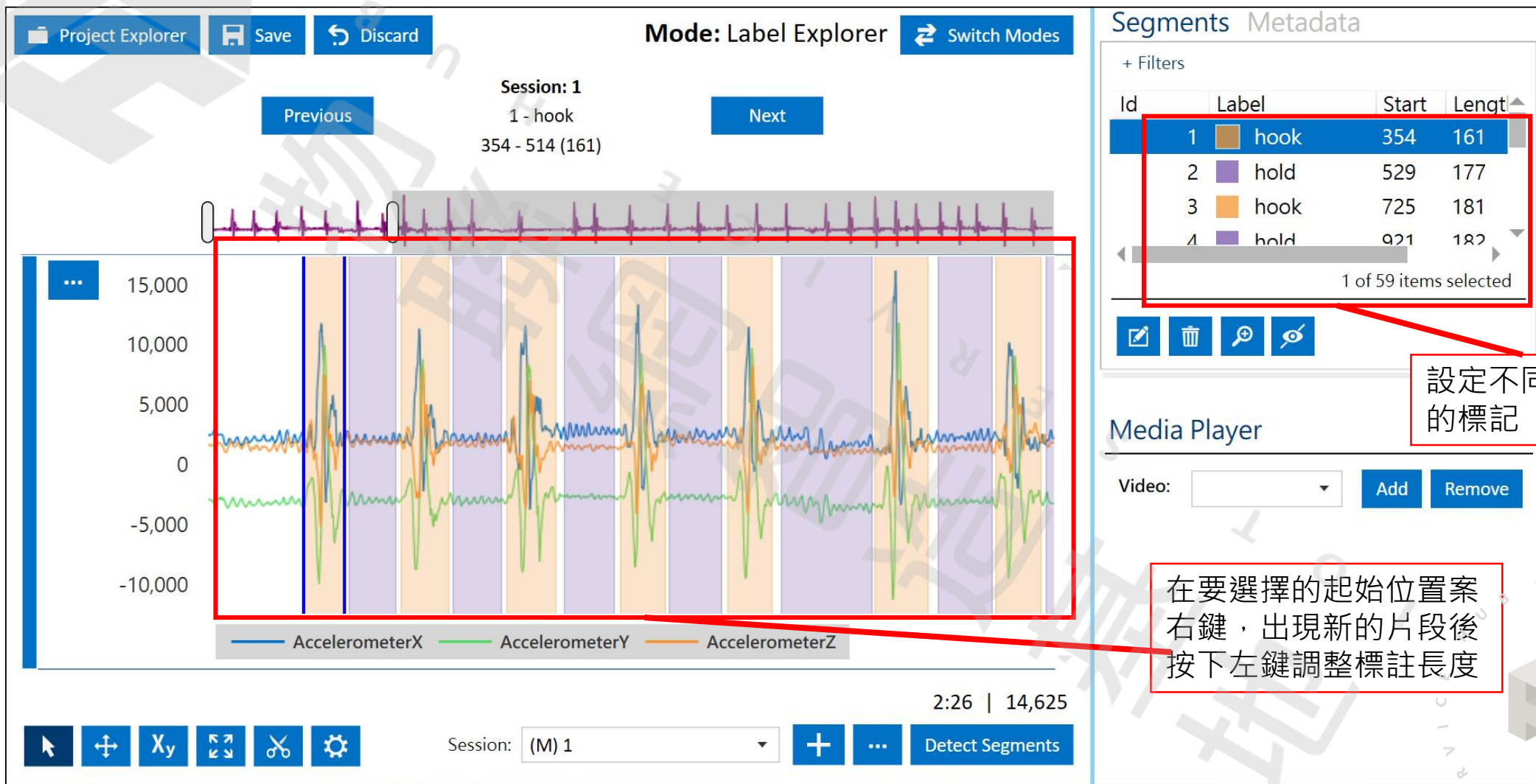
建立Label名稱，勾選
Create another 建立
另外一個類別。



確認建立無誤後，
點選Done完成設置。

標記數據

右鉤拳和預備姿勢的標記資料



設定不同動作的標記

在要選擇的起始位置案
右鍵，出現新的片段後
按下左鍵調整標註長度

標記數據

直拳和預備姿勢的標記資料



Segments Metadata

+ Filters

Id	Label	Start	Length
1	straight	570	134
2	hold	722	154
3	straight	893	125
4	hold	1,027	115

1 of 59 items selected

Media Player

Video:

Add

Remove

標記數據

站立姿勢的標記資料



Segments Metadata

+ Filters

Id	Label	Start	Length
1	lay down	490	412
2	lay down	1,399	444
3	lay down	2,304	405
4	lay down	3,176	289

1 of 16 items selected

Media Player

Video:

Add Remove

模型建立

Analytics Studio

app.sensiml.cloud/auth/login/

Learn to Code - To... Google Coursera | Online... YouTube Udacity edX | Free online c...

SensiML™

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

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

SensiML™ Analytics Studio

m103020015@student.nsysu.edu.tw

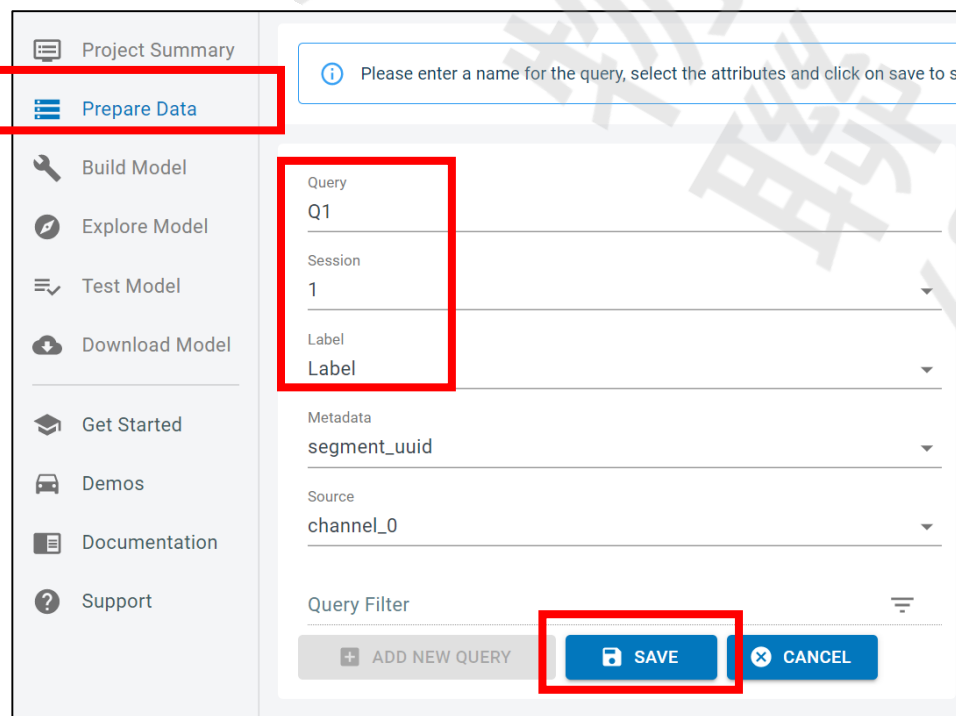
Open Project

NAME	FILES	PIPELINES	SIZE (MB)	QUERIES	MODELS	SEGMENTS
7-27	1	0	1.29	0	0	0 of 2500
MyProject	1	3	6.88	3	10	60 of 2500
MyProject16000	3	2	168.09	2	10	234 of 2500
punch	3	1	70.51	1	5	110 of 2500
punch 1	3	1	0.84	1	5	134 of 2500

網頁主畫面，選擇我們要訓練模型的Project

模型建立

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



Project Summary

Please enter a name for the query, select the attributes and click on save to save the query.

Prepare Data

Query: Q1

Session: 1

Label: Label

Metadata: segment_uuid

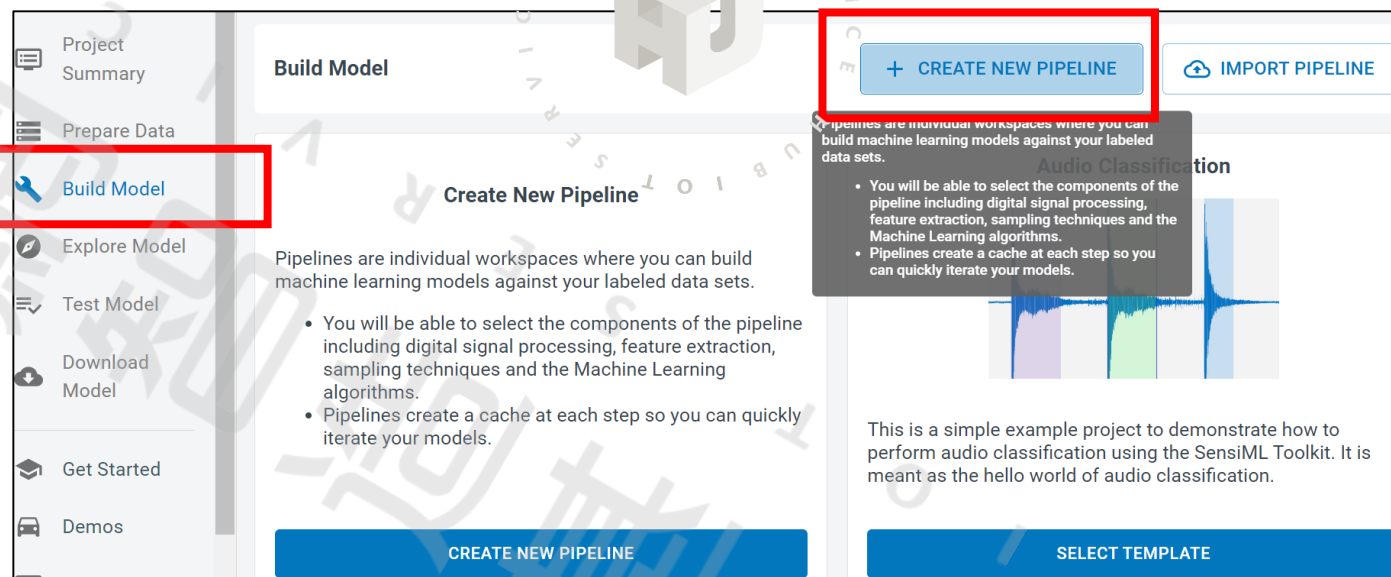
Source: channel_0

Query Filter

+ ADD NEW QUERY

SAVE CANCEL

於左方目錄欄點選Build Model，點選建立新的Pipeline



Project Summary

Build Model

+ CREATE NEW PIPELINE IMPORT PIPELINE

Pipelines are individual workspaces where you can build machine learning models against your labeled data sets.

Create New Pipeline

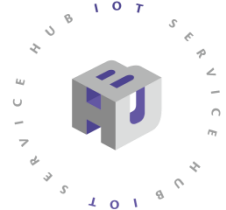
Pipelines are individual workspaces where you can build machine learning models against your labeled data sets.

- You will be able to select the components of the pipeline including digital signal processing, feature extraction, sampling techniques and the Machine Learning algorithms.
- Pipelines create a cache at each step so you can quickly iterate your models.

CREATE NEW PIPELINE SELECT TEMPLATE

Audio Classification

This is a simple example project to demonstrate how to perform audio classification using the SensiML Toolkit. It is meant as the hello world of audio classification.



模型建立

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

The screenshot shows the 'Create New Pipeline' dialog box with a progress bar at the top indicating two steps: '1 Select parameters' and '2 Create pipeline'. Step 1 is active. A blue information box contains text about AutoML. At the bottom, there is a toggle switch for 'Use SensiML AutoML to find the best machine learning algorithm' which is turned on. A red rectangle highlights a blue 'NEXT' button at the bottom right.

Create New Pipeline

1 Select parameters

2 Create pipeline

Info

AutoML is used to create a set of models within the desired statistical (accuracy, f1-score, sensitivity, etc.) and classifier size (neurons, features) parameters. As the algorithm iterates each optimization step, it narrow downs the searching space to find a desired number of models. The optimization terminates when the desired model is found or the number of iterations reaches the max number of iterations.

☒ Use SensiML AutoML to find the best machine learning algorithm

NEXT

The screenshot shows the 'Create New Pipeline' dialog box with the progress bar indicating Step 2 'Create pipeline' is active. A red rectangle highlights the input fields for 'New name' (containing 'P1') and 'Query' (containing 'Q1'). Below these fields, there is a toggle switch for 'Use session preprocessor' which is turned on. Further down, a table lists pipeline details: Name (Q1), Label Column (Label), Columns (channel_0), Metadata Columns (segment_uuid), Session (1), and Cache Status (NOT BUILT). A red rectangle highlights a blue 'CREATE PIPELINE' button at the bottom right.

Create New Pipeline

1 Select parameters

2 Create pipeline

New name: P1

Query: Q1

☒ Use session preprocessor

Name	Q1
Label Column	Label
Columns	channel_0
Metadata Columns	segment_uuid
Session	1
Cache Status	NOT BUILT

CREATE PIPELINE

模型建立

設定辨識長度以及辨識的間隔時間，其Window Size的數值要小於資料標記中擷取片段的最小長度

Segmenter

Segmenter ☐ JSON Editor

Segmenter Windowing

Window Size

100

1 16384

Slide

100

1 16384

☐ Use Training Slide

設定完成後點選OPTIMIZE開始建構模型

Project Summary

Prepare Data

Build Model

Explore Model

Test Model

Download Model

Get Started

Demos

Documentation

Support

Name: Q1
Type: Input Query

Name: Windowing
Type: Segmenter

Type: AutoML Parameters

Classifier SRAM	32000
f1-score	100

OPTIMIZE

模型輸出

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

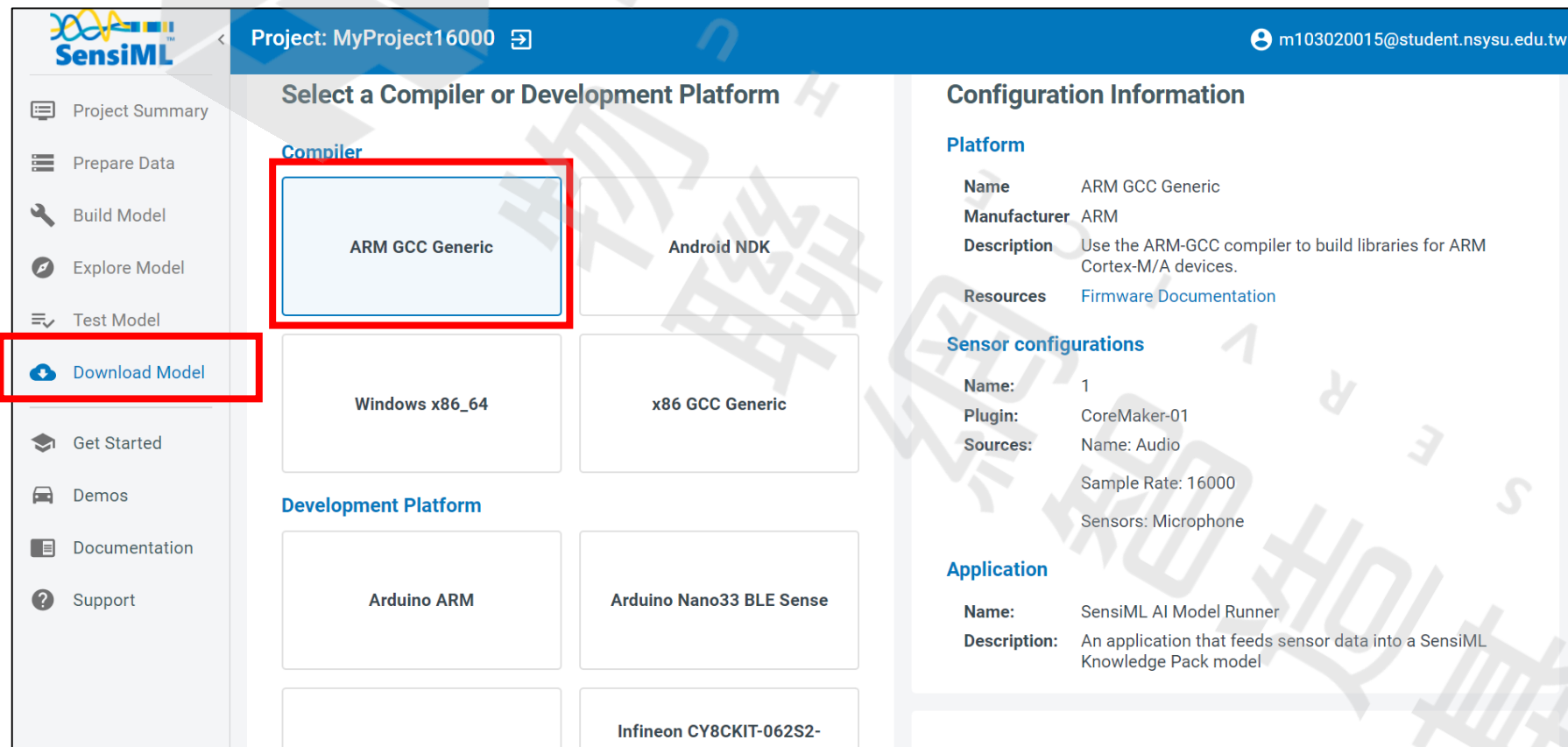
Project: punch 1 [CHANGE PROJECT](#)

PROJECT DESCRIPTION CAPTURES QUERIES PIPELINES KNOWLEDGE PACKS

NAME	CLASSIFIER	ACCURACY	MODEL SIZE (BYTES)	FEATURE COUNT	PIPELINE	CREATED DATE	UUID	EXPLORE	TEST	DOWNLOAD	RENAME	DELETE
P1_rank_4	Decision Tree Ensemble	100.00 %	4184	16	P1	2022/9/1 12:39:02 PM	fffa1657-14c3-44ac-9ab4-ecf10240f5b5					
P1_rank_3	PME	100.00 %	280	11	P1	2022/9/1 12:39:01 PM	98e0c9f1-9bba-4828-ae6-984452ee2118					
P1_rank_2	PME	100.00 %	308	8	P1	2022/9/1 12:39:01 PM	9e54a4c3-6b63-42f1-8356-e671ac3e497d					
P1_rank_1	Decision Tree Ensemble	100.00 %	4472	4	P1	2022/9/1 12:39:01 PM	3a3f0288-9e1c-4fa2-9024-b8113b7f4012					
P1_rank_0	Decision Tree Ensemble	100.00 %	1838	4	P1	2022/9/1 12:39:01 PM	432917f0-7dd0-4866-99bb-fa0d9de6def2					

1-5 of 5

模型下載



The screenshot shows the SensiML web interface for a project named 'MyProject16000'. The left sidebar contains a menu with the following items: Project Summary, Prepare Data, Build Model, Explore Model, Test Model, **Download Model** (highlighted with a red box), Get Started, Demos, Documentation, and Support. The main content area is titled 'Select a Compiler or Development Platform' and is divided into two sections: 'Compiler' and 'Development Platform'. The 'Compiler' section has a red box around the 'ARM GCC Generic' option. The 'Development Platform' section shows options like 'Arduino ARM', 'Arduino Nano33 BLE Sense', and 'Infineon CY8CKIT-062S2-'. On the right, the 'Configuration Information' section displays details for the selected platform, including Name (ARM GCC Generic), Manufacturer (ARM), Description, Resources (Firmware Documentation), Sensor configurations (Name: 1, Plugin: CoreMaker-01, Sources: Name: Audio, Sample Rate: 16000, Sensors: Microphone), and 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。

模型下載

更改部分:

Float options → Soft FP

Compiler → GNU 10.3.1

按下Download下載模型的壓縮檔

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

模型輸出的數字1到4，代表相對應的動作

- 1.預備動作
- 2.右鉤拳
- 3.站立姿勢
- 4.直拳

Knowledge Pack information

Class Map:

1 - hold

2 - hook

3 - lay down

4 - straight

Resource Estimates

模型編譯

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

韌體程式碼

將原本第80行程式碼改為NULL，第82行改為kb_run_model，使CoreMaker-01開發板使用加速度感測器辨識

```
SensorHub.cpp
C: > Users > 88692 > CoreMaker-01 > sensors > SensorHub.cpp

66  * The arrangement must be the same with
67  *
68  */
69  Sensor *SensorHub::sensors[] =
70  {
71      &acoustic_node,
72      &bme680,
73      &kx122,
74      &gmc306,
75      &gmp102
76  };
77
78  RUN_MODEL run_ai_model[] =
79  {
80      NULL,
81      NULL,
82      kb_run_model,
83      NULL,
84      NULLS
85  };
86
```

修改辨識結果打印訊息，降低資料量

```
> sensors > SensorHub.cpp

if(m_DCLStatus == DCL_CONNECTED)
{
    serial.send((uint8_t*)m_dataBuffer, m_dataLen);
}
else if(run_ai_model[i])
{
    int ret = run_ai_model[i]((SENSOR_DATA_T*)m_dataBuffer, m_dataLen/sizeof(SENSOR_DATA_T));
    if (ret > -1)
    {
        printf("%d\n", ret);
        kb_reset_model(0); // Reset running model to initial state.
    }
    else if (ret == -2)
    {
        printf("This segment has been filtered.\n");
    }
    else if (ret < -2)
    {
        printf("AI error: %d\n", ret);
    }
}
```


編譯與燒錄韌體

- 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯
- 使用CMC_ISP.exe透過USB進行燒錄下載bin檔至CoreMaker01開發板上

Step1執行編譯指令

```
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>cd CoreMaker-01

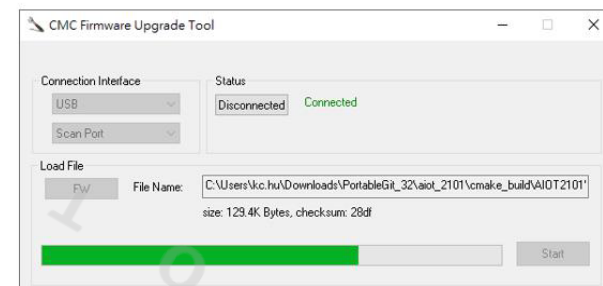
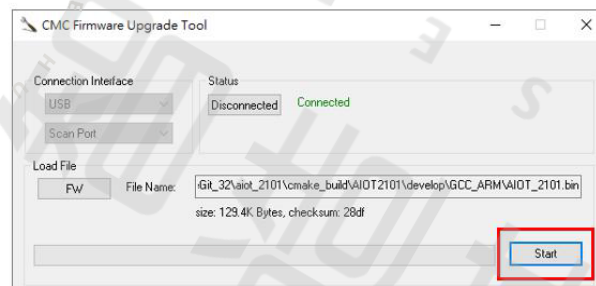
C:\Users\kc.hu\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
```

Step2編譯出bin檔

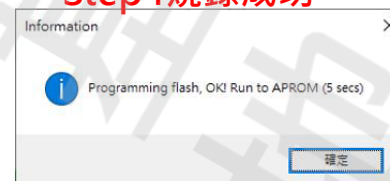
```
命令提示字元
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex

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

Step3連接開發板

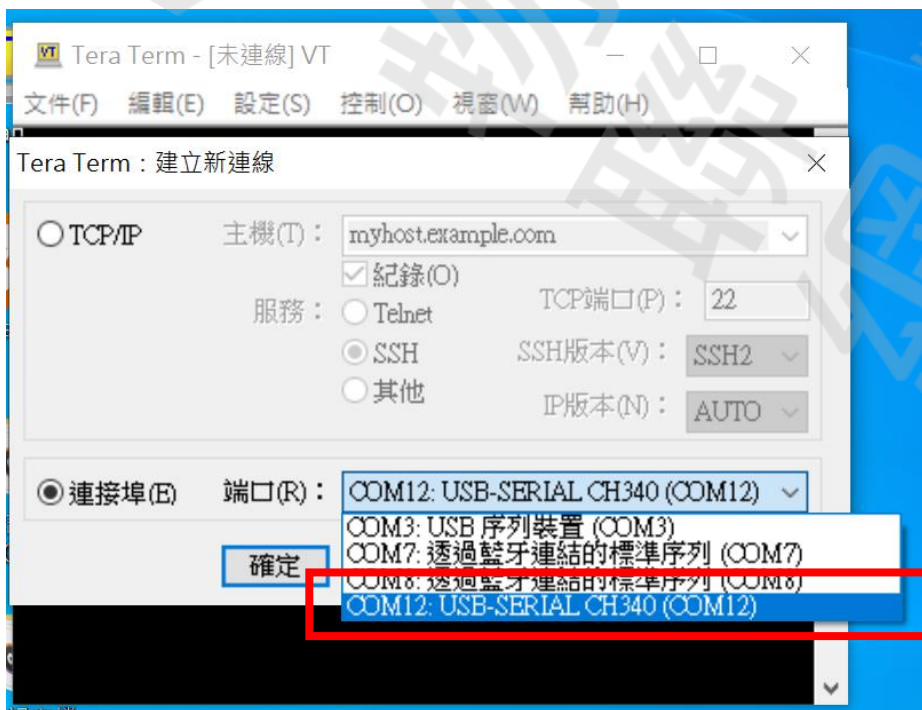


Step4燒錄成功

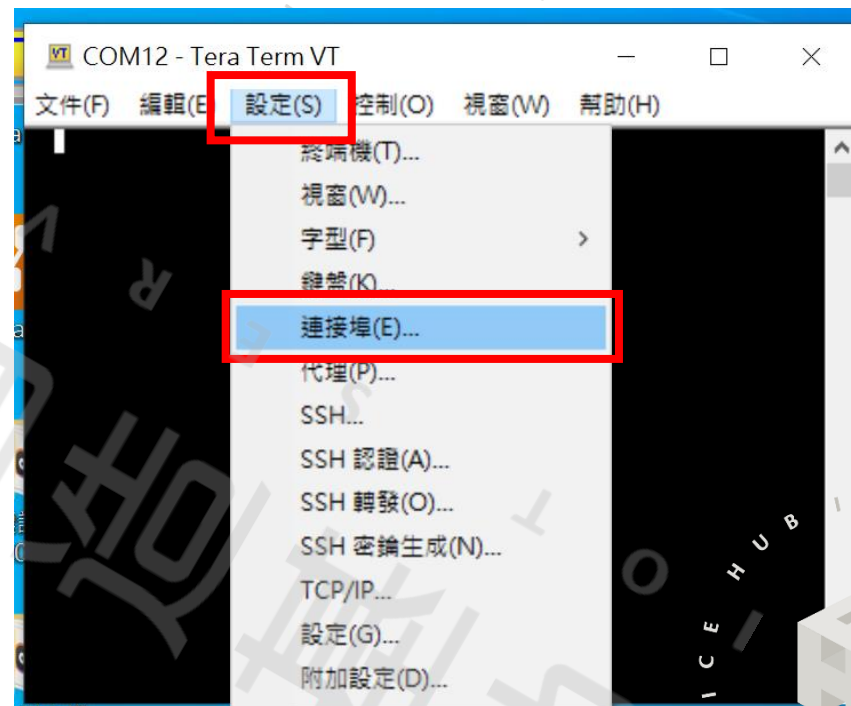


顯示輸出結果

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)



選擇連接的序列埠



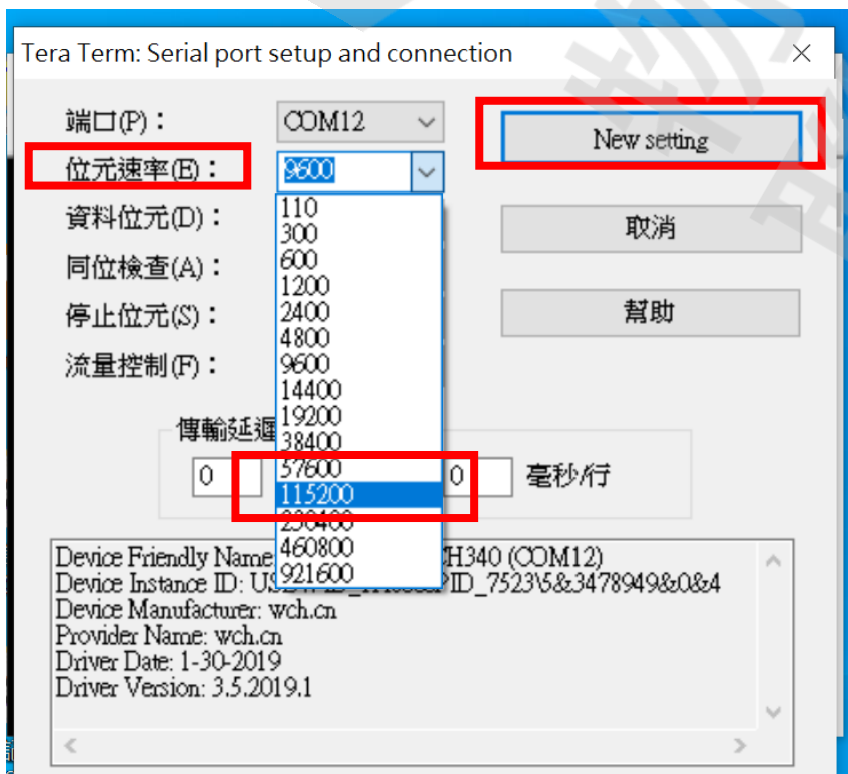
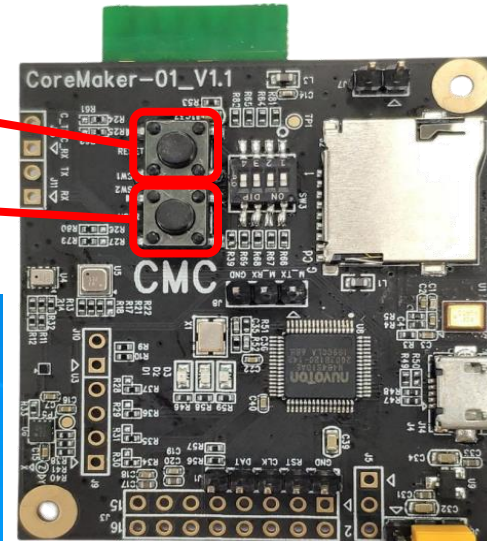
選擇設定下的序列埠

顯示輸出結果

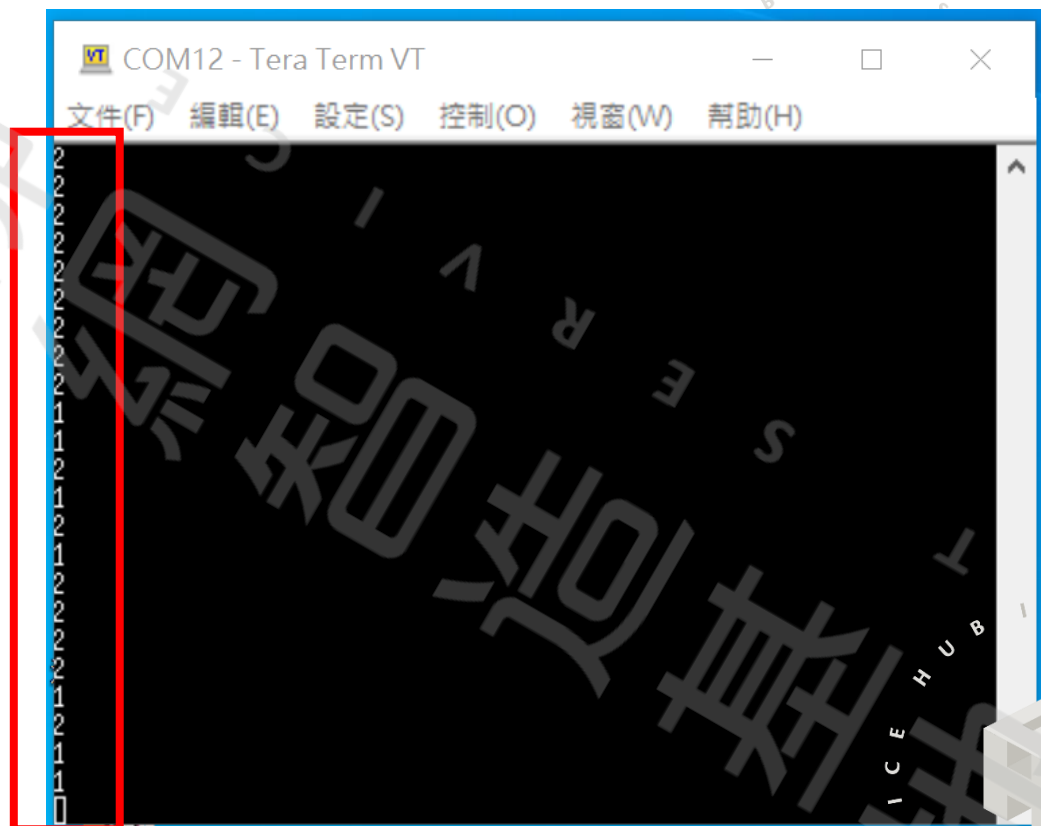
連線設定完成後，按下 CoreMaker-01上的 SW2 鍵，當我們開始拳擊動作時，可在螢幕視窗上看到 AI 辨識的分類結果。

按鍵1

按鍵2



位元速率設定11520

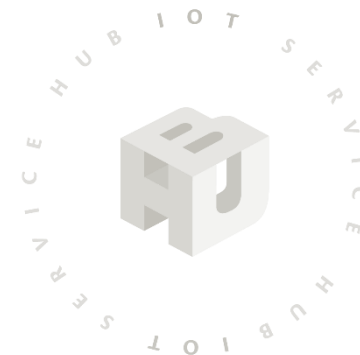


Tera Term顯示結果



實際測試影片

- 請參考Ideas Hatch網站說明





Thank you

