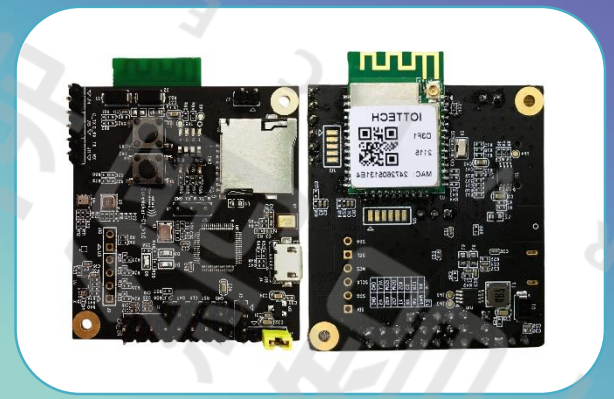
CoreMaker-01



人體姿態智能辨識

指導單位：

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

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:

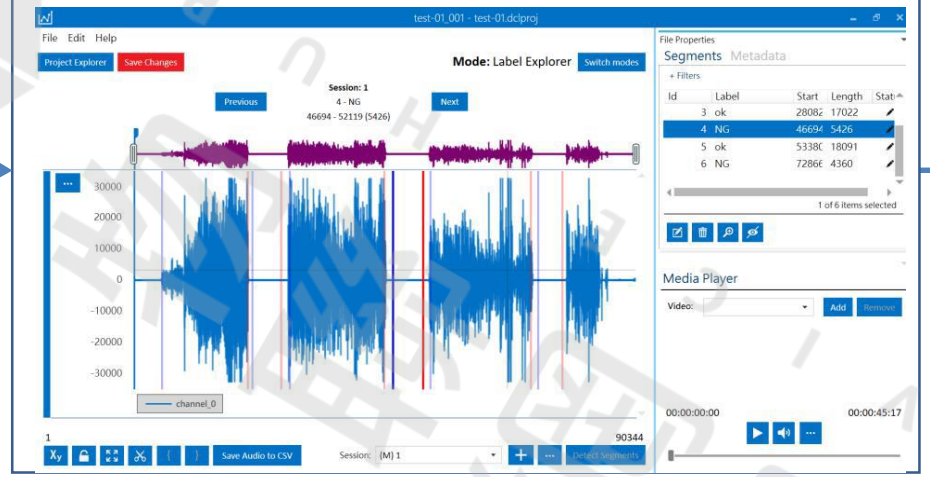
1. 數據隱私性以及資料安全性
2. 可離線工作, 反應即時
3. 低功耗, 持久性佳
4. 小且容易安裝, 成本低

開發流程圖

開發

手部姿態

蒐集與標註資料(Data Capture Lab)

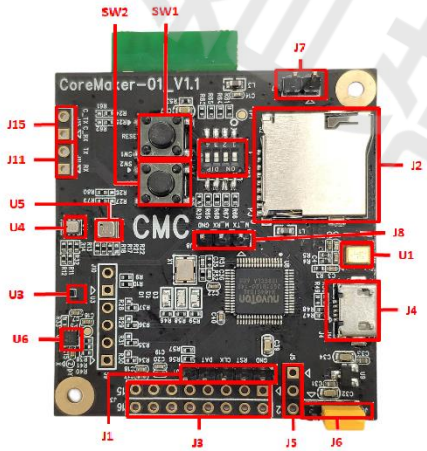


訓練與編譯模型(Analytics Studio)

下載燒錄

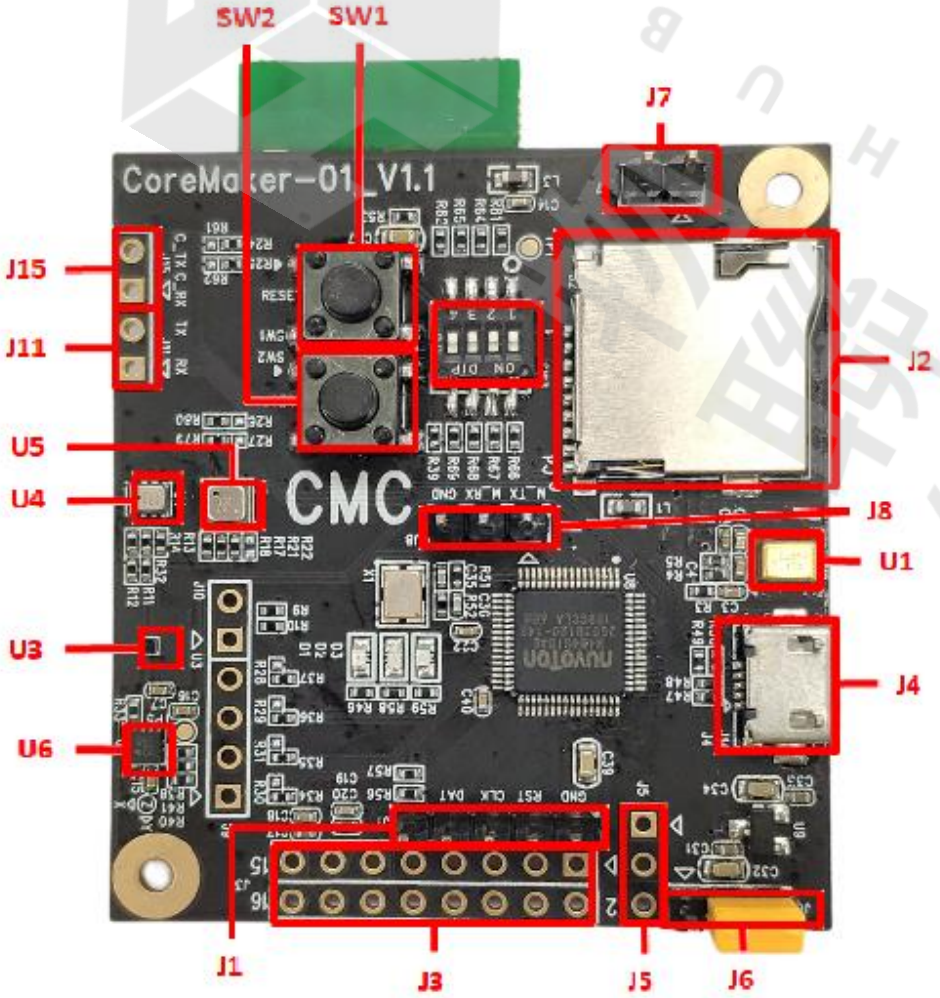
應用

手部姿態



- 直拳
- 右鉤拳
- 預備動作
- 手放於大腿旁

開發環境介紹---硬體



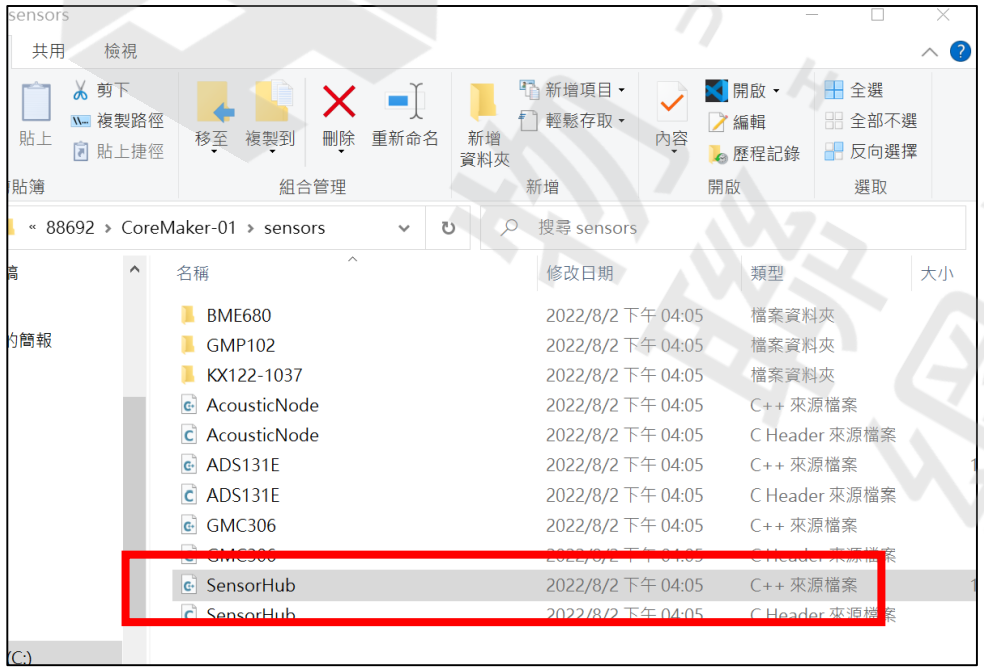
- **主控制器M484SIDAE**
  - 工作主頻高達 192 MHz，512KB Flash/160 KB SRAM，內建FPU和DSP可加速算法，1.8V到3.6V操作電壓，以及 - 40 ℃ 到 105 ℃ 的工業級工作溫度
- **板上元件**
  - 磁力計(GMC306A)
  - 氣壓計(GMP102)
  - 環境偵測(BME680)
  - 加速度計(KX122-1037)
  - 麥克風
  - WIFI/BLE無線模組
- **支援介面**
  - mikro bus

開發環境建立

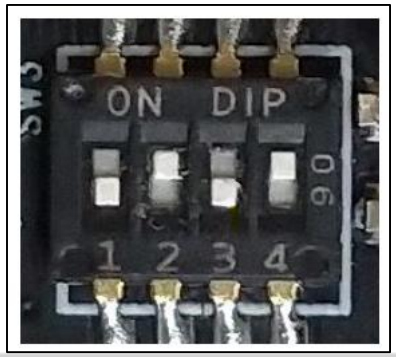
- CoreMaker-01_V1.1 操作指南:環境設定、下載及編譯原始碼、加入AI
- 範例程式
  - WiFi 運行指南
  - SD Card 運行指南
  - LED 燈效控制範例
- Sensor Datasheet
  - KX122-103 加速度感知器
  - BME680 環境感知器
  - GMP102 氣壓感知器
  - SPU0410LR5H-QB 聲音感知器
- CoreMaker-01電路圖
  - CoreMaker-01_V1.1
- 應用指南
  - CoreMaker-01_V1.1電源供應模式
  - CoreMaker-01_V1.1 PWM控制LED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

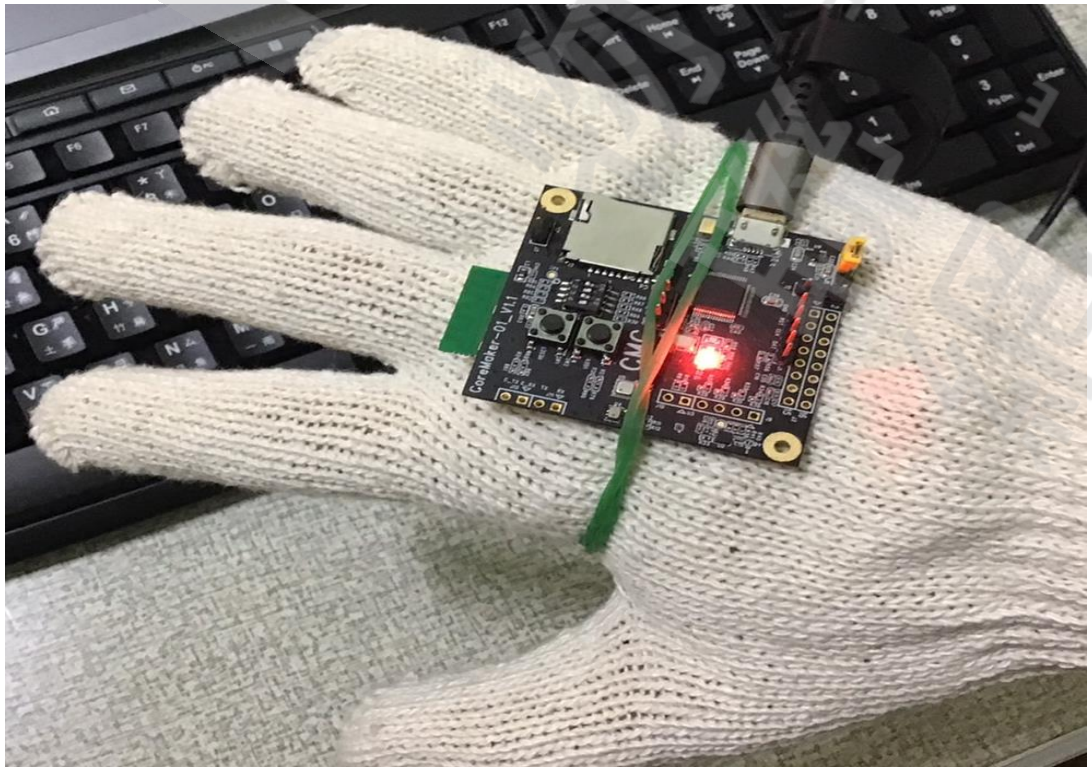


把SW3 撥鈕調整至加速度蒐集模式

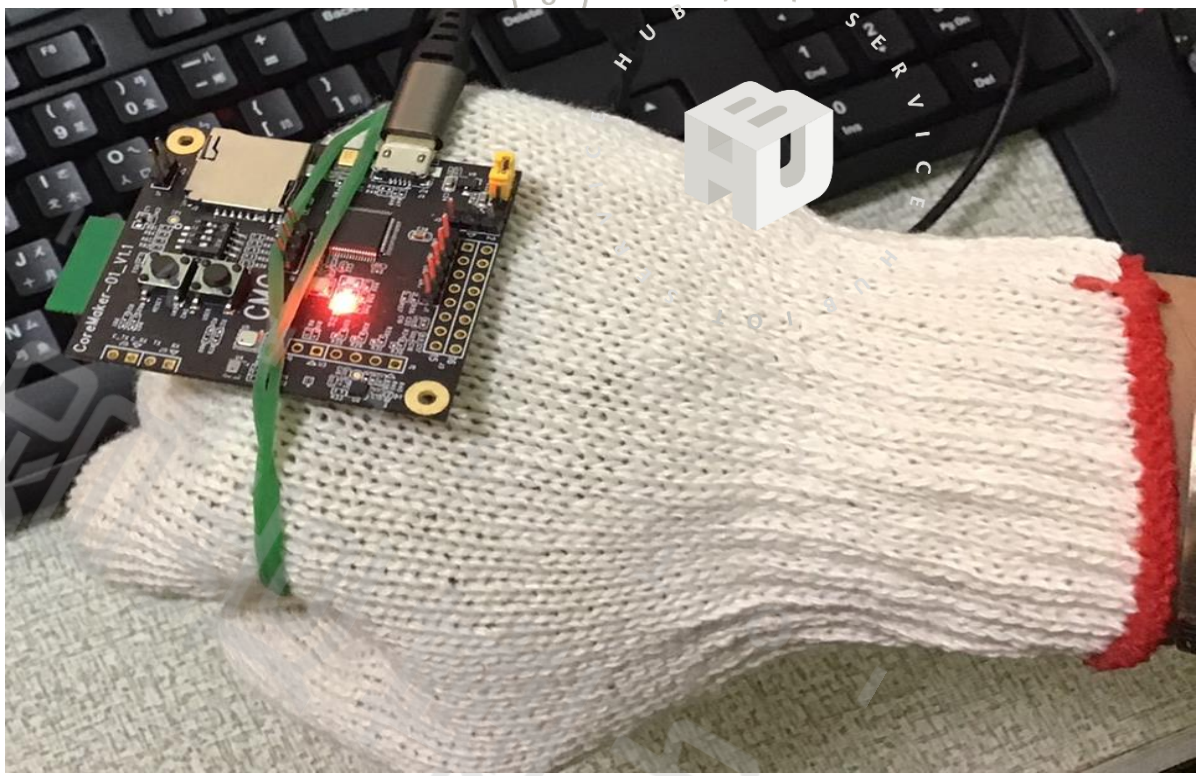
在SensorHub.cpp的60行
將聲音取樣類別調整至100Hz當作採樣頻率

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

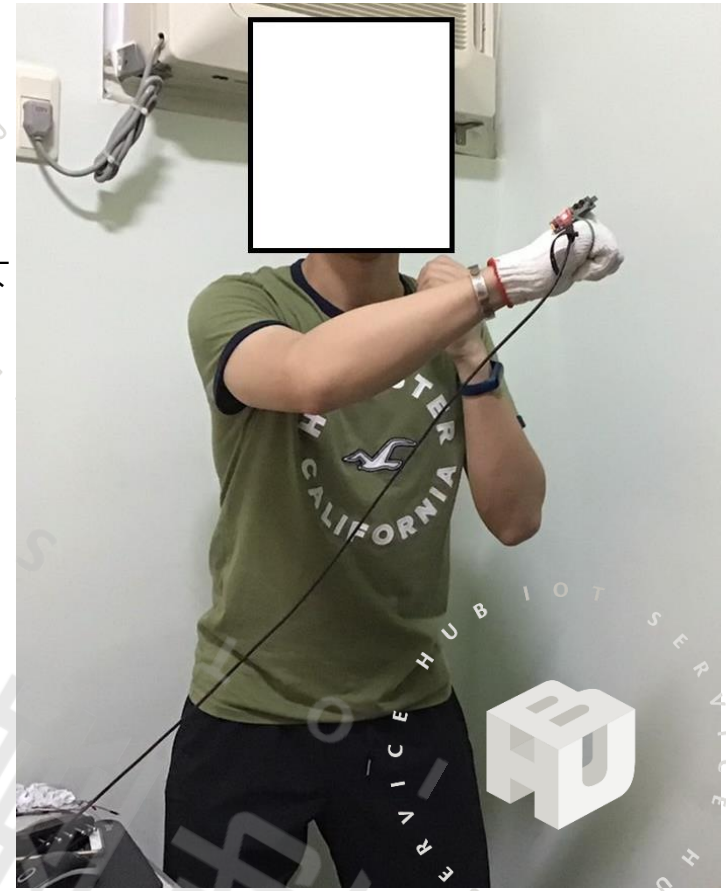
動作演示

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

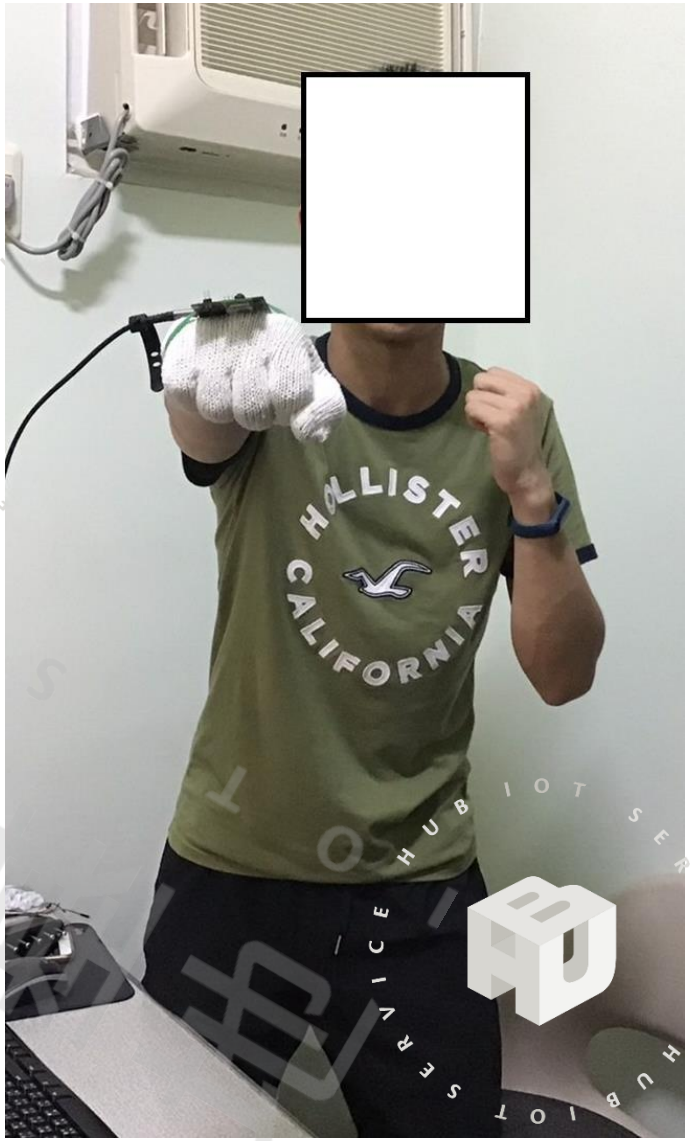
動作演示

站立姿勢

手放於大腿旁

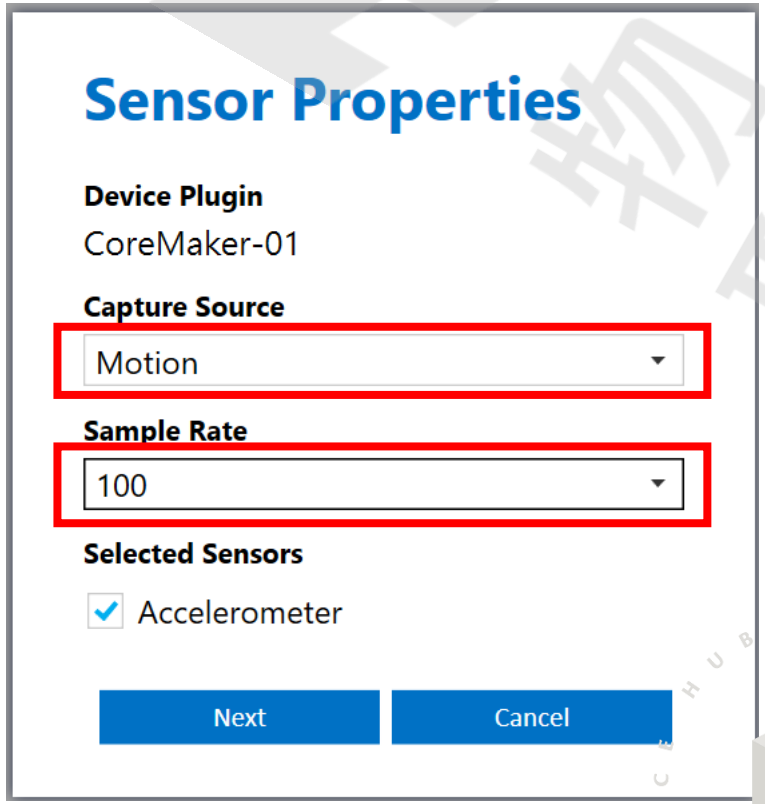
直拳

揮擊前動作
參考上頁的
右勾拳

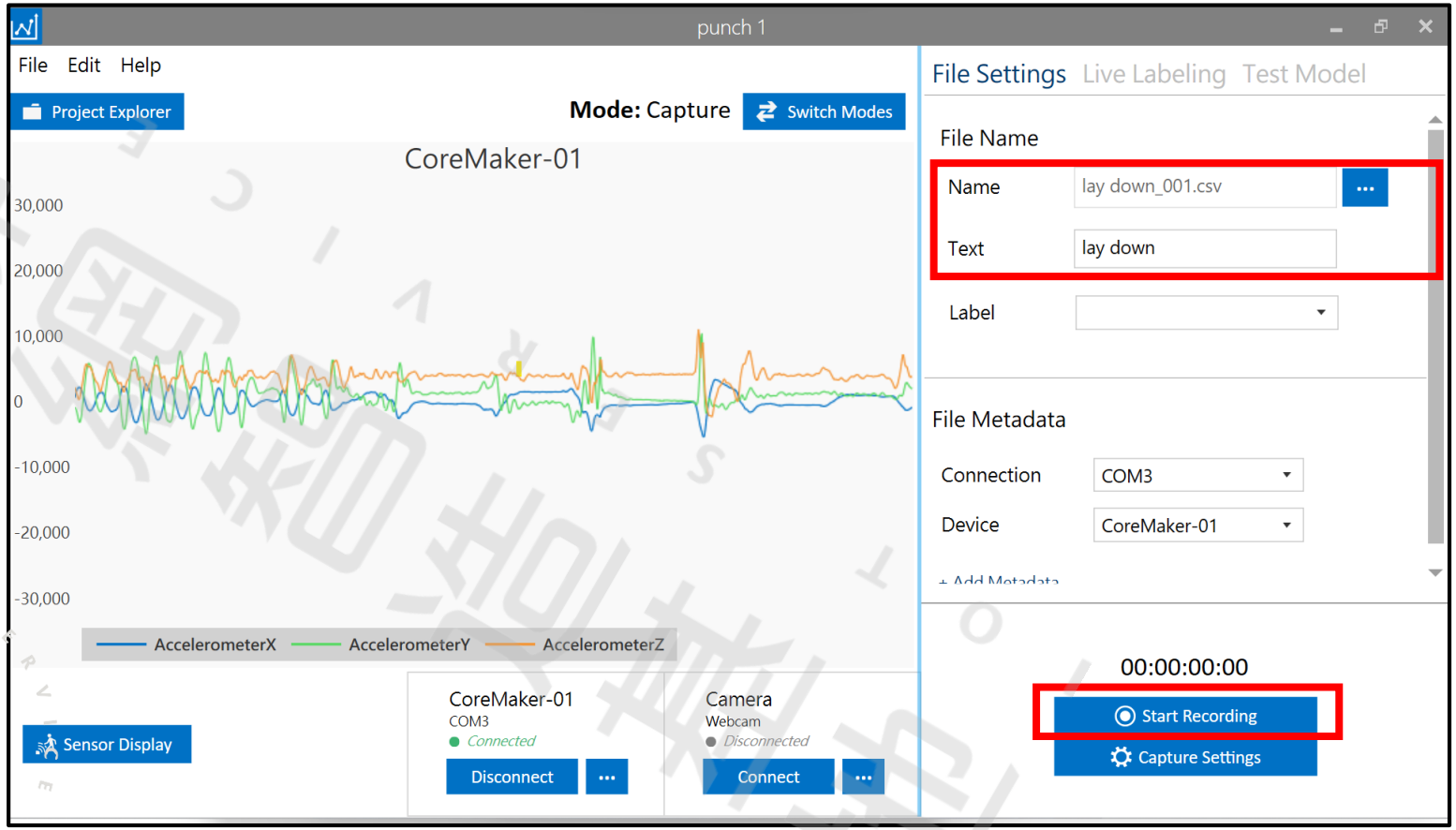


蒐集數據設置

設定感測器來源為motion
其取樣頻率為100Hz



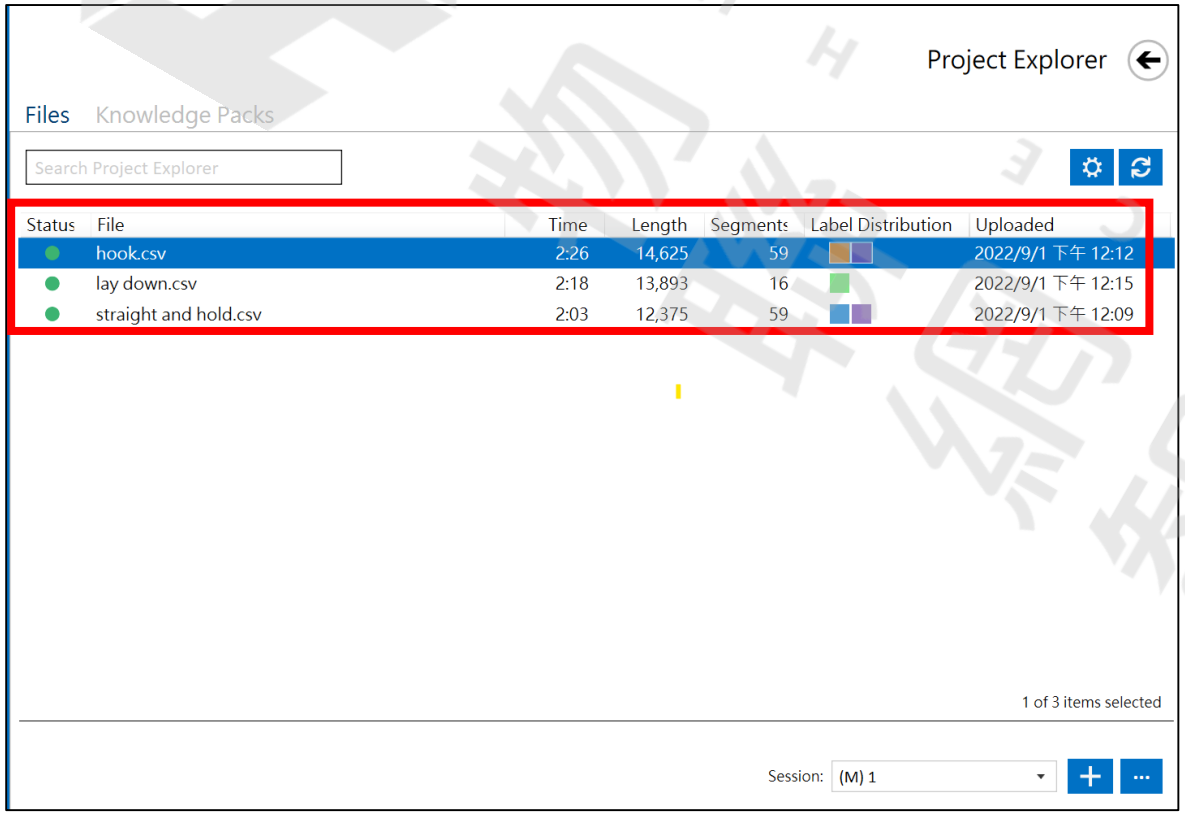
輸入錄製數據的檔名



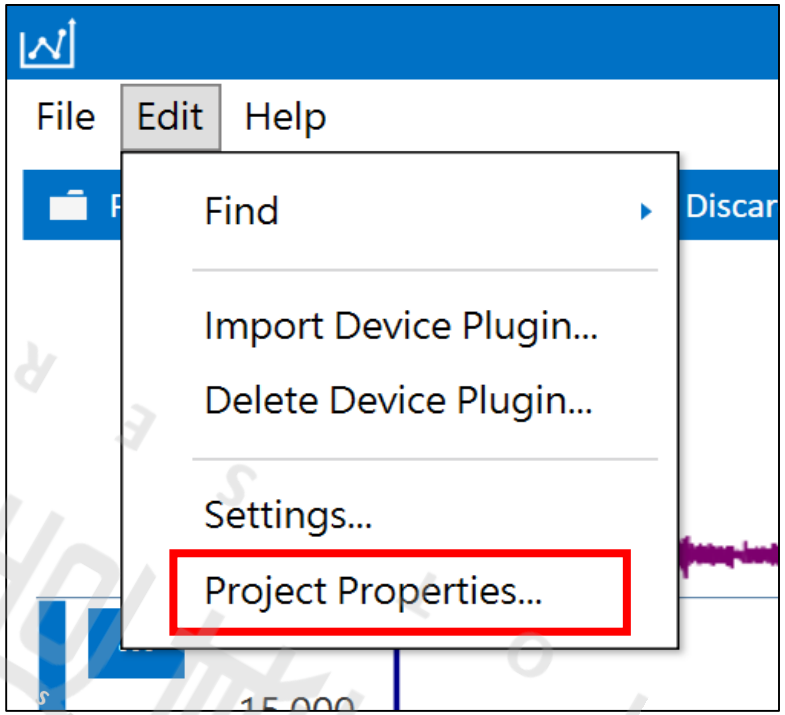
開始錄製數據

標記數據

匯入數據:
選擇要標記的感測器檔案

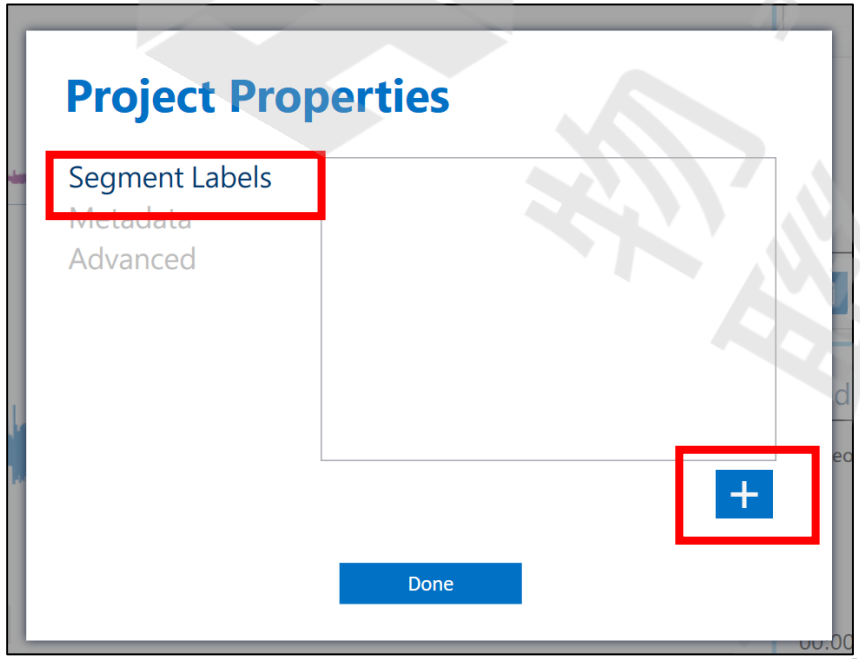


建立label:
點左上Edit，選擇 project properties



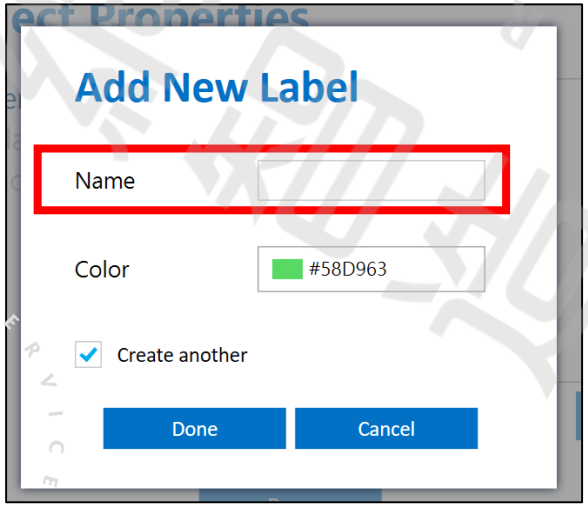
標記數據

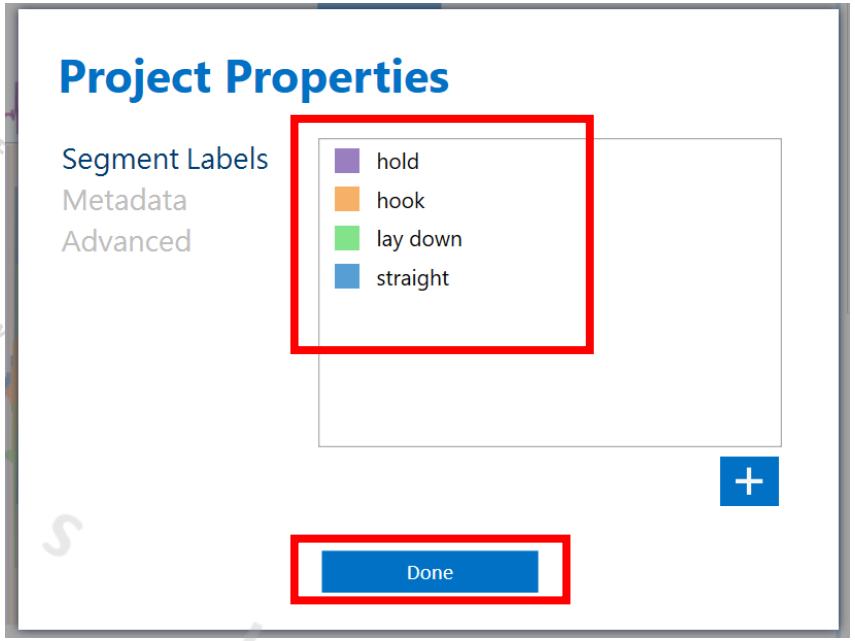
此次範例要建立四種類別，分別是預備動作、右鉤拳、站立姿勢以及直拳



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。

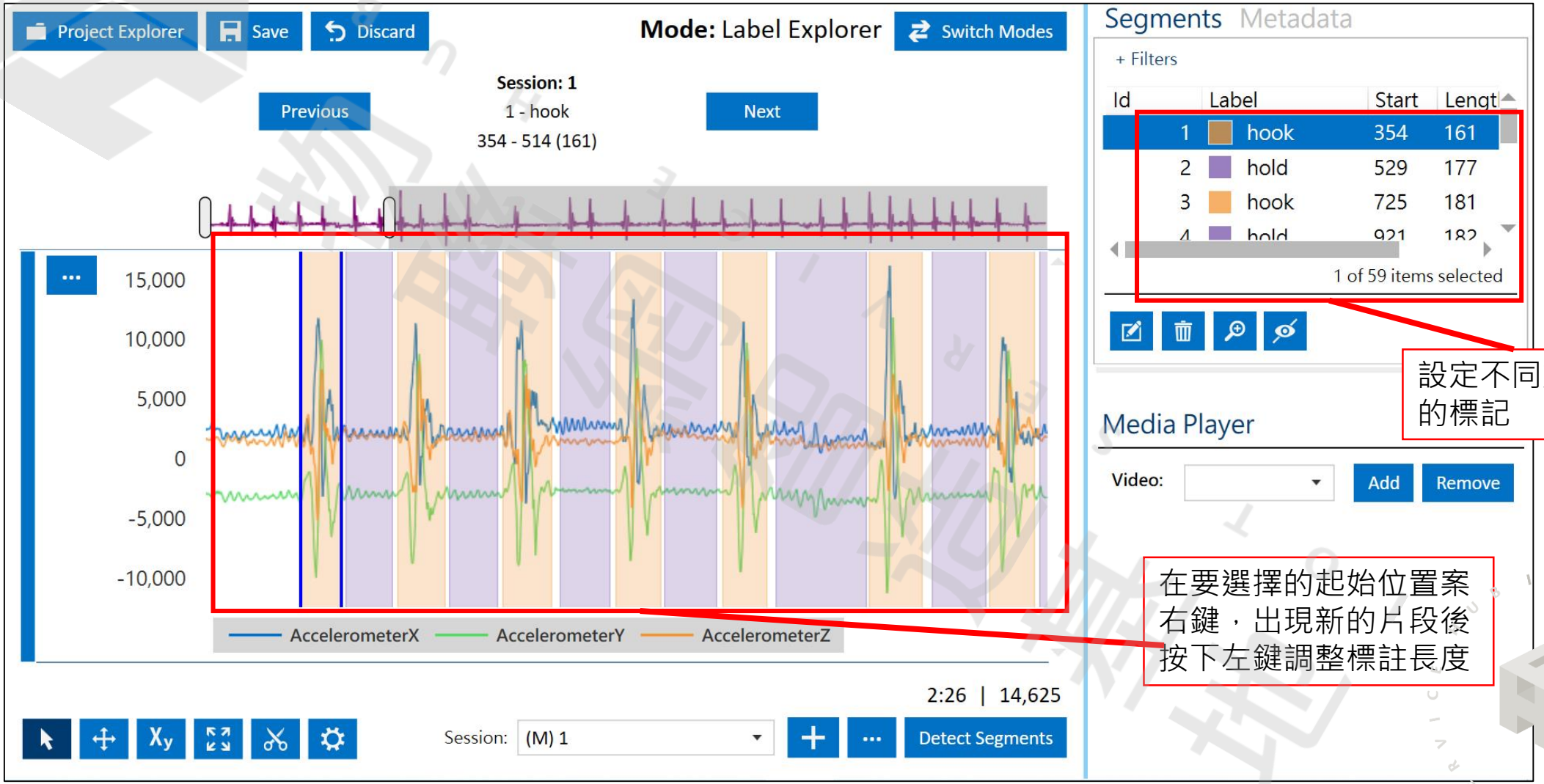




確認建立無誤後，
點選Done完成設置。

標記數據

右鉤拳和預備姿勢的標記資料



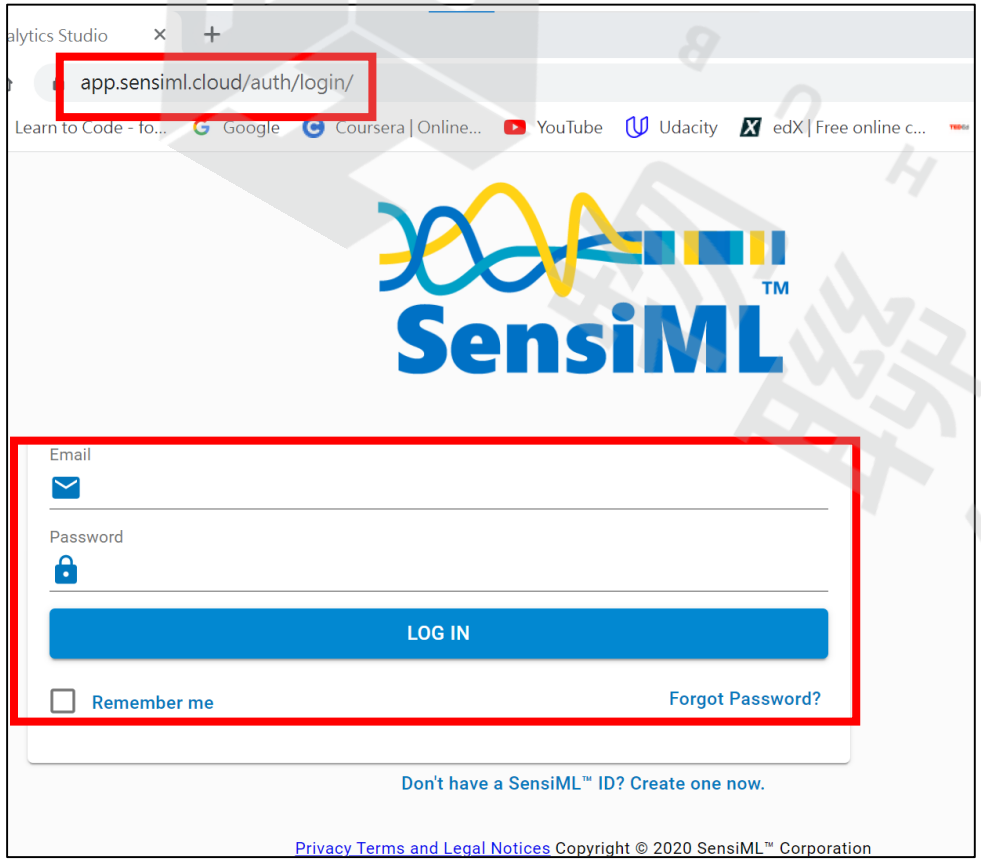
標記數據

直拳和預備姿勢的標記資料

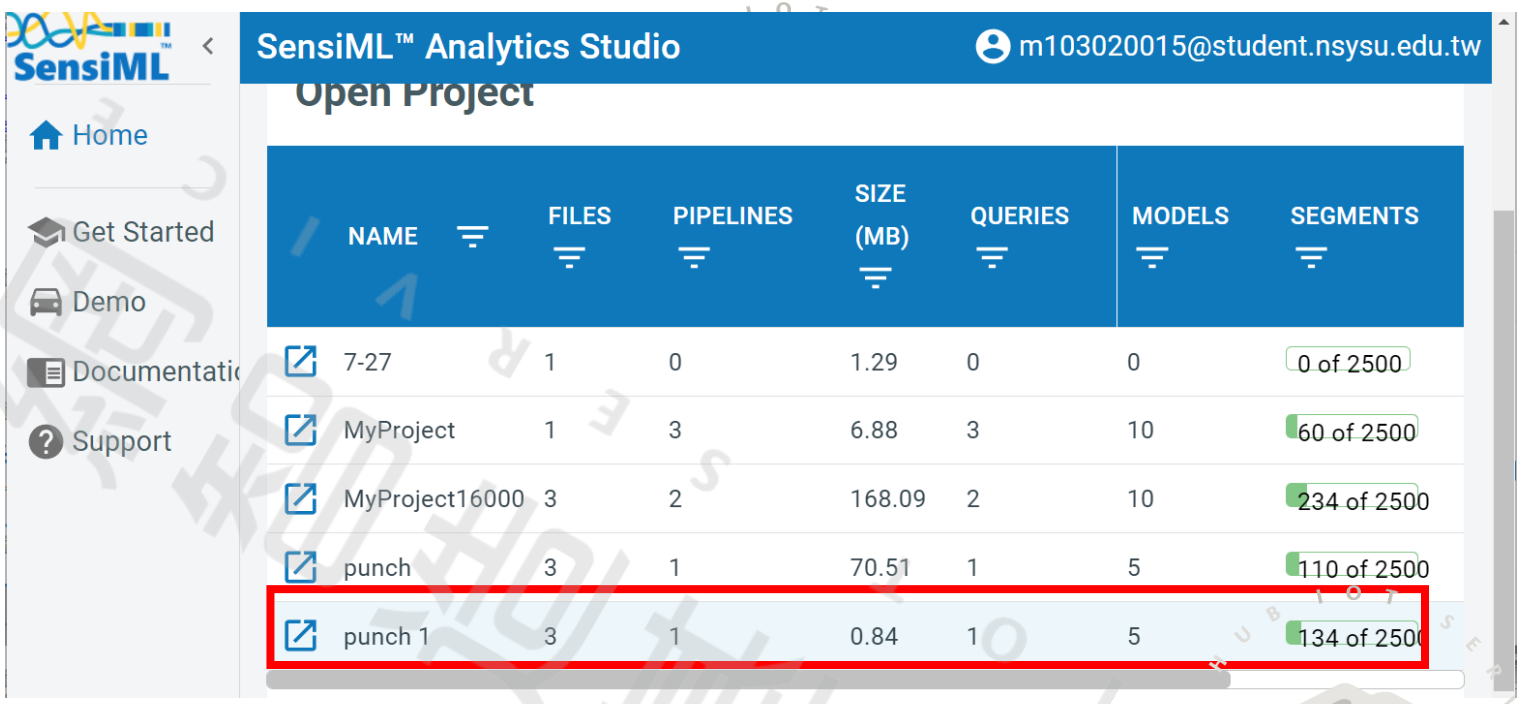
標記數據

站立姿勢的標記資料

模型建立



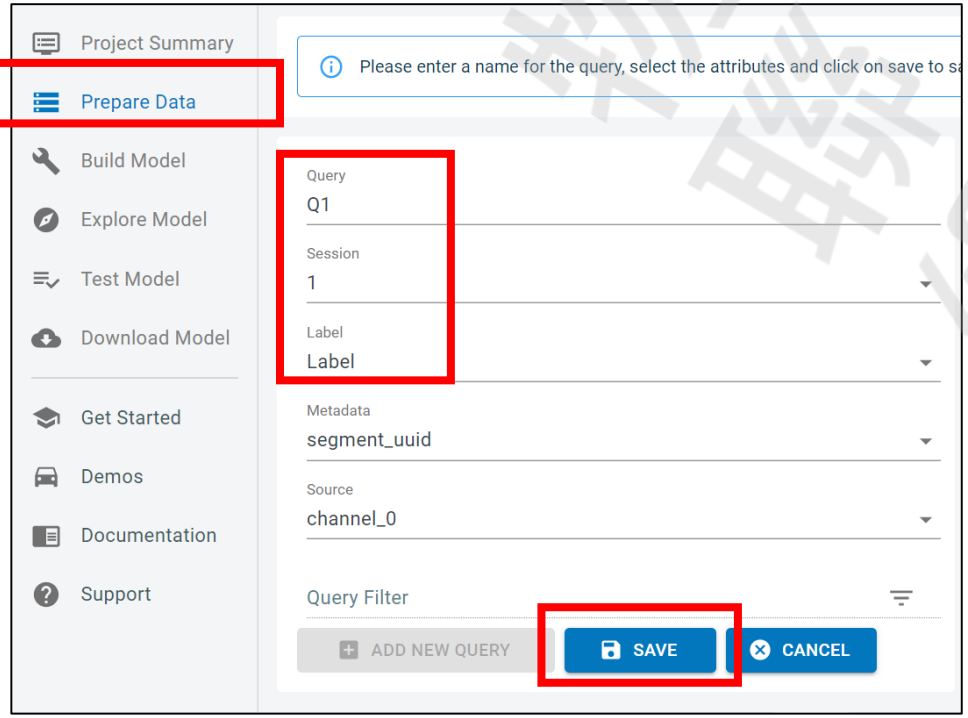
開啟 SensiML Analytics Studio 網頁:
https://app.sensiml.cloud/auth/login/
登入註冊帳號



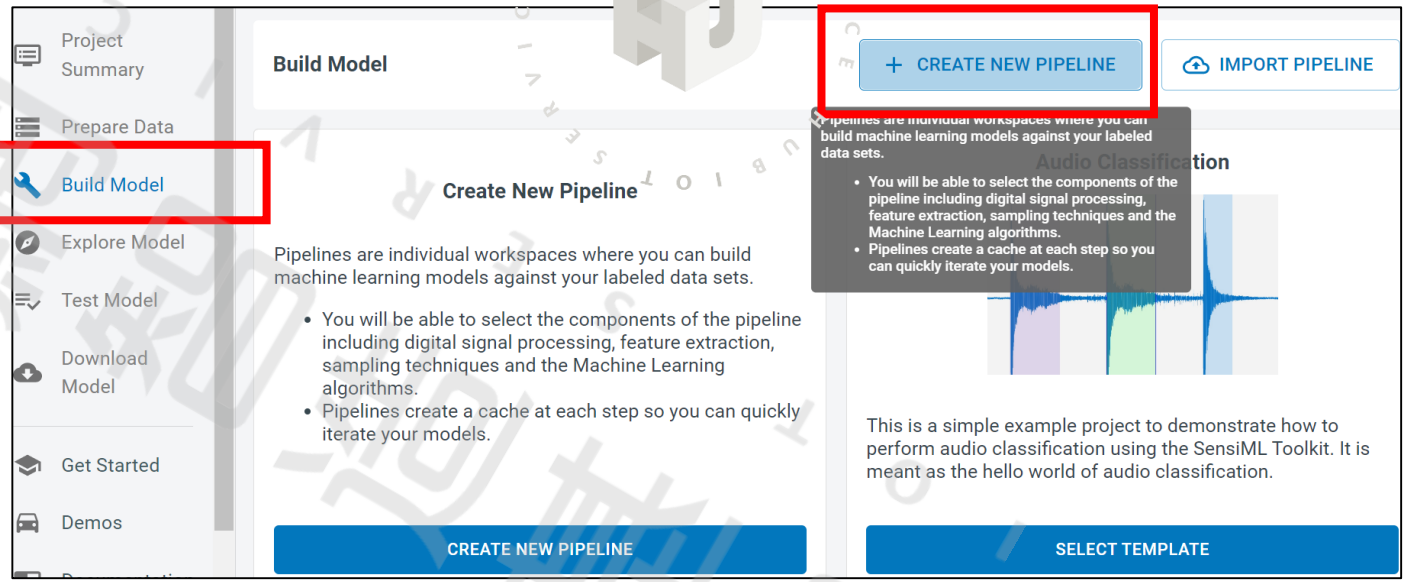
網頁主畫面，選擇我們要訓練模型的Project

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

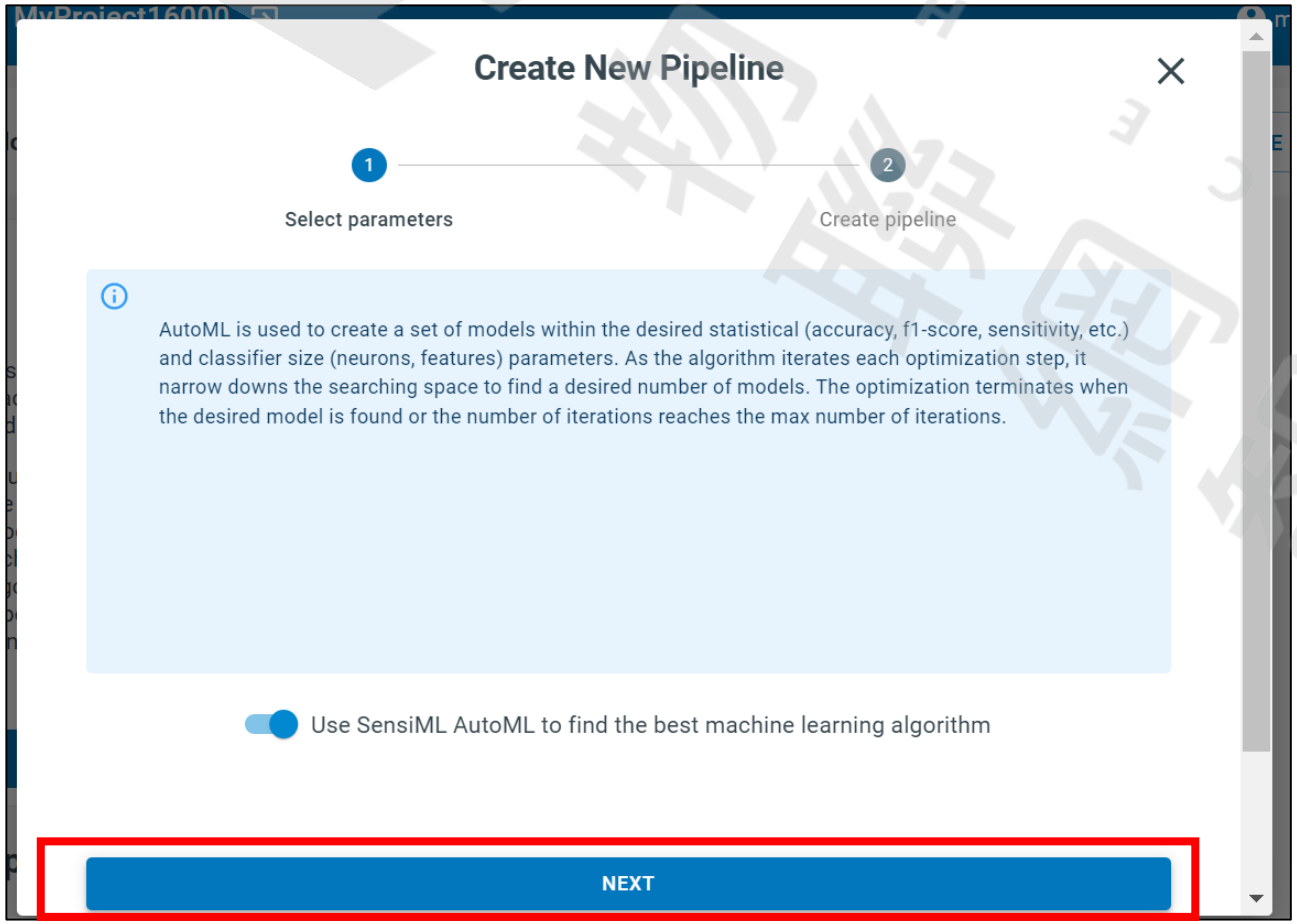


於左方目錄欄點選Build Model，點選建立新的Pipeline

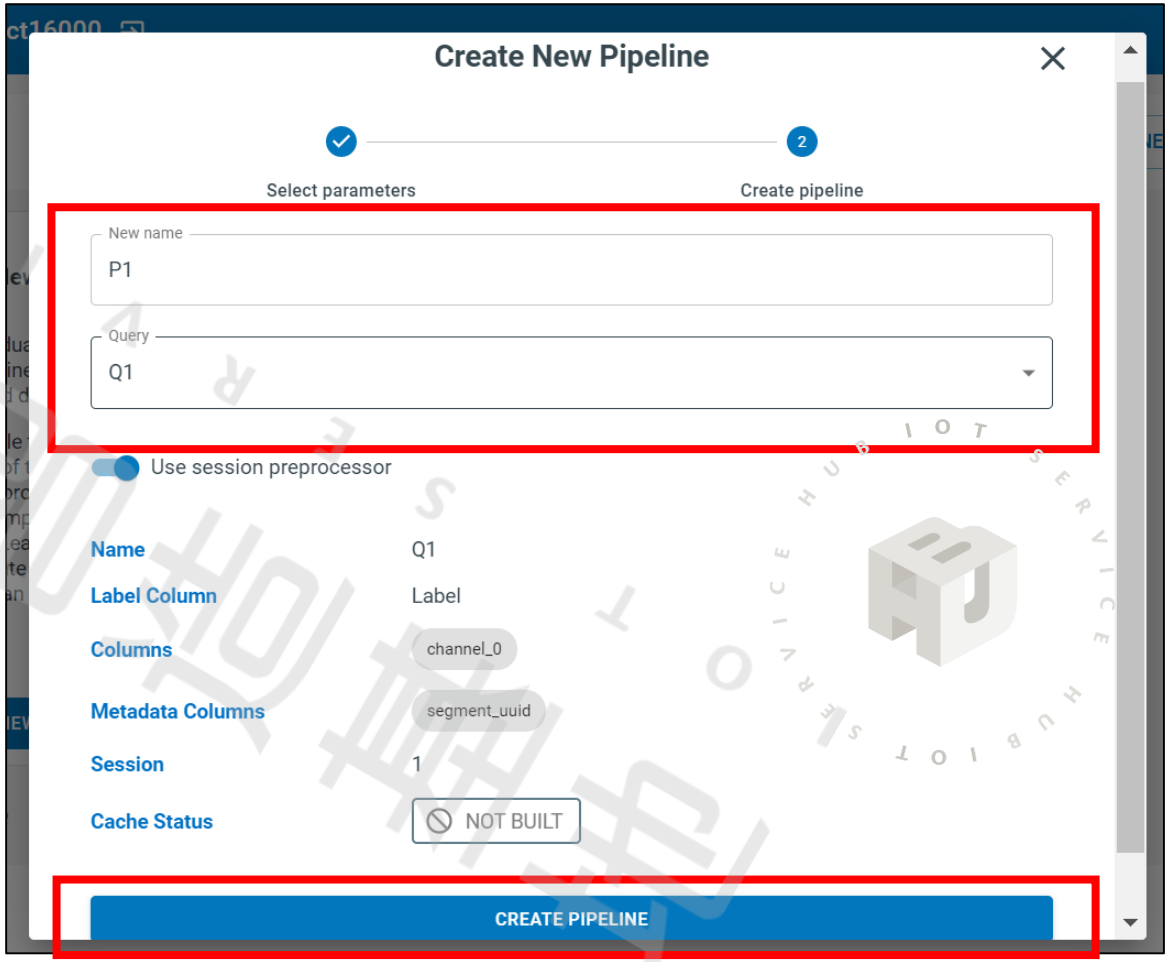


模型建立

跳出資訊欄，按Next

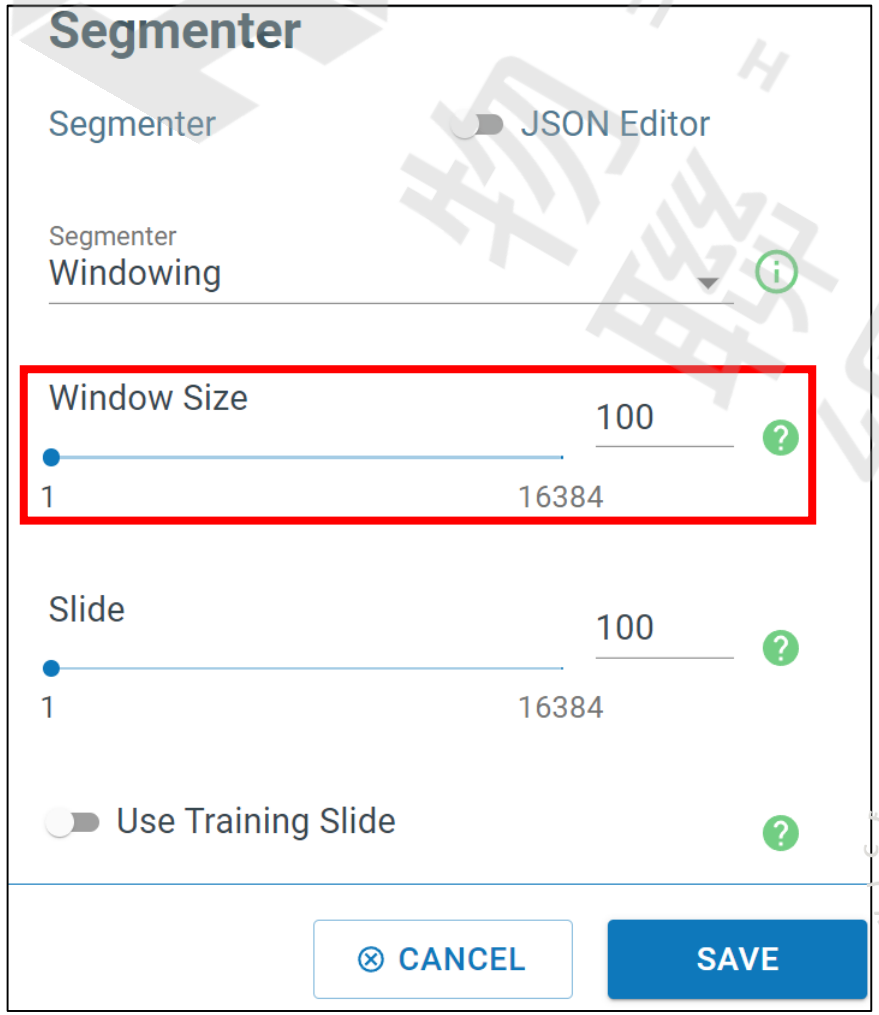


輸入pipeline的名稱(自訂)，選擇Query，完成後點選Create pipeline

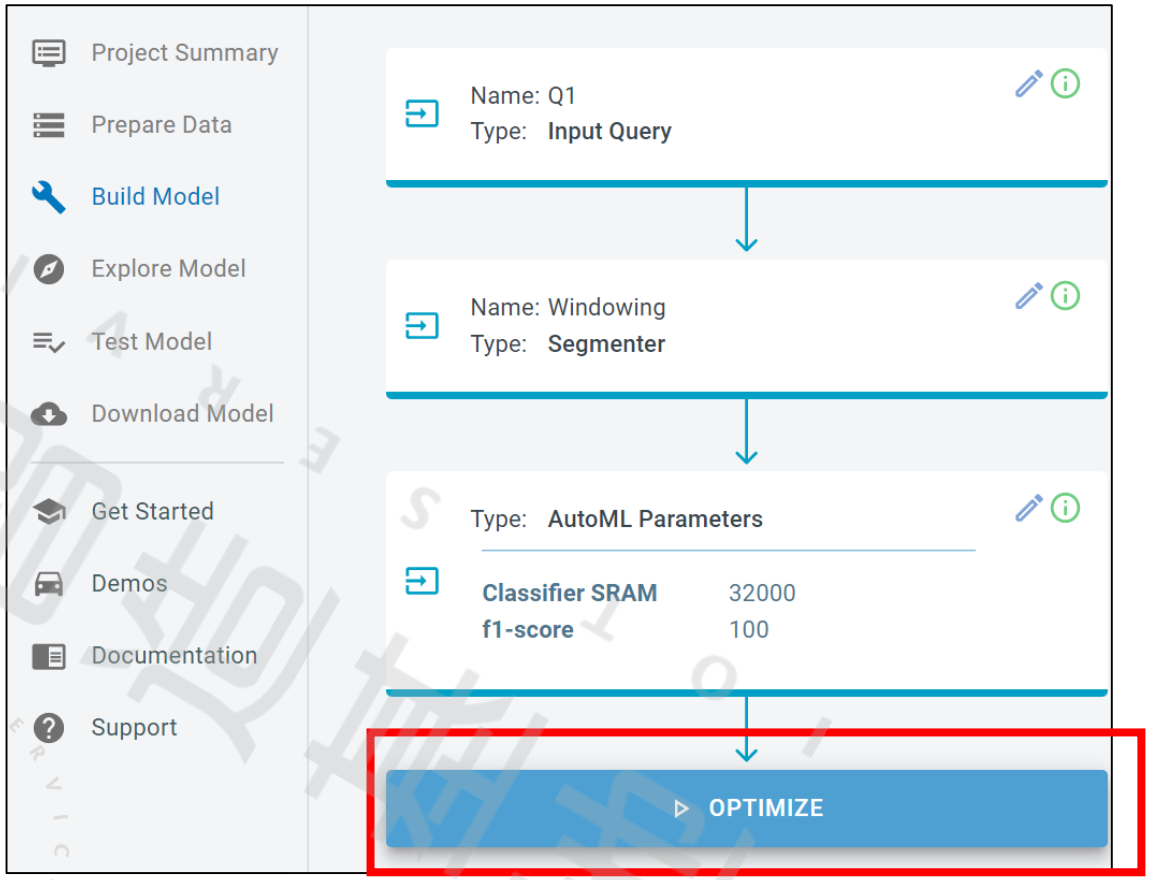


模型建立

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



設定完成後點選OPTIMIZE開始建構模型



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

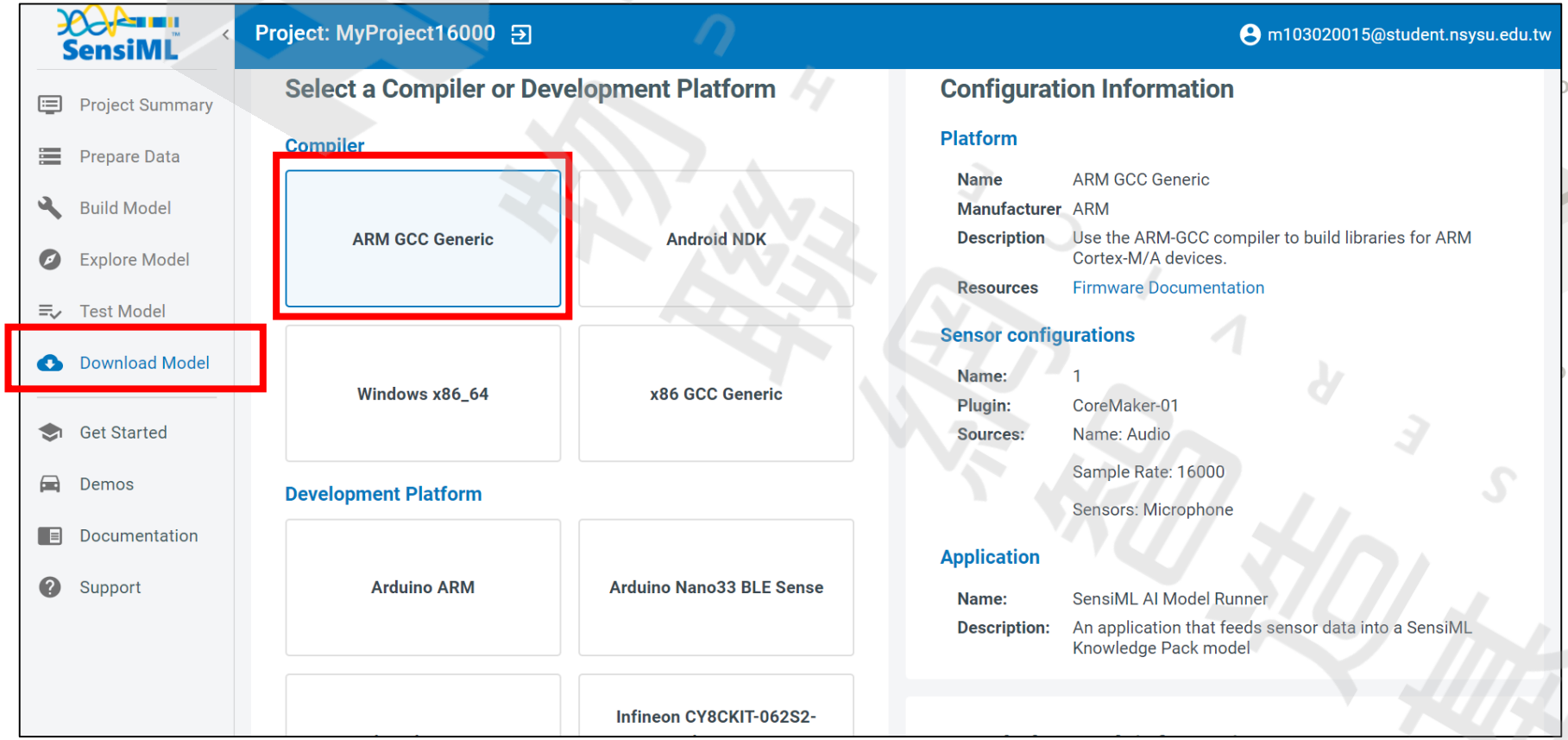
CHANGE PROJECT

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-aee6-984452ee2118					
	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

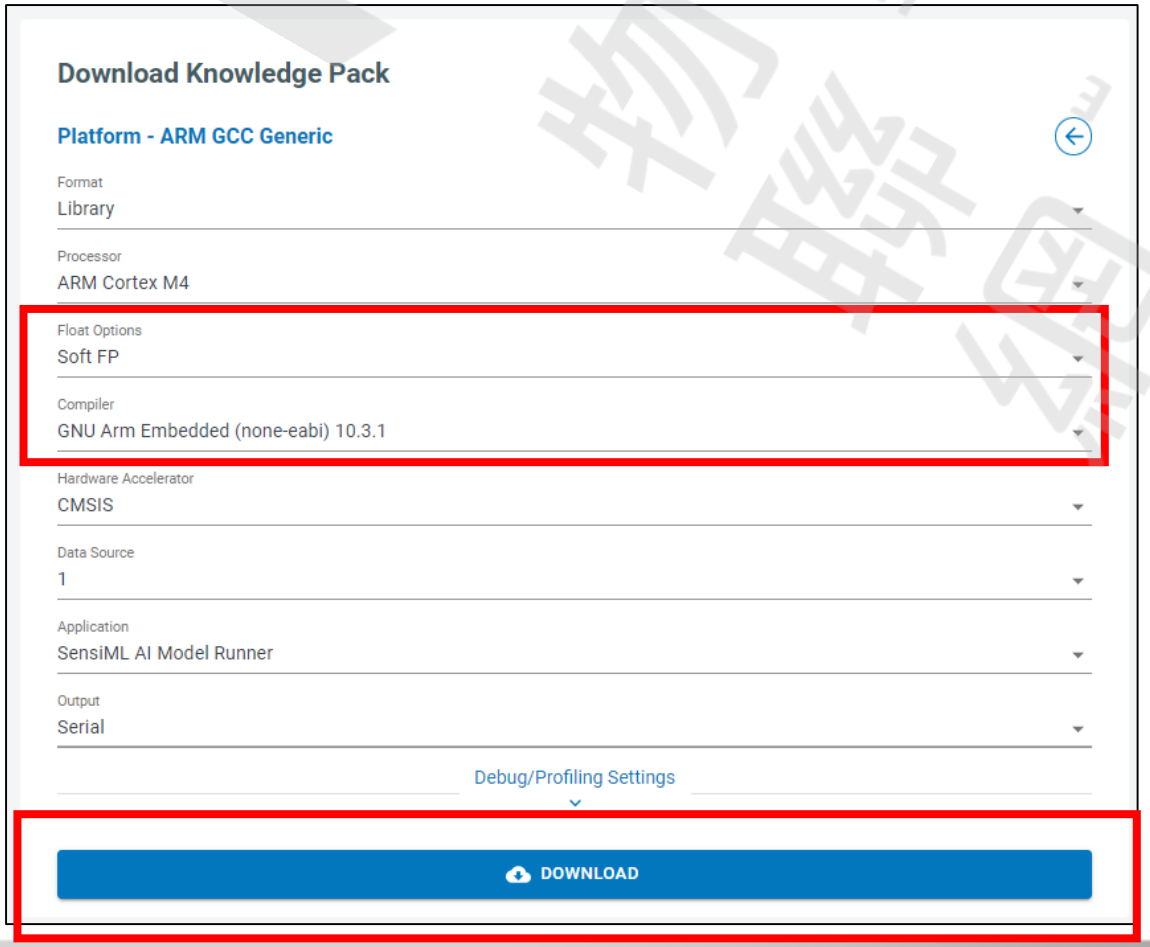
模型下載



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。

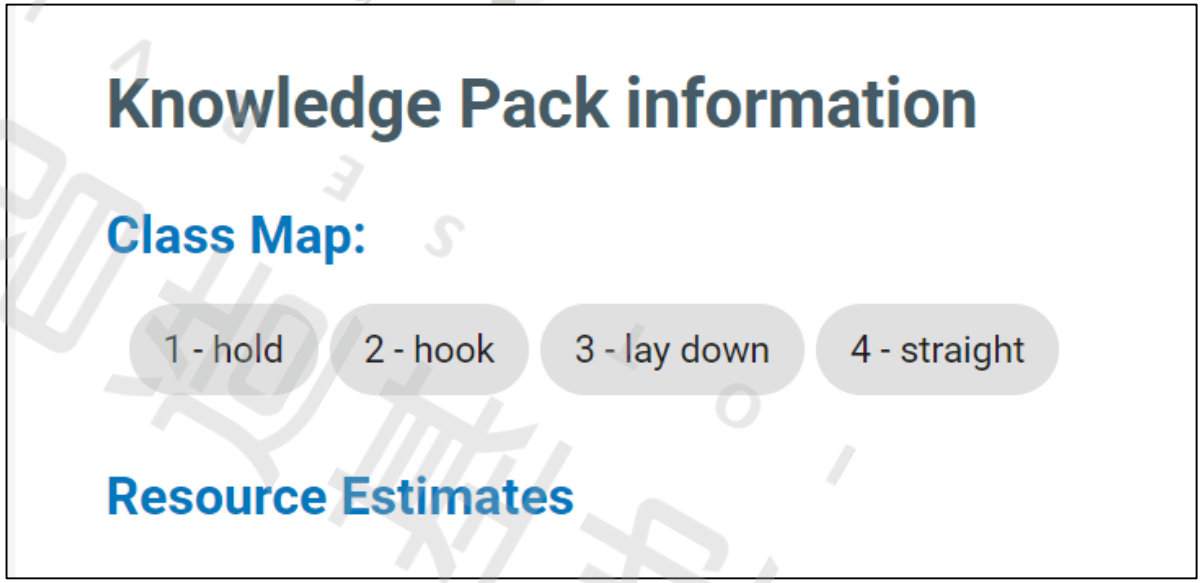
模型下載

更改部分:
Float options → Soft FP
Compiler → GNU 10.3.1
按下Download下載模型的壓縮檔



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



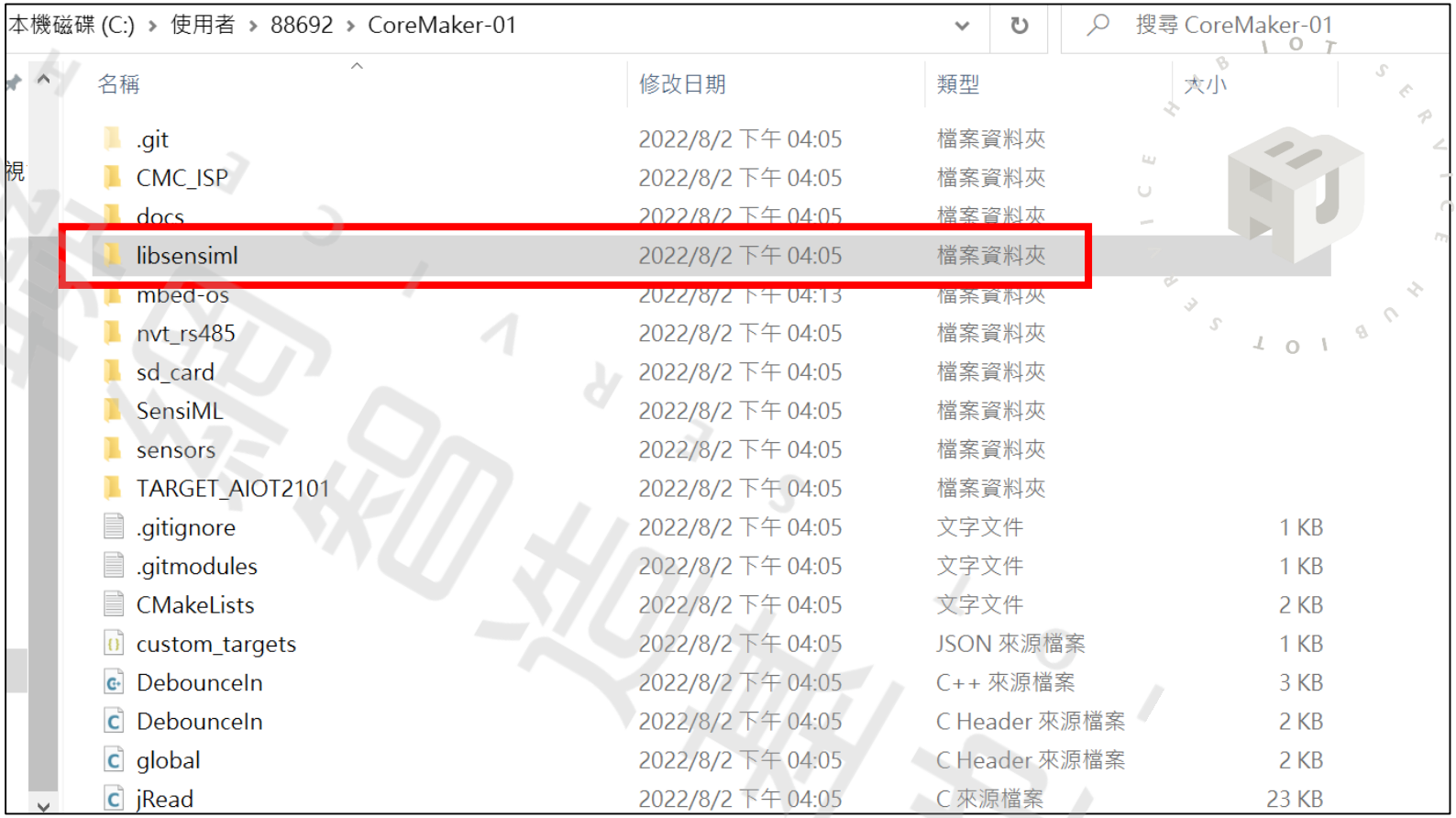
模型編譯

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



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.cp

```
66      * The arrangement must be the same wit
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
        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);
    }
```

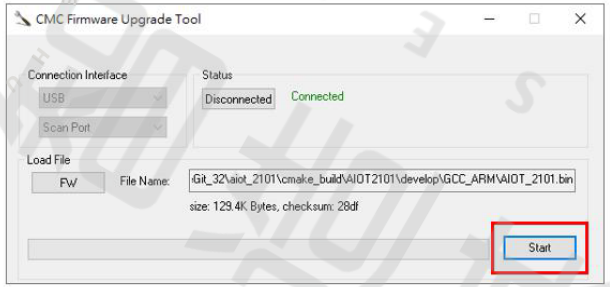
編譯與燒錄韌體

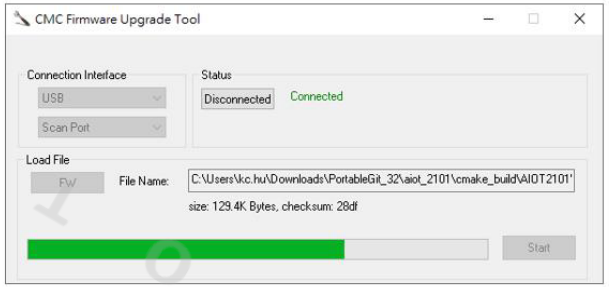
- 輸入“mbed-tools compile -m AIOT2101 -t GCC_ARM”開始編譯
- 使用CMC_ISP.exe透過USB進行燒錄下載bin檔至CoreMaker01開發板上

Step1執行編譯指令

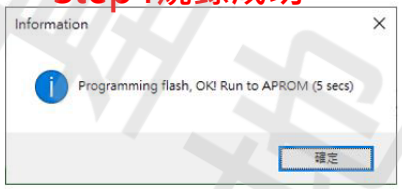
Step2編譯出bin檔

Step3連接開發板



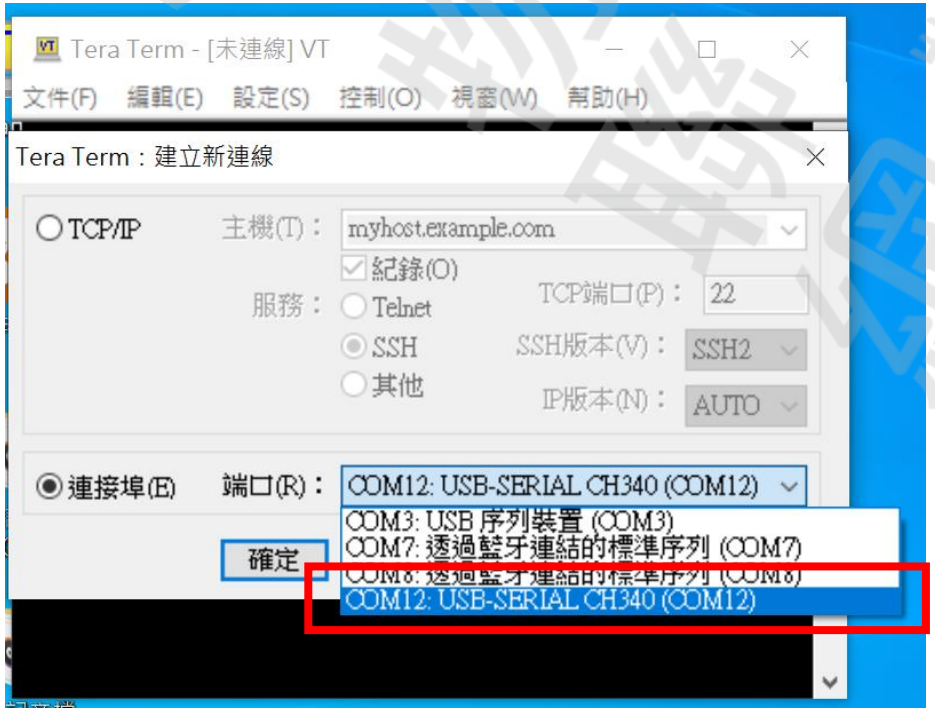


Step4燒錄成功



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)

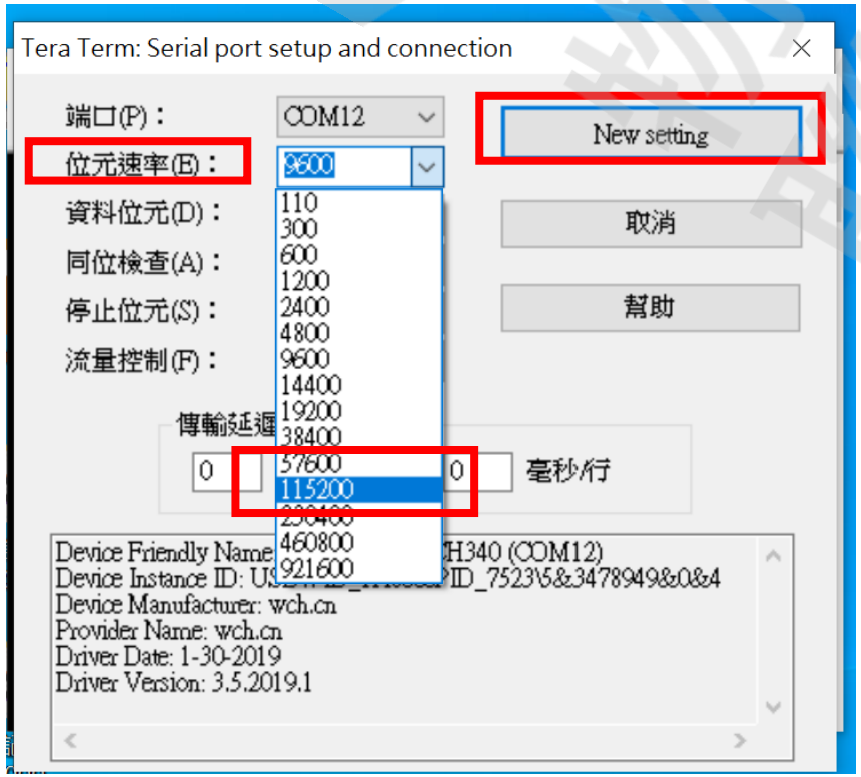


選擇連接的序列埠

選擇設定下的序列埠

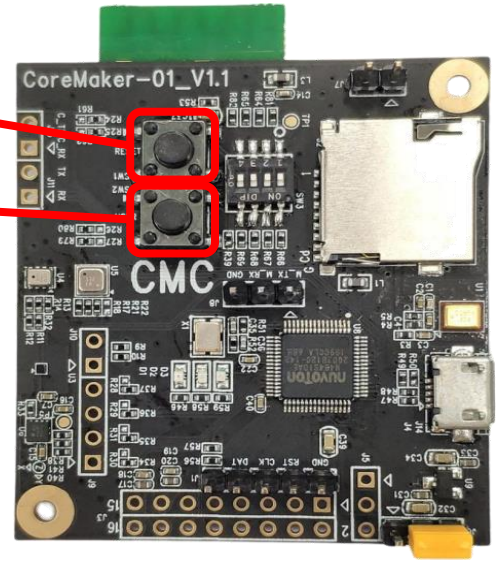
顯示輸出結果

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。



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

物聯網智造基地
IOT SERVICE HUB