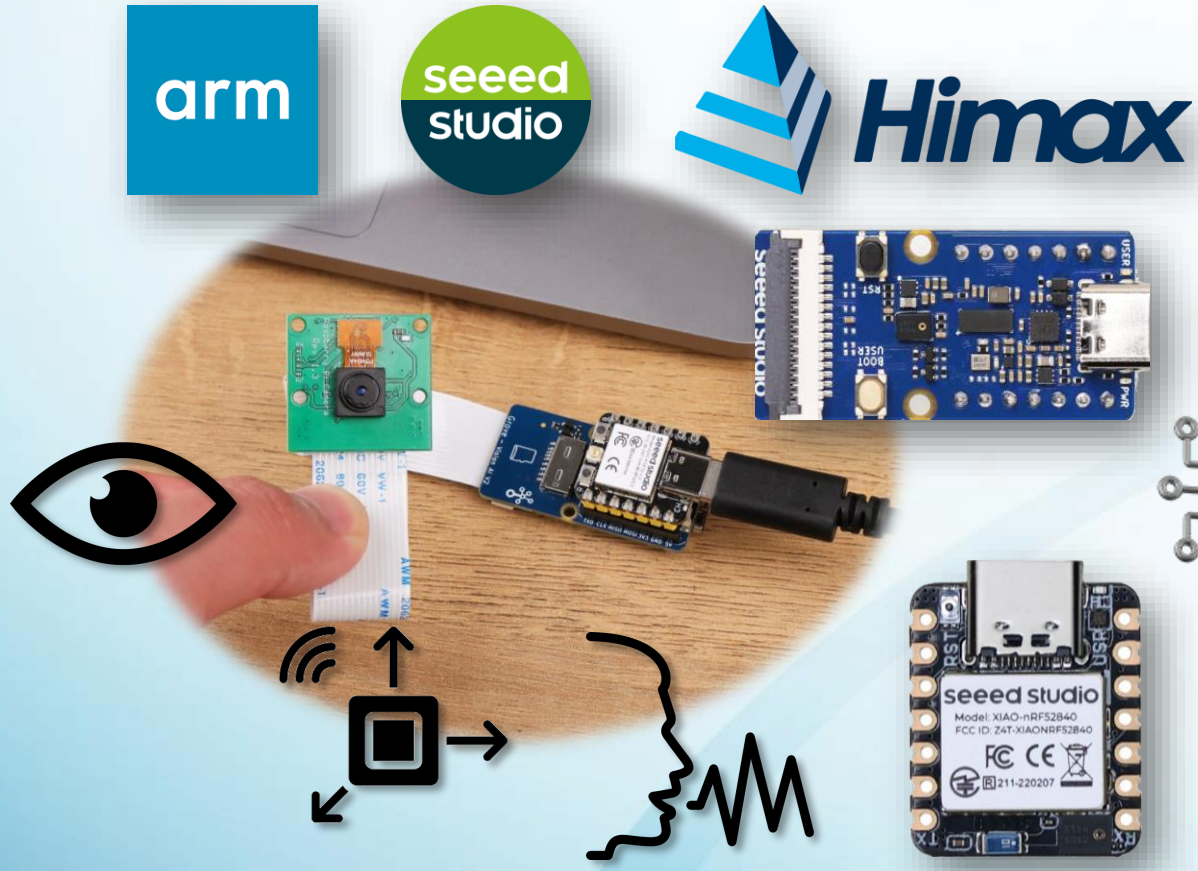
OmniXRI TinyML 小學堂 2025



Cortex-M Processor

Ethos-U MicroNPU

【第 9 講】

運動感測器應用—手勢辨識

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

簡報大綱

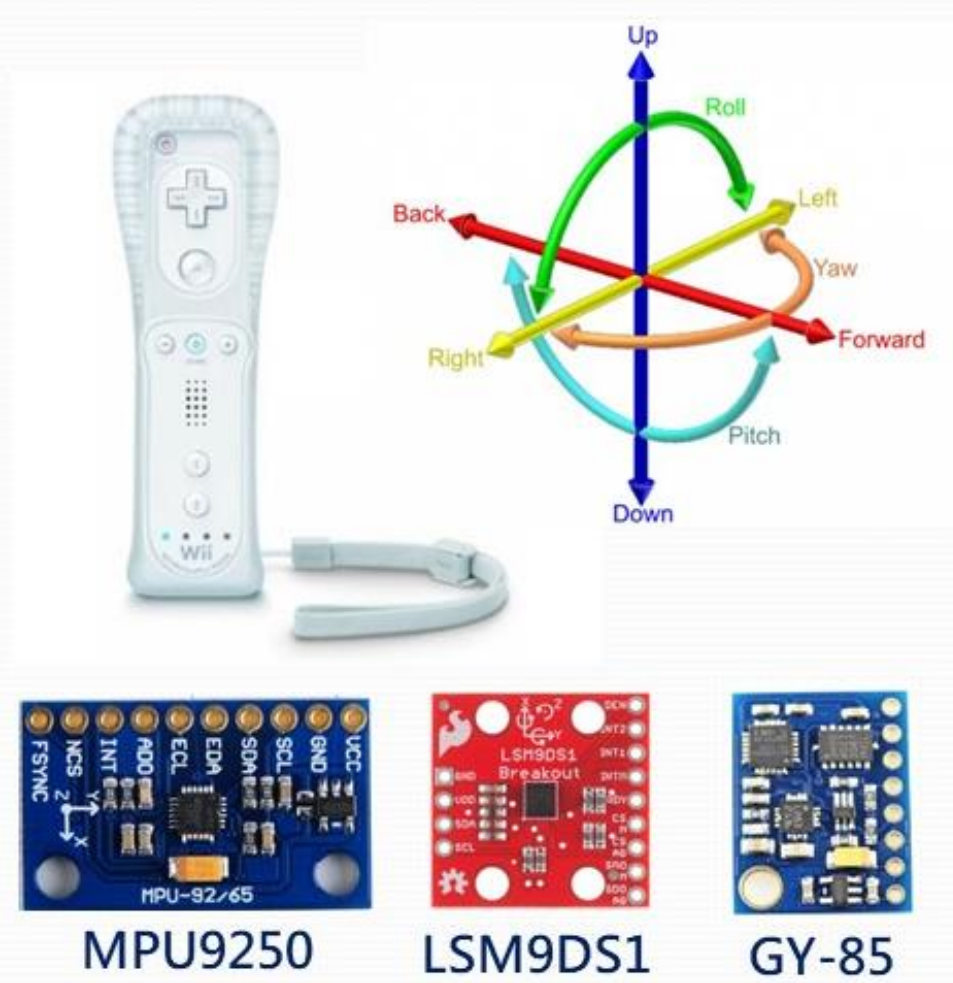
- 9.1. 運動資料集建置
- 9.2. **Edge Impulse** 開發環境建置
- 9.3. 模型選用與訓練
- 9.4. 模型部署與測試

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

完整課程大綱： https://omnixri.blogspot.com/2025/03/omnixri-tinyml-2025-0.html
課程直播清單： https://www.youtube.com/@omnixri1784/streams

9.1. 運動資料集建置

何謂運動感測器



運動感測器（Motion Sensor)

- 加速度計（重力感測）
  Accelerometer (G-Sensor)
- 陀螺儀（旋轉量感測）
  Gyroscope(Gyro-Sensor)
- 地磁感測計（方向感測）
  Magnetic(M-Sensor)
- 自由度(Degree of Freedom)
  3 DoF, 6 DoF, 9 DoF
- 慣性量測單元（IMU）
  Inertial Measurement Unit

資料來源：https://omnixri.blogspot.com/2022/04/20220408.html

運動感測常見應用

人機互動
(Wii / PS Move 搖桿)

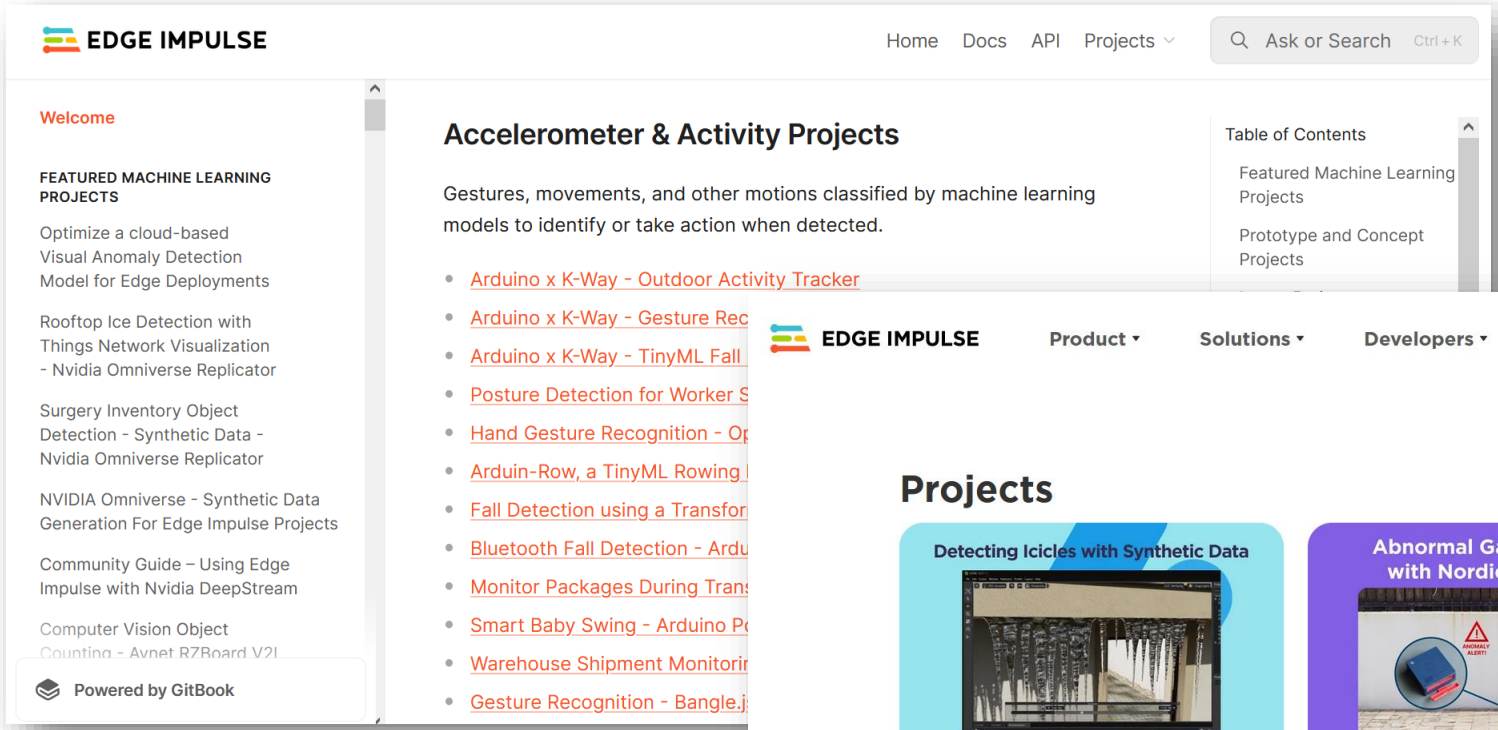
競速分析 / 駕駛習慣
(打龜號 / 車涯 04事件)

慣性導航 / 室內定位
(GPS無法看見天空)

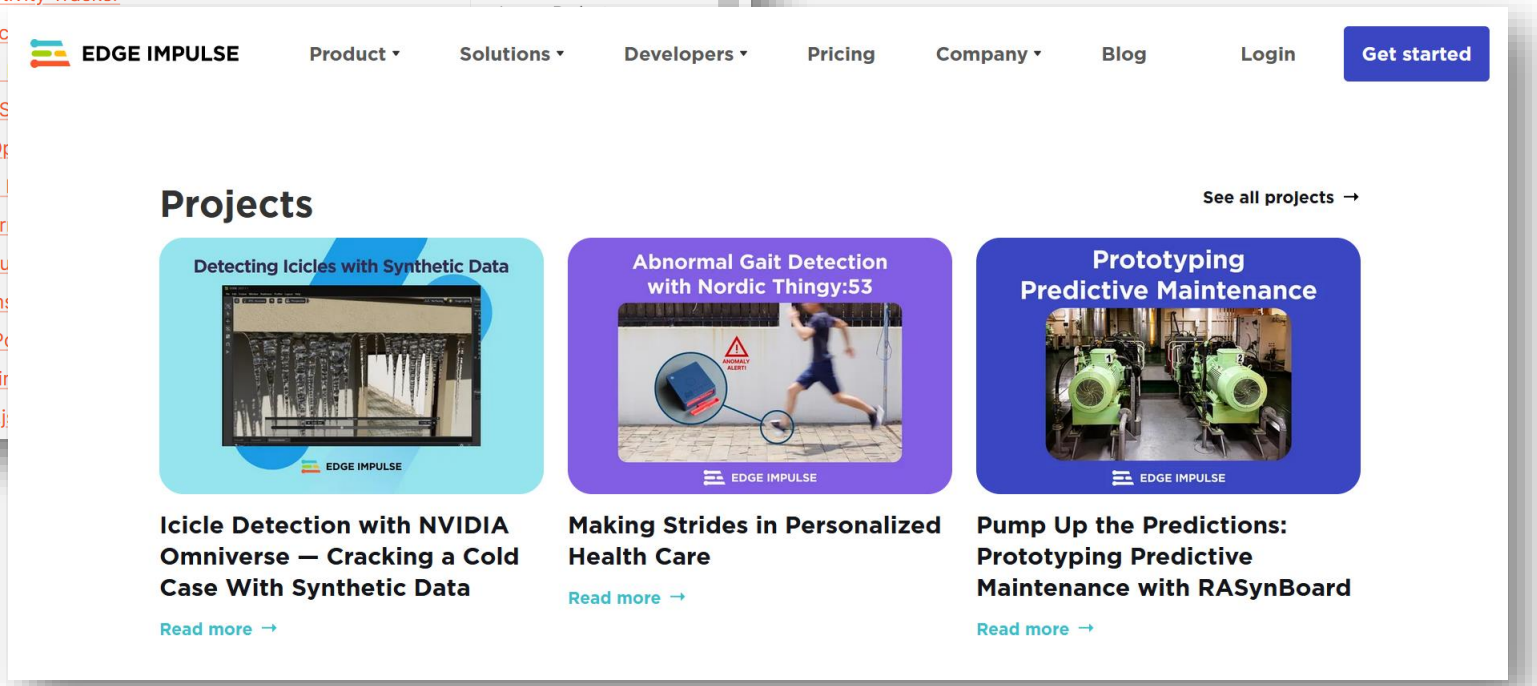
異常振動
(馬達/ 輸送帶)

自動平衡
(平衡車、無人機、人形機器人)

Edge Impulse 運動感測相關案例

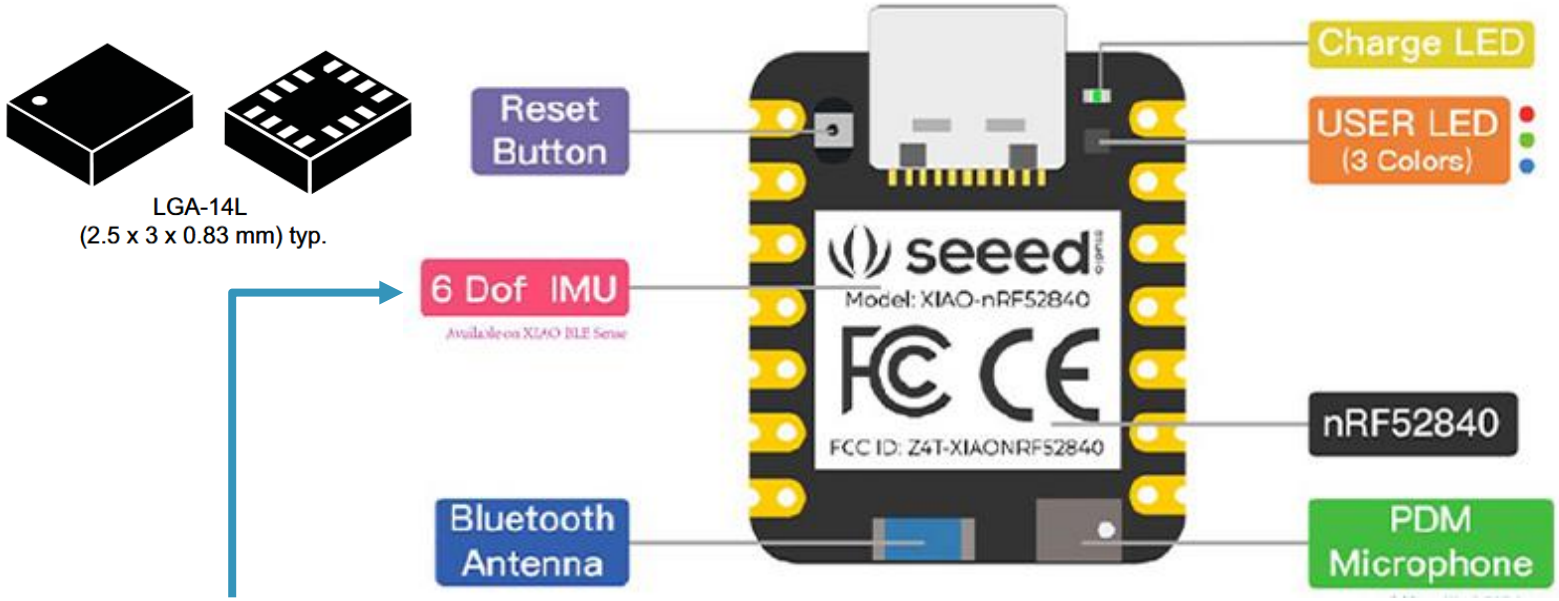


https://docs.edgeimpulse.com/experts#accelerometer-and-activity-projects



https://edgeimpulse.com/blog/

Seeed Xiao nRF52840 Sense

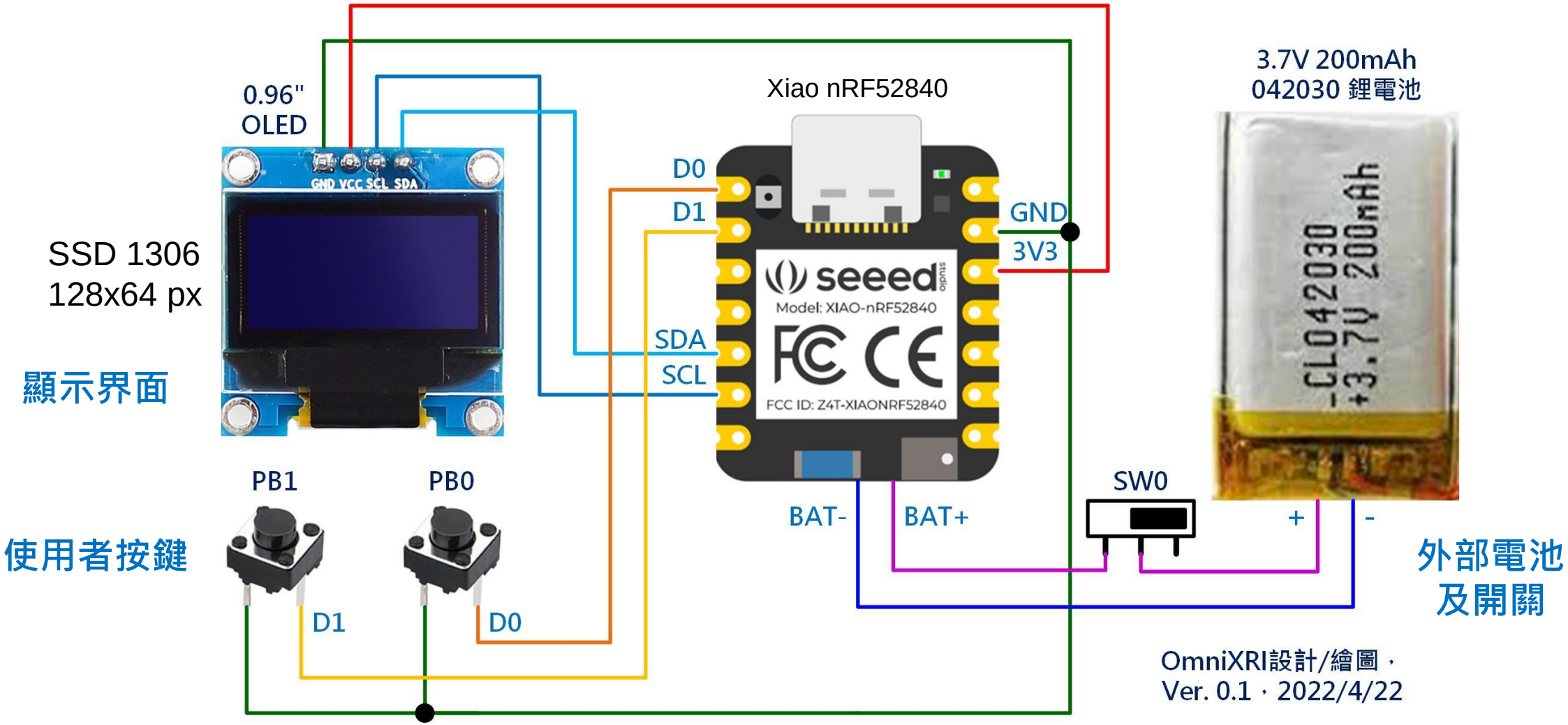


LSM6DS3TR-C

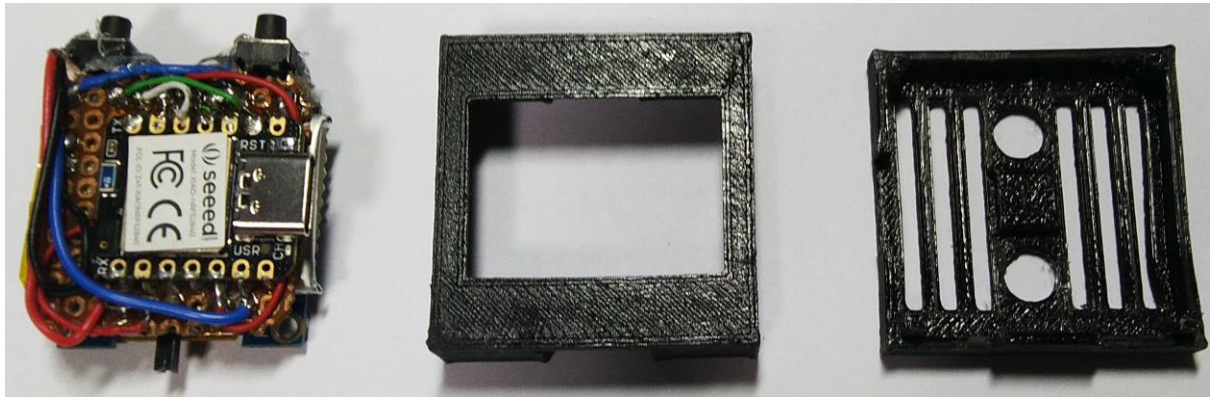
- 3D加速度計： ±2/±4/±8/±16 g full scale
- 3D陀螺儀： ±125/±250/±500/±1000/±2000 dps full scale
- 工作電壓：1.71 V to 3.6 V
- 通訊界面：SPI & I2 C serial interface

資料來源：https://wiki.seeedstudio.com/cn/XIAO_BLE/

Xiao nRF52840 Sense 模組 – 參考電路

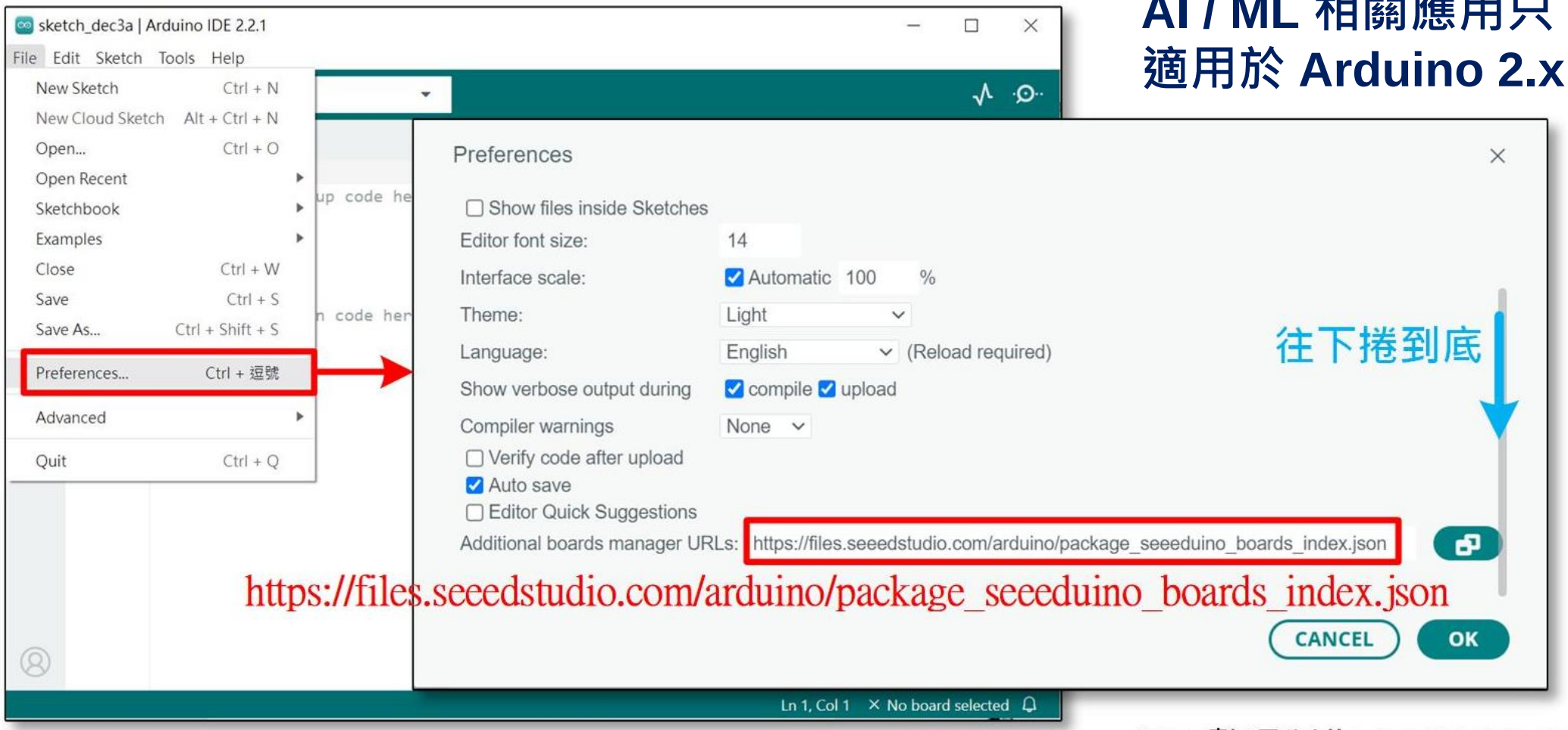


穿戴式智慧人工智慧裝置 – 參考外形



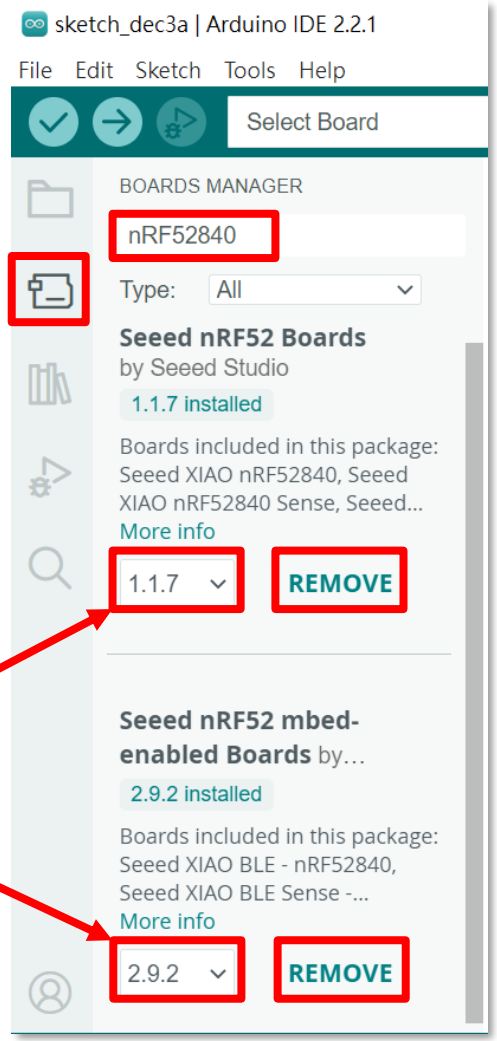
Arduino 新增開發板設定

AI / ML 相關應用只適用於 Arduino 2.x。



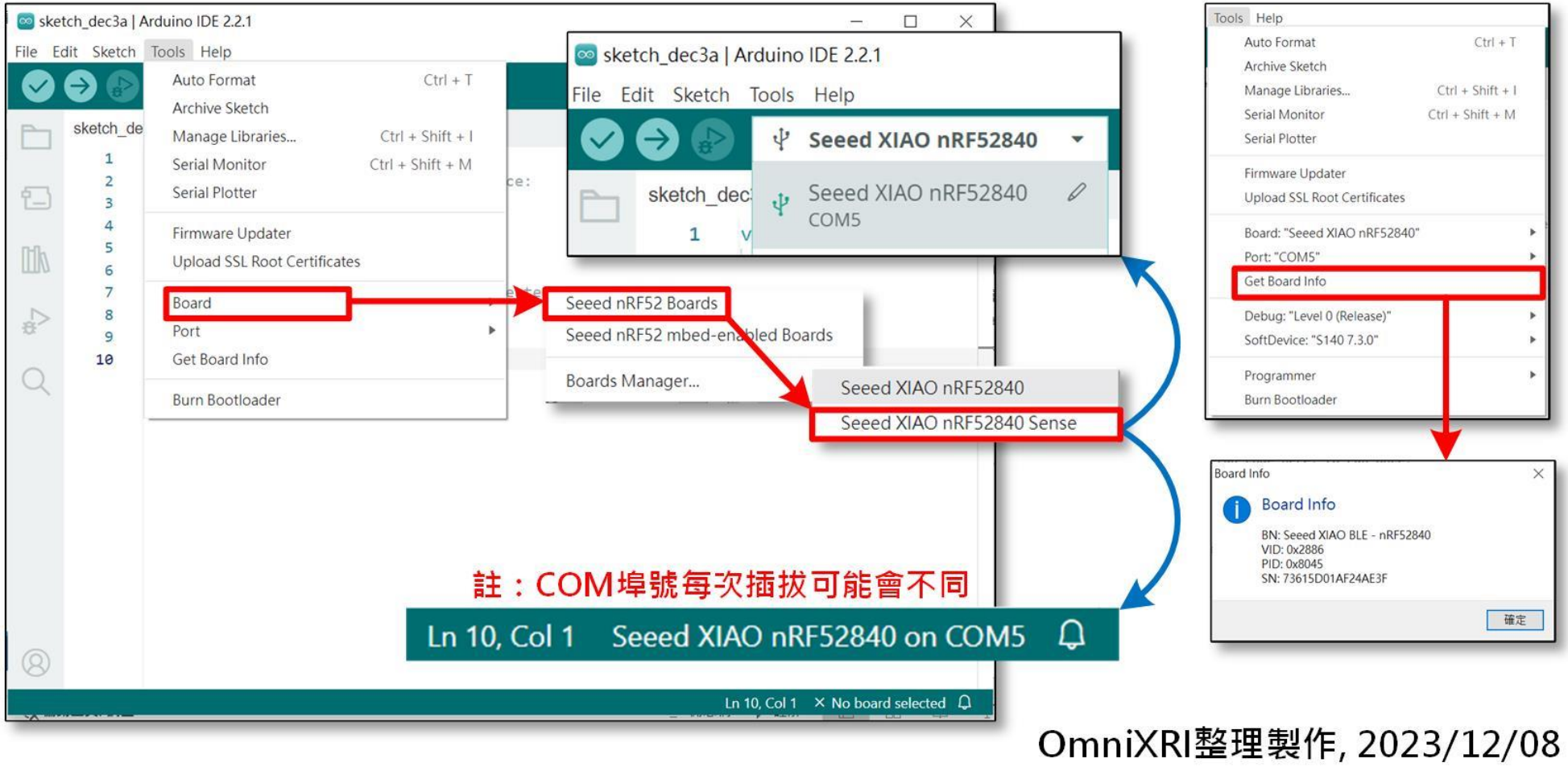
OmniXRI整理製作, 2023/12/08

安裝Seeed nRF52840函式庫



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

指定工作開發板及埠號

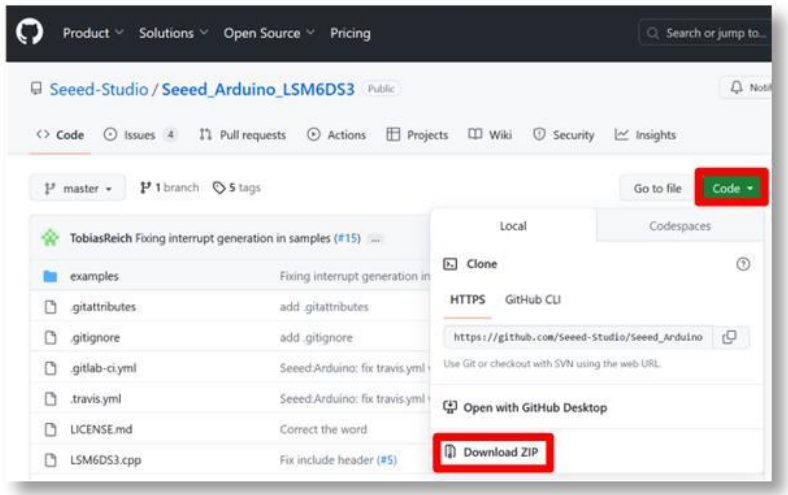


OmniXRI整理製作, 2023/12/08

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

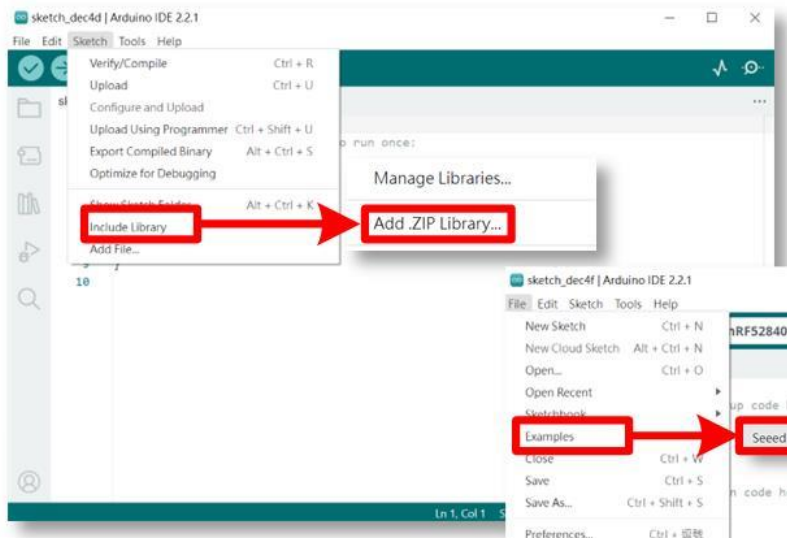
安裝運動感測器（IMU）函式庫

https://github.com/Seeed-Studio/Seeed_Arduino_LSM6DS3



Seeed_Arduino_LSM6DS3-master.zip

下載運動感測器函式庫



安裝函式庫

開啟範例程式

https://github.com/Seeed-Studio/Seeed_Arduino_LSM6DS3

OmniXRI整理製作, 2023/12/08

運動感測器（IMU）取值範例

File > Examples > Seeed Aruino LSM6DS3 > IMU_capture

```
#include <LSM6DS3.h>
#include <Wire.h>
LSM6DS3 myIMU(I2C_MODE, 0x6A); // 建立LSM6DS3實例並設定I2C裝置位址
float aX, aY, aZ, gX, gY, gZ; // 儲存三軸加速度及陀螺儀數值變數
const float accelerationThreshold = 2.5; // 加速度門檻值
const int numSamples = 119; // 樣本數量
int samplesRead = numSamples; // 讀取樣本數量
void setup() { // 初始化設定
  Serial.begin(9600); // 設定串例通信速度(bps)
  while (!Serial); // 等待串列埠初始化完成
  if (myIMU.begin() != 0) { // 初始化運動感測器
    Serial.println("Device error");
  } else {
    Serial.println("aX,aY,aZ,gX,gY,gZ");
  }
}
void loop() { // 無限迴圈
  // 等待取得足夠數量樣本
  while (samplesRead == numSamples) {
    // 讀取加速度計值
    aX = myIMU.readFloatAccelX();
    aY = myIMU.readFloatAccelY();
    aZ = myIMU.readFloatAccelZ();
    // 求得加速度值總合
    float aSum = fabs(aX) + fabs(aY) + fabs(aZ);
    // 檢查是否超過門檻值，若是重置已讀樣本數量
    if (aSum >= accelerationThreshold) {
      samplesRead = 0;
      break;
    }
  }
  // 檢查自上次檢測到顯著運動以來是否已讀取所有必需的樣本
  while (samplesRead < numSamples) {
    samplesRead++; // 已讀取樣本數加1
    // print the data in CSV format
    Serial.print(myIMU.readFloatAccelX(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatAccelY(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatAccelZ(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroX(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroY(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroZ(), 3);
    Serial.println();
    // 若已讀樣本數已足夠則列印換行
    if (samplesRead == numSamples) {
      Serial.println();
    }
  }
}
```

觸發值大於門檻才輸出

練習：將板子以不同方向移動及旋轉測試讀值是否正常

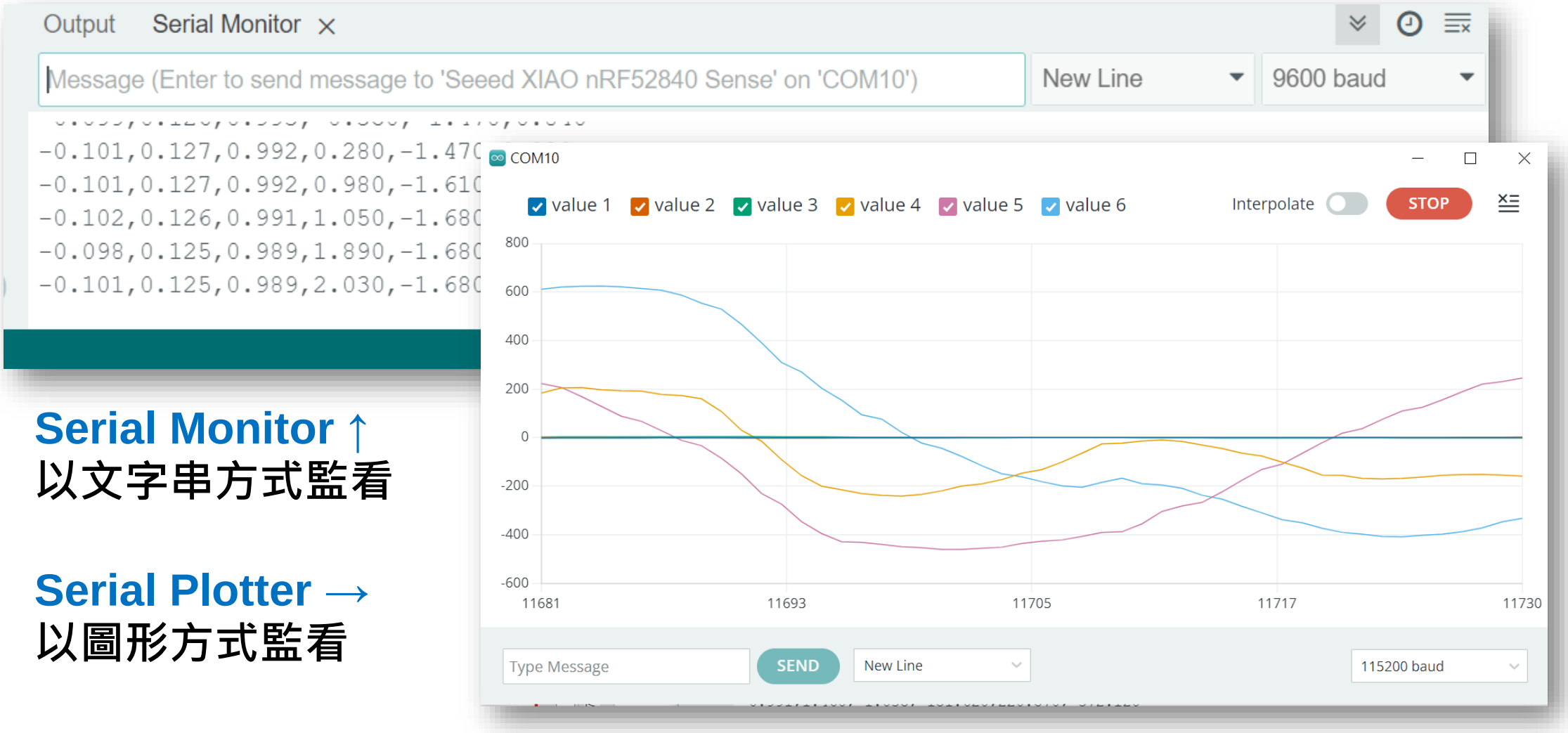
IMU 連續取值輸出至 Edge Impulse

```ino
#include <LSM6DS3.h>
#include <Wire.h>

//Create a instance of class LSM6DS3
LSM6DS3 myIMU(I2C_MODE, 0x6A);    //I2C device address 0x6A
float aX, aY, aZ, gX, gY, gZ;
// const float accelerationThreshold = 2.5; // threshold of significant in G's
// const int numSamples = 119;
// int samplesRead = numSamples;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  while (!Serial);
  //Call .begin() to configure the IMUs
  if (myIMU.begin() != 0) {
    Serial.println("Device error");
  } else {
    Serial.println("aX,aY,aZ,gX,gY,gZ");
  }
}

void loop() {
  // wait for significant motion
//   while (samplesRead == numSamples) {
//     // read the acceleration data
//     aX = myIMU.readFloatAccelX();
//     aY = myIMU.readFloatAccelY();
//     aZ = myIMU.readFloatAccelZ();

//     // sum up the absolutes
//     float aSum = fabs(aX) + fabs(aY) + fabs(aZ);

//     // check if it's above the threshold
//     if (aSum >= accelerationThreshold) {
//       // reset the sample read count
//       samplesRead = 0;
//       break;
//     }
//   }

//   // check if the all the required samples have been read since
//   // the last time the significant motion was detected
//   while (samplesRead < numSamples) {
//     // check if both new acceleration and gyroscope data is
//     // available
//     // read the acceleration and gyroscope data

//     samplesRead++;

    // print the data in CSV format
    Serial.print(myIMU.readFloatAccelX(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatAccelY(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatAccelZ(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroX(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroY(), 3);
    Serial.print(',');
    Serial.print(myIMU.readFloatGyroZ(), 3);
    Serial.println();

//     if (samplesRead == numSamples) {
//       // add an empty line if it's the last sample
//       Serial.println();
//     }
//   }
}
```

註解或刪除紅色框區域

連續取值並輸出至 COM 搭配 edge-impulse-data-forwarder 可將感測器值送到雲端

監看運動感測器（IMU）輸出值



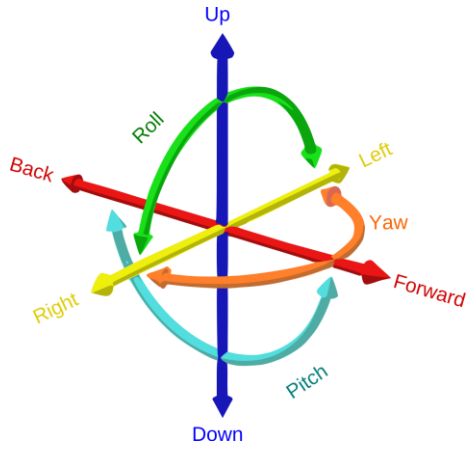
Serial Monitor ↑
以文字串方式監看

Serial Plotter →
以圖形方式監看

設計自定義操作手勢

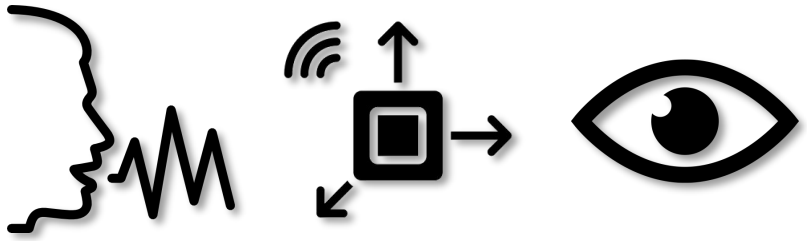
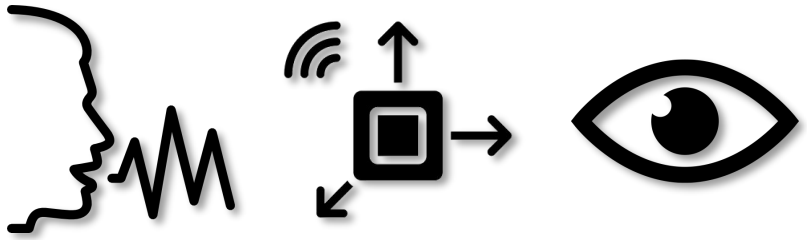
- Start – 向前張開手掌
- Next – 由左向右快速平移並有頓點
- Prev – 由下向上擺動並有頓點
- Stop – 雙手快速交叉於胸口
- Unknow – 任意輕微擺動或不動，可為任意姿勢。

- **運動感測器必須固定在手部特定位置以免相同動作但感測值差異過大**
- **動作設定最好於不同軸向有明顯差異可只用加速度計而不使用陀螺儀**



9.2. Edge Impulse 開發環境建置

TinyML 開發流程選項

模型訓練工具名稱	適用感測器	主要特色
TensorFlow Lite (現更名LiteRT)		免費版：資料集需自行建置，僅提供模型訓練及轉成 TensorFlow Lite for Microcontroller 格式。可利用雲端 Google Colab 或本地端 Jupyter Notebooks 進行模型訓練及部署。
EDGE IMPULSE		免費版：可支援各式感測器及開發板，提供一定額度資料集建置、資料前處理（特徵提取）模型訓練、模型優化及多種部署方式。亦可導入已訓練好的 TensorFlow Lite, ONNX 等模型。
SenseCraft		免費版：影像分類可多分類、物件偵測同一模型僅能一類。事件觸發只能設一種條件。可支援雲端資料收集、模型訓練及部署，亦導入外部已訓練好之 TensorFlow Lite 模型。

Edge Impulse 資料集建置方式

從裝置

EDGE IMPULSE

手機掃描 QR Code 建立裝置連結，可傳送聲音、運動及影像感測資料。

edge-impulse-daemon 開發板直送訊號

（限Edge Impulse 指定開發板）

edge-impulse-data-forward

開發板透過 COM 埠 直送文字串資料，適合連續、低速（2~400Hz）感測器

利用電腦上裝置直接將聲音、運動及影像感測資料上傳到雲端。

從檔案

可將 CBOR, JSON, CSV, WAV, JPG, PNG, AVI, MP4等格式檔案（夾）直接上傳作為資料集。

edge-impulse-uploader

資料集建置方式

OmniXRI整理製作, 2025/04/16

快速操作指令表

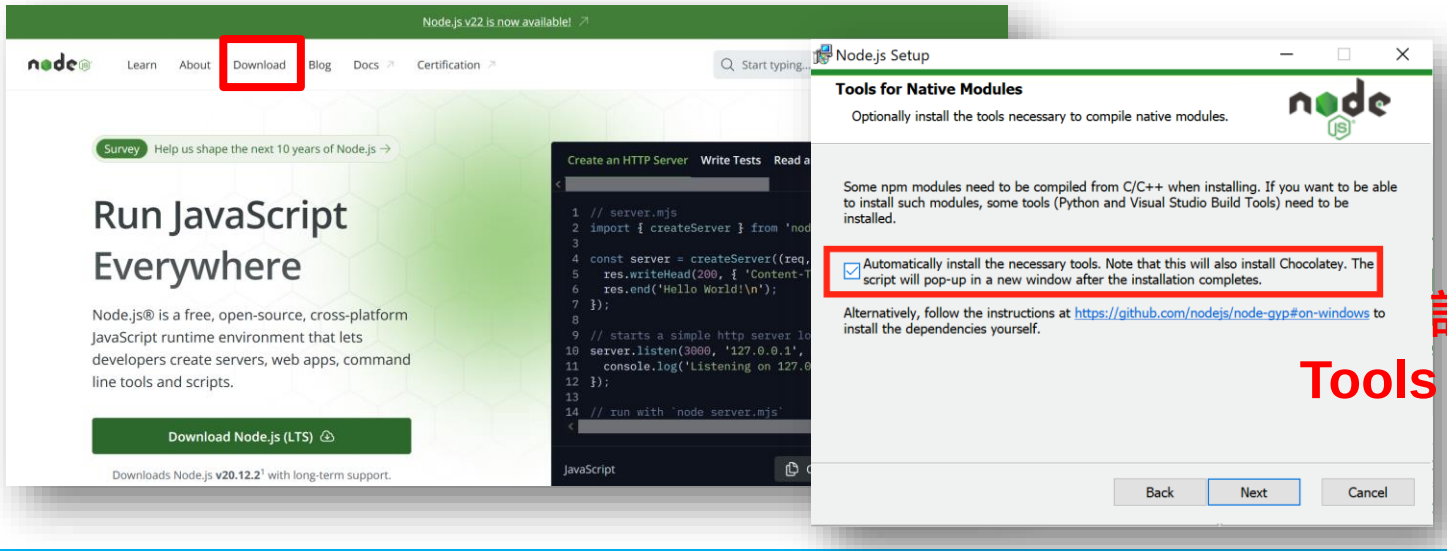
https://github.com/OmniXRI/Edge_AI_TinyML_Course_2024/blob/main/Ch13_Motion_Recognition/IMU_Quick_Guide.md

下載及安裝必要工具

1. [Python 3.x]

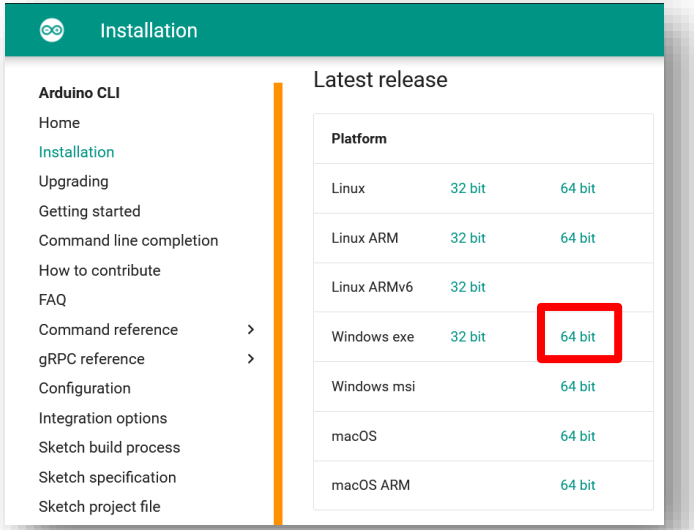
2. [Node.js]

(若後續 Edge Impulse 無法安裝 可換回舊版，例： v18.20.2 LTS)



3. [Arduino CLI]

(選配，依不同開發板需求)



安裝時要注意
記得一定要勾選
Tools for Native Modules

以 node -v 及 npm -v
檢查安裝好的版本

安裝 Edge Impulse windows 工作環境

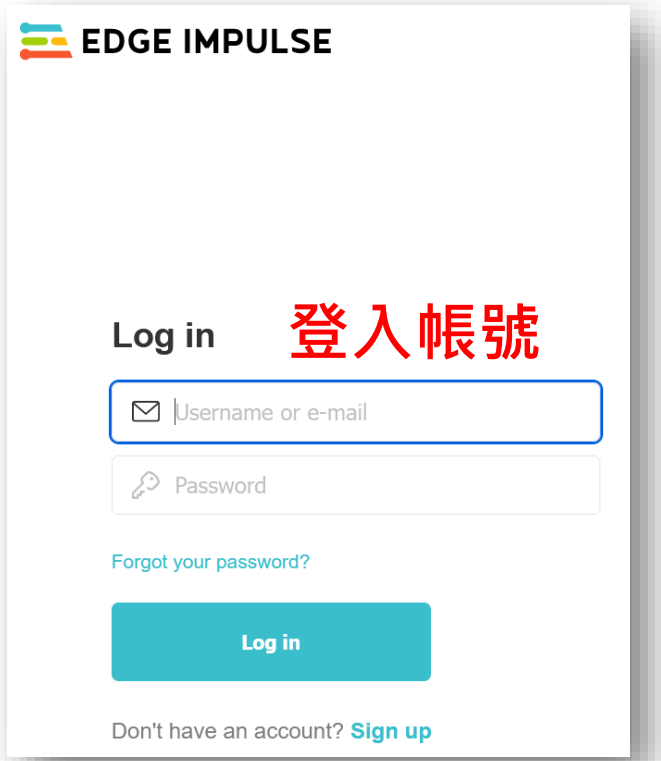
- 安裝**Edge Impulse CLI**

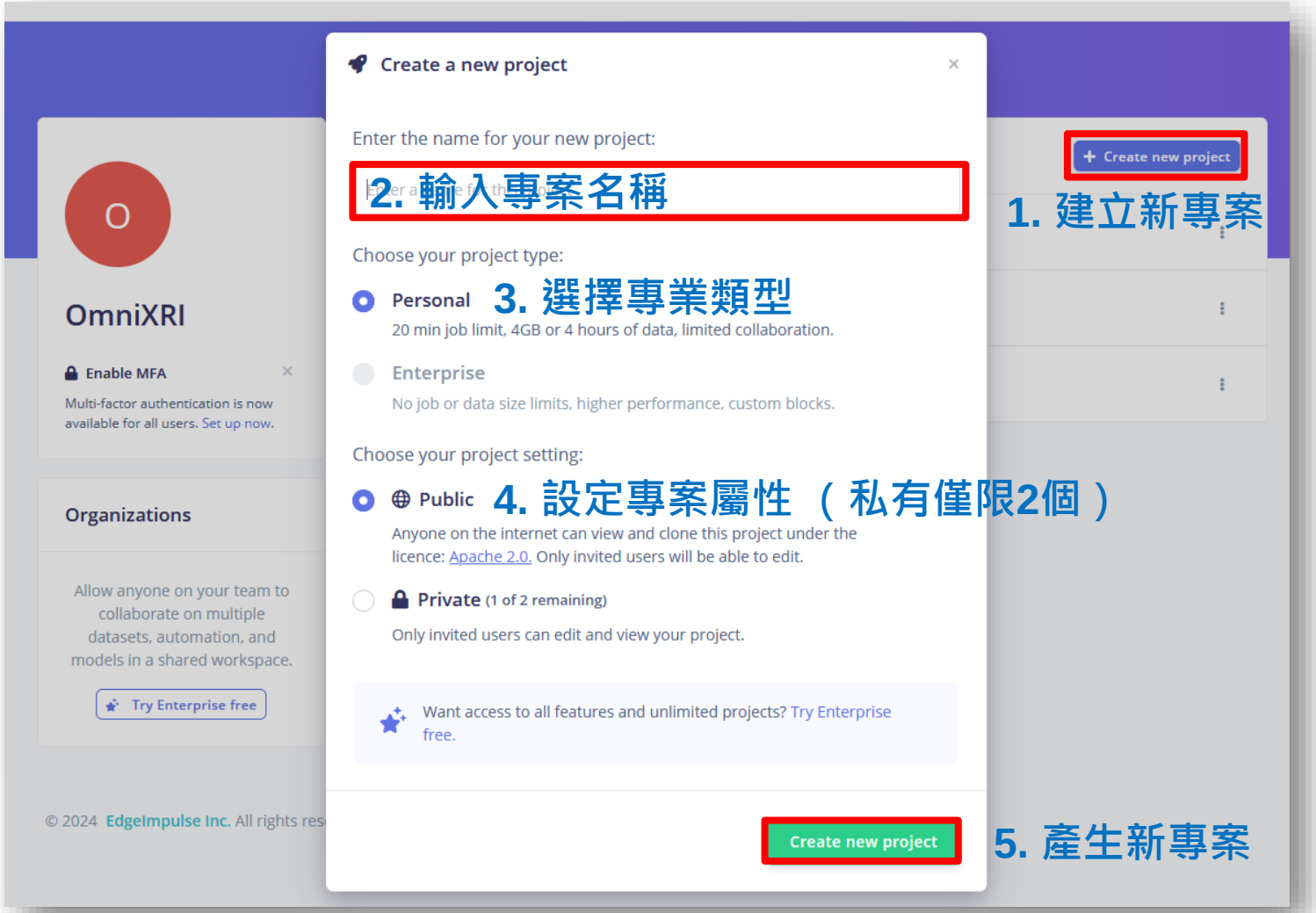
  npm install -g edge-impulse-cli --force

- 安裝完成後會得到下列工具
  - edge-impulse-daemon **透過 COM 埠和雲端平台連接**
  - edge-impulse-uploader 允許上傳本機檔案
  - edge-impulse-data-forwarder **透過 COM 埠上傳開發板感測器數值**
  - edge-impulse-run-impulse 顯示裝置上運行的模型
  - edge-impulse-blocks 建立自定義區塊
  - himax-flash-tool 燒錄 Haimax WE-I Plus

https://docs.edgeimpulse.com/docs/tools/edge-impulse-cli/cli-installation

建立 Edge Impulse 新專案



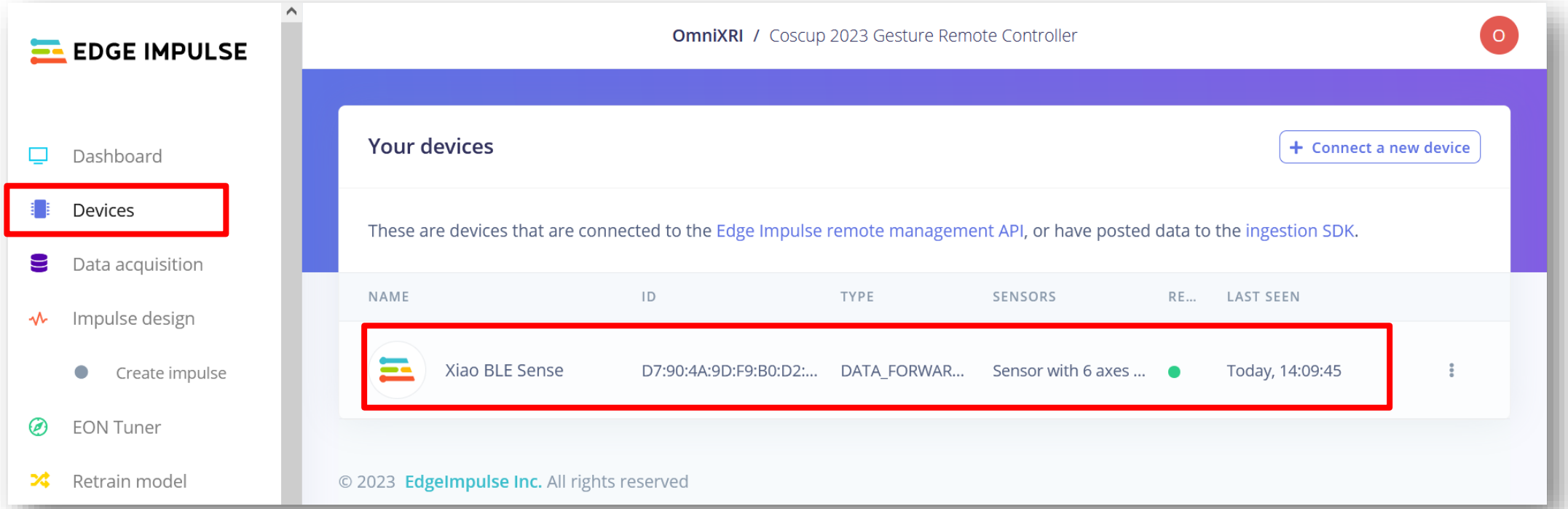


參考資料：https://wiki.seeedstudio.com/cn/XIAO-BLE-PDM-EI/

啟動edge-impulse-data-forwarder

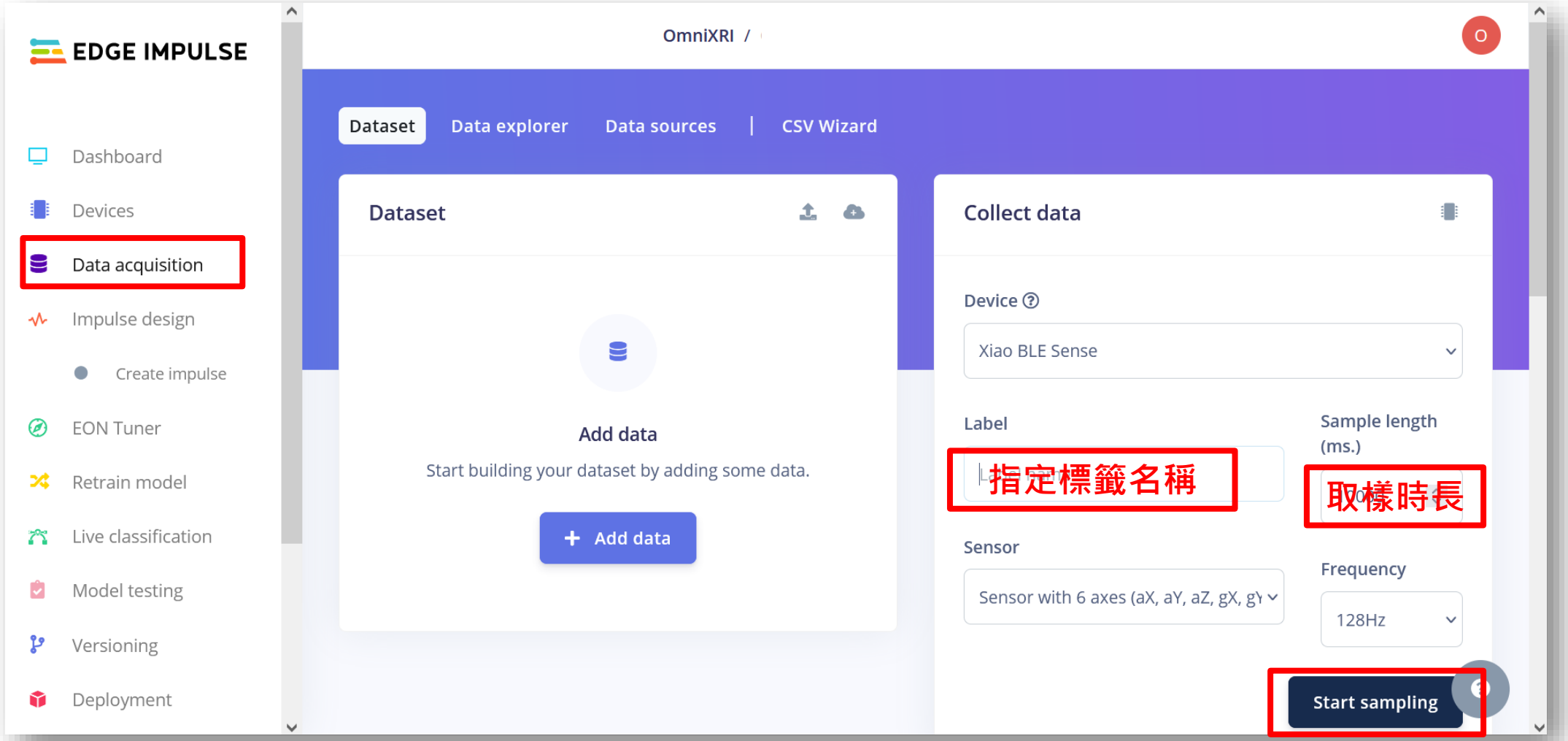
輸入帳號密碼並選定專案。準備接收來自開發板傳送的資料並給予並應的名稱。傳送的資料頻度必須大於2Hz。
若無法連接 COM 埠，請按開發板重置鍵一次，並請關閉 Arduino 及有可能佔用 COM 之程式。

檢查裝置是否已連線

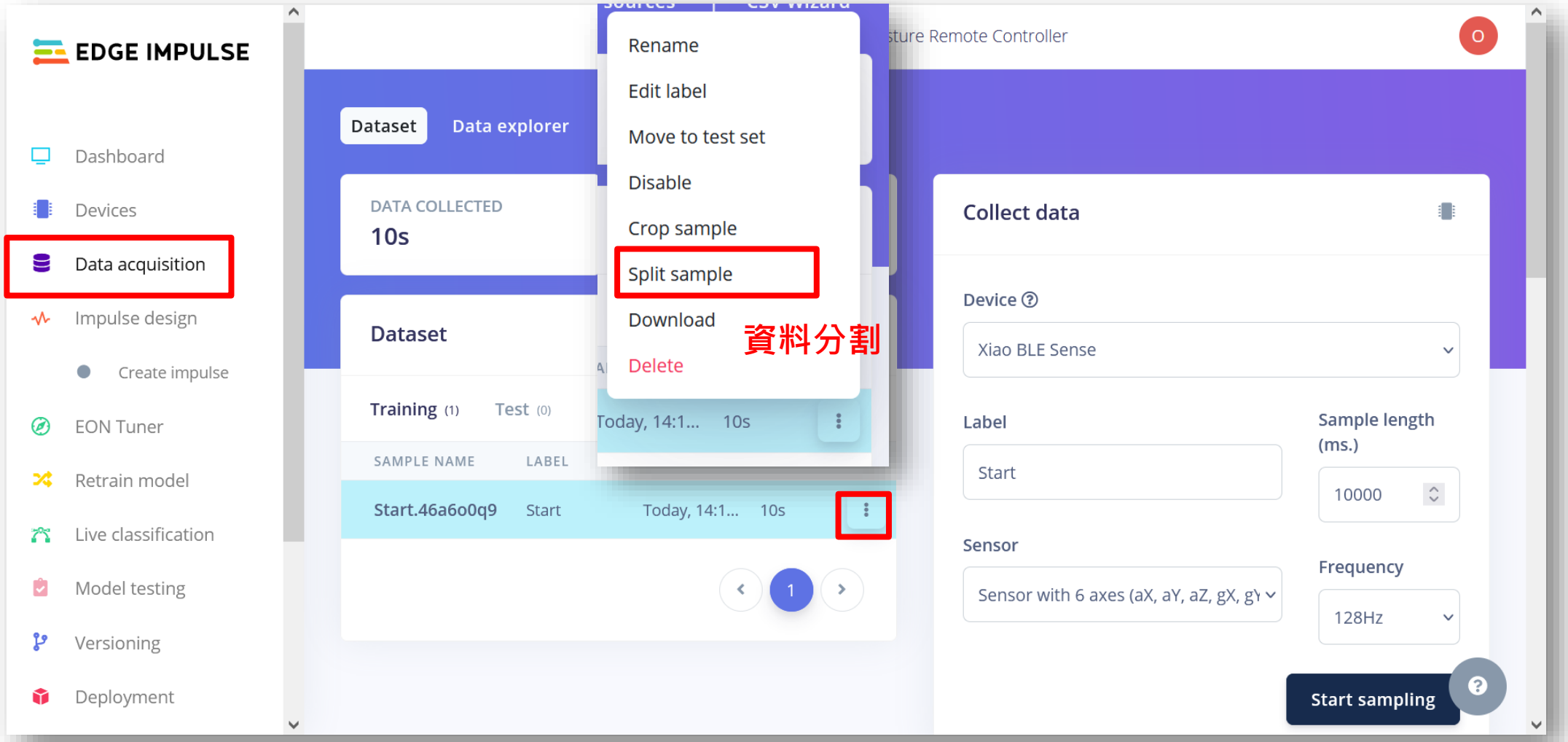


- **開發板運行IMU連續取值輸出程式。**
- **啟動 edge-impulse-data-forwarder ，將裝置連線到雲端。**
- **將開發板輸出之內容，傳送到雲端。**

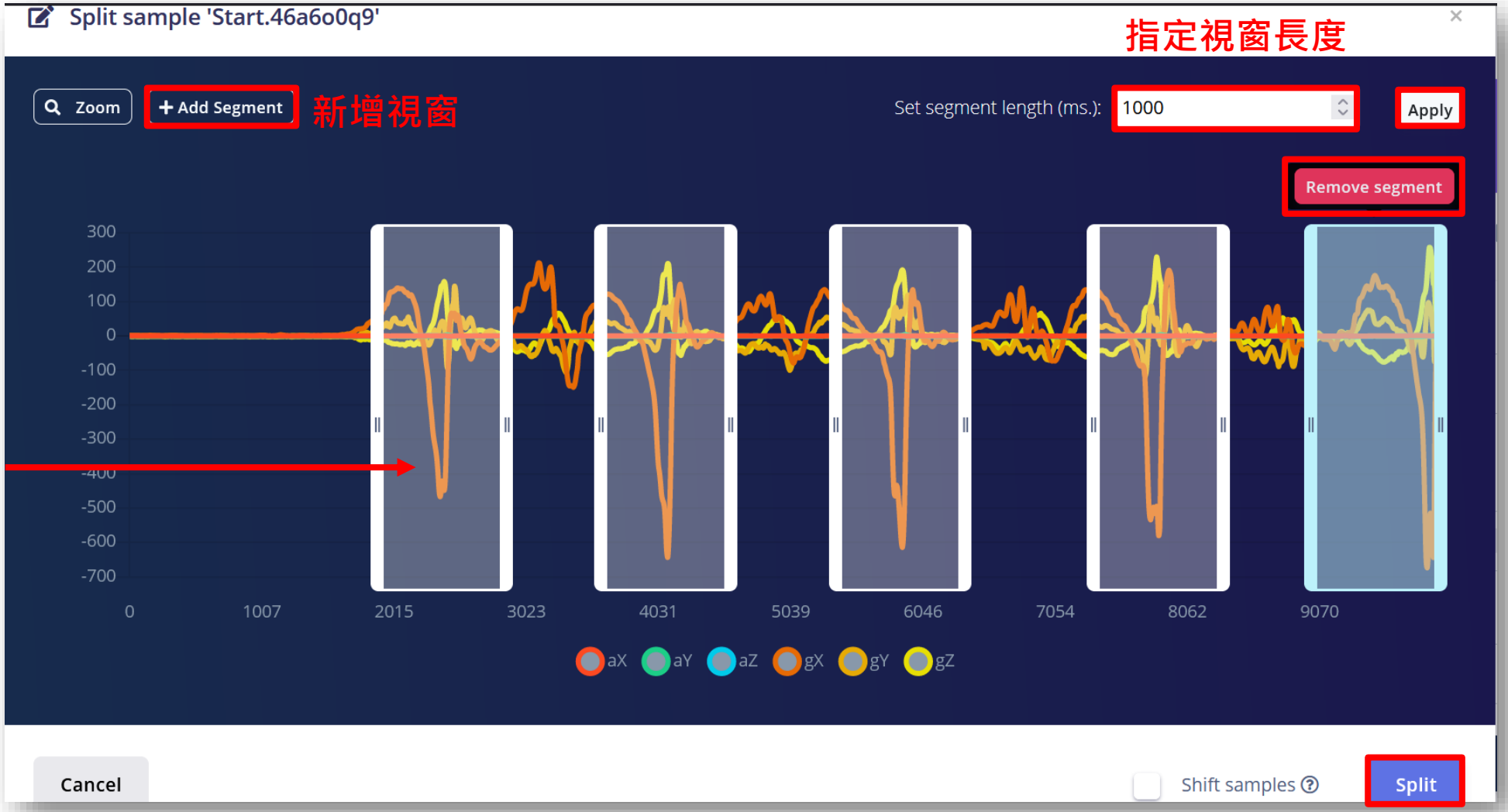
從外部裝置取得資料


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
OmniXRI /
Dataset
Data explorer
Data sources
CSV Wizard
Dataset
Add data
Start building your dataset by adding some data.
+ Add data
Collect data
Device
Xiao BLE Sense
Label
指定標籤名稱
Sample length (ms.)
取樣時長
Sensor
Sensor with 6 axes (aX, aY, aZ, gX, gY
Frequency
128Hz
Start sampling

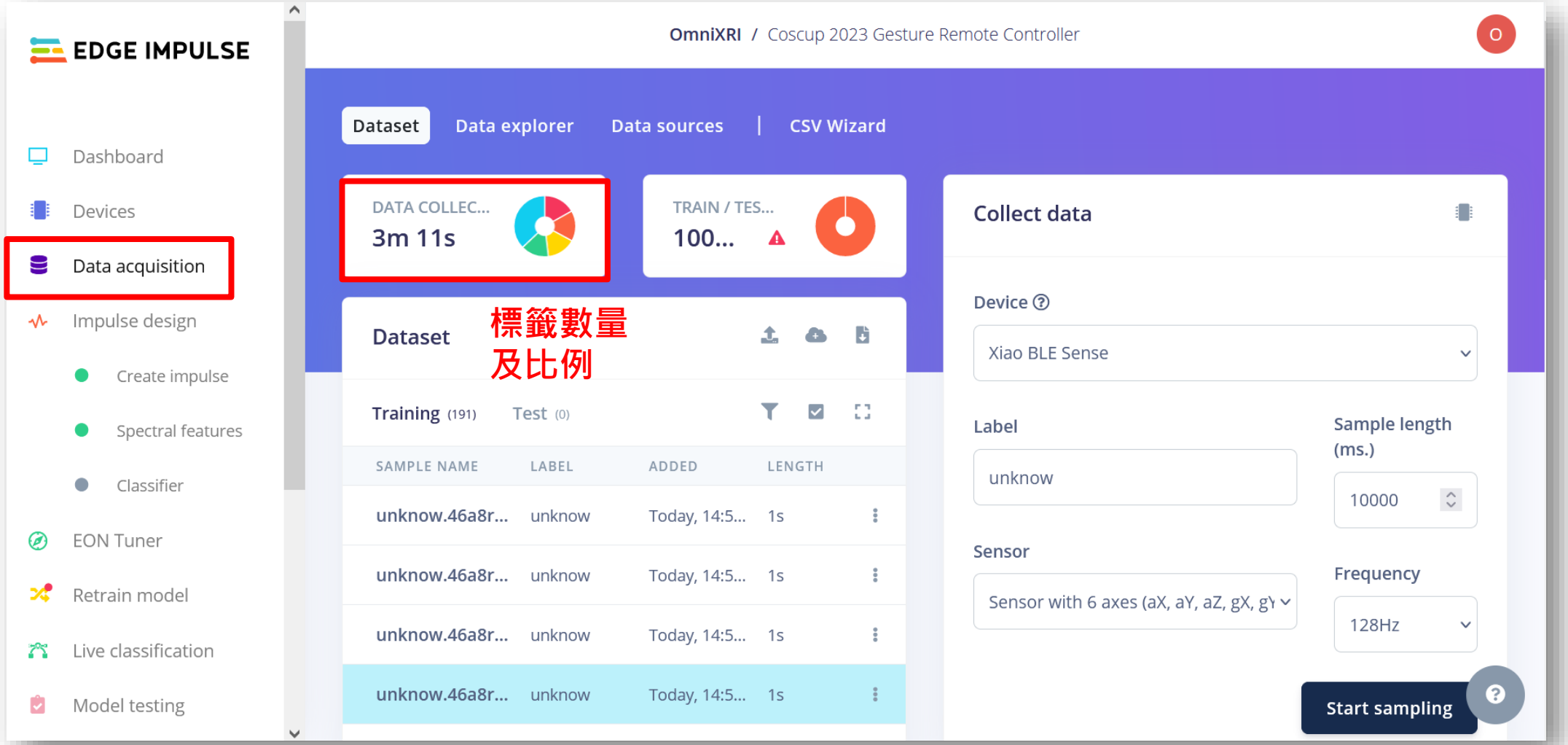
大量收集樣本並分割成獨立可訓練樣本



原始資料與自動分割

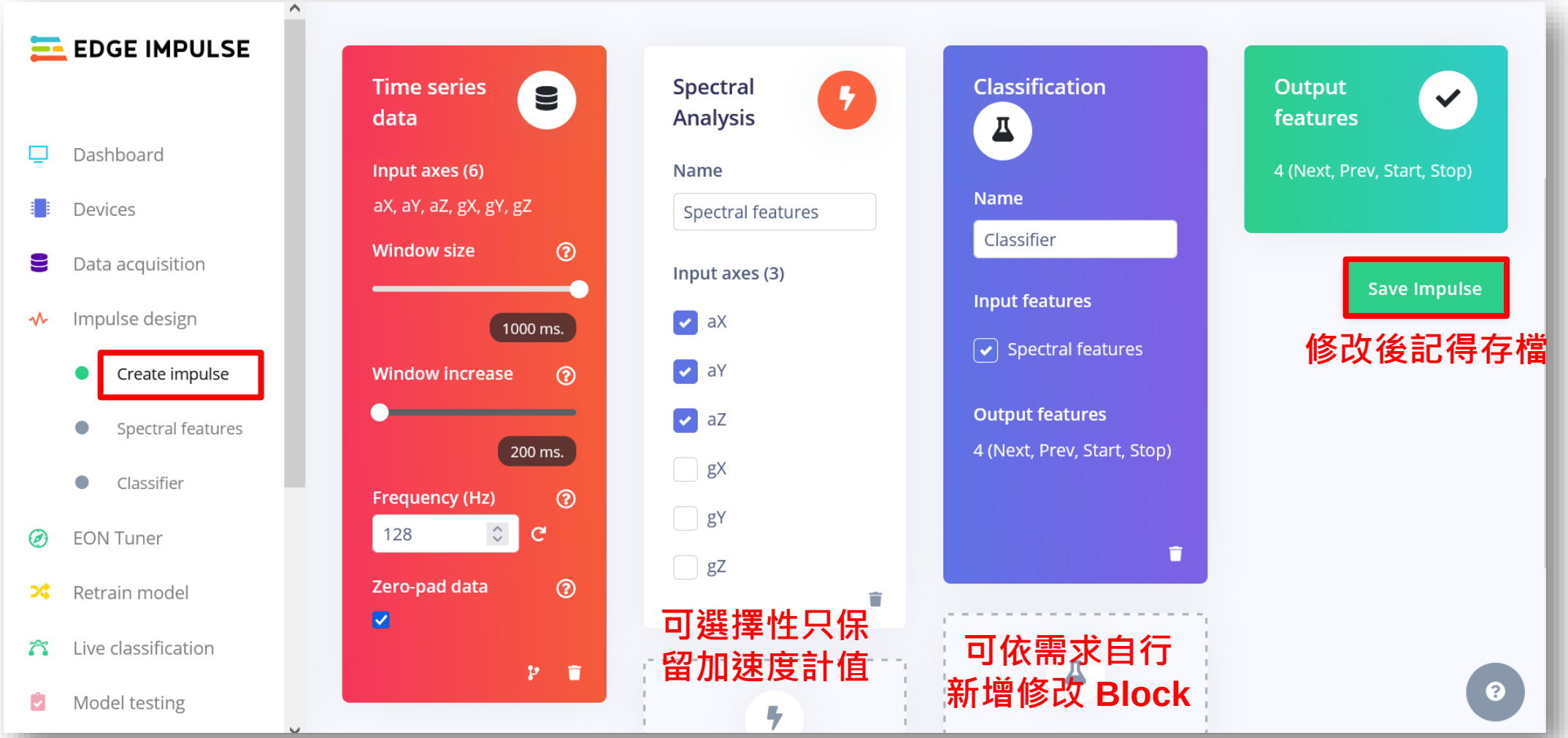

Split sample 'Start.46a6o0q9'
指定視窗長度
Zoom
+ Add Segment
新增視窗
Set segment length (ms.):
1000
Apply
自動分割
Remove segment
刪除視窗
300
200
100
0
-100
-200
-300
-400
-500
-600
-700
0
1007
2015
3023
4031
5039
6046
7054
8062
9070
aX
aY
aZ
gX
gY
gZ
按住滑鼠
左鍵不放
可左右拖
拉移動
Cancel
Shift samples
Split
完成分割

反複收集分割，建立完整資料集

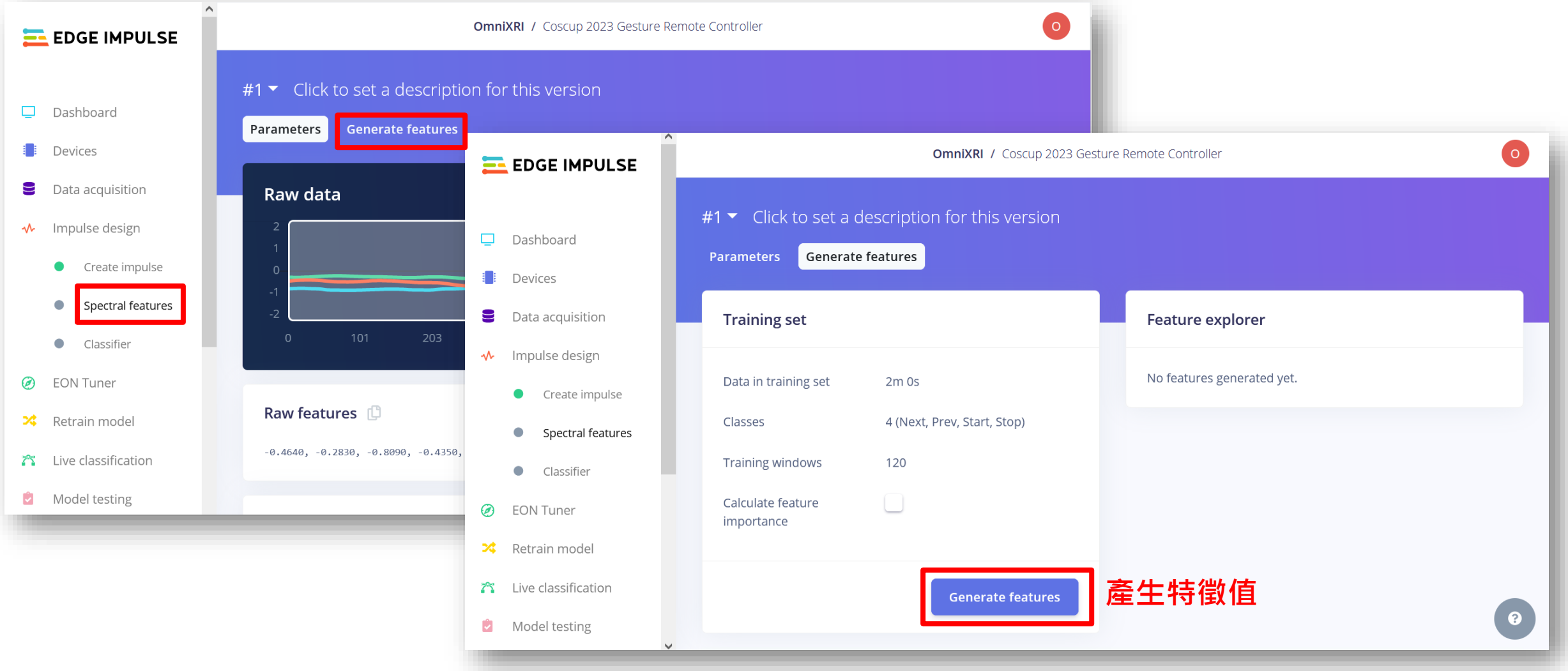


9.3. 模型選用與訓練

選擇模型及設定必要參數



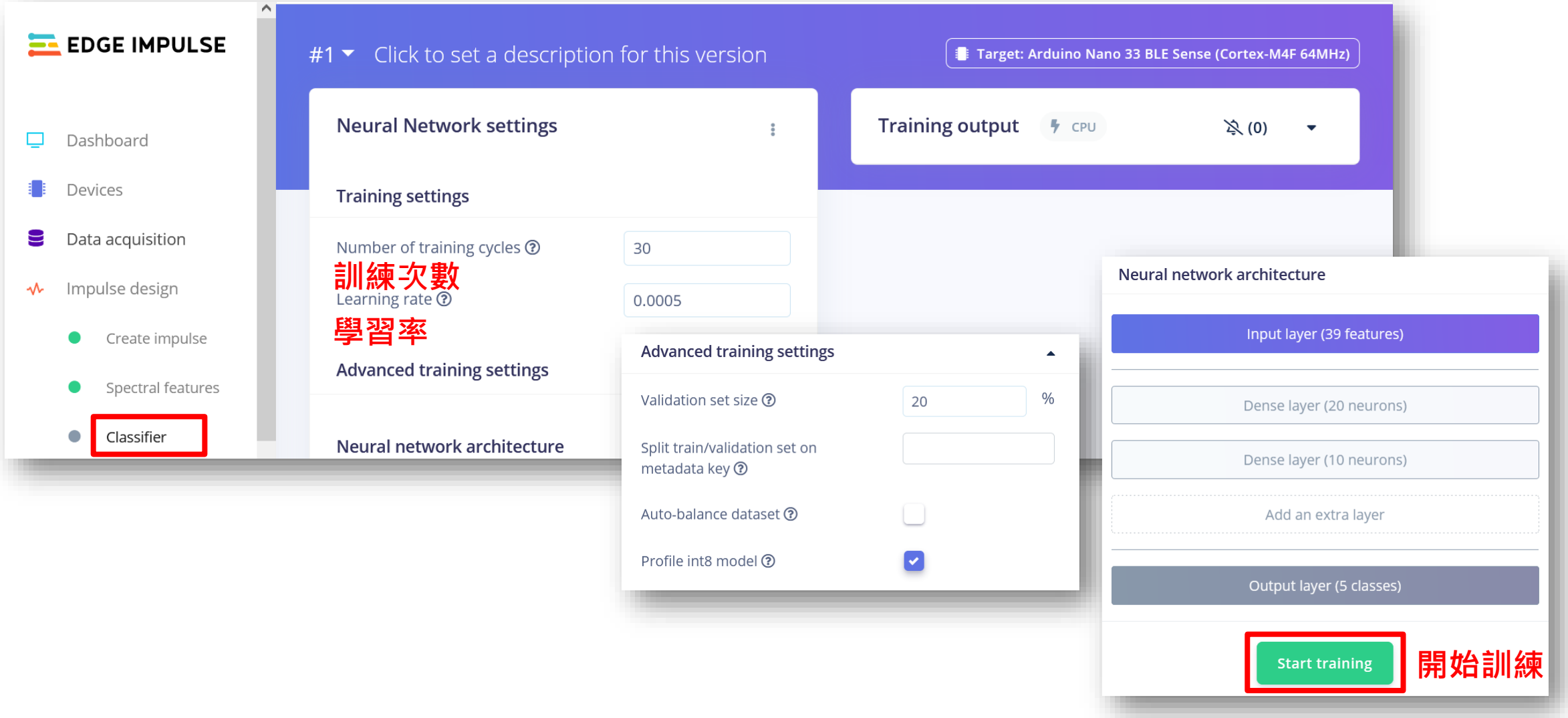
提取資料特徵


EDGE IMPULSE
OmniXRI / Coscup 2023 Gesture Remote Controller
#1 Click to set a description for this version
Parameters
Generate features
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Raw data
Raw features
-0.4640, -0.2830, -0.8090, -0.4350,
Training set
Data in training set
2m 0s
Classes
4 (Next, Prev, Start, Stop)
Training windows
120
Calculate feature importance
Feature explorer
No features generated yet.
Generate features
產生特徵值

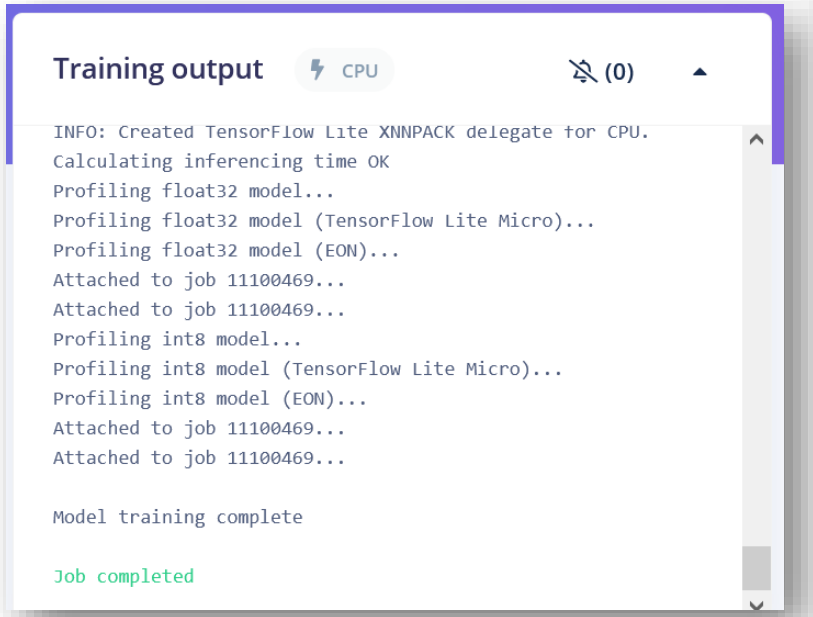
提取特徵結果

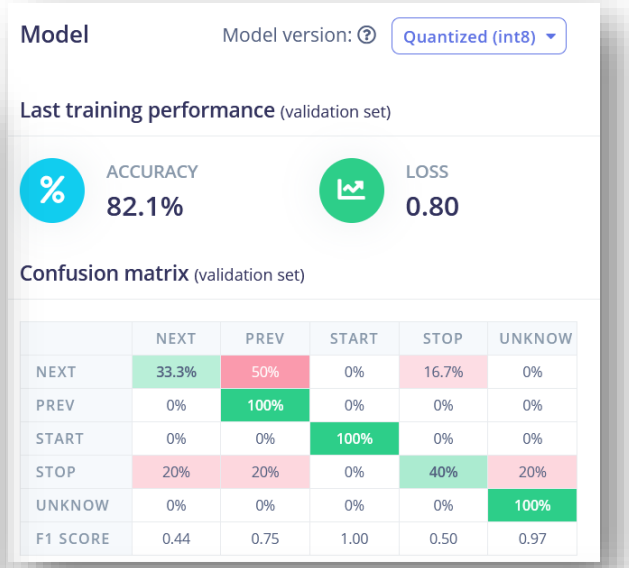
以指定的硬體進行資源
使用及推論時間估測

設定分類訓練相關參數


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
#1
Click to set a description for this version
Target: Arduino Nano 33 BLE Sense (Cortex-M4F 64MHz)
Neural Network settings
Training settings
Number of training cycles
30
訓練次數
Learning rate
0.0005
學習率
Advanced training settings
Neural network architecture
Training output
CPU
(0)
Advanced training settings
Validation set size
20
%
Split train/validation set on metadata key
Auto-balance dataset
Profile int8 model
Neural network architecture
Input layer (39 features)
Dense layer (20 neurons)
Dense layer (10 neurons)
Add an extra layer
Output layer (5 classes)
Start training
開始訓練

開始進行模型訓練及結果顯示





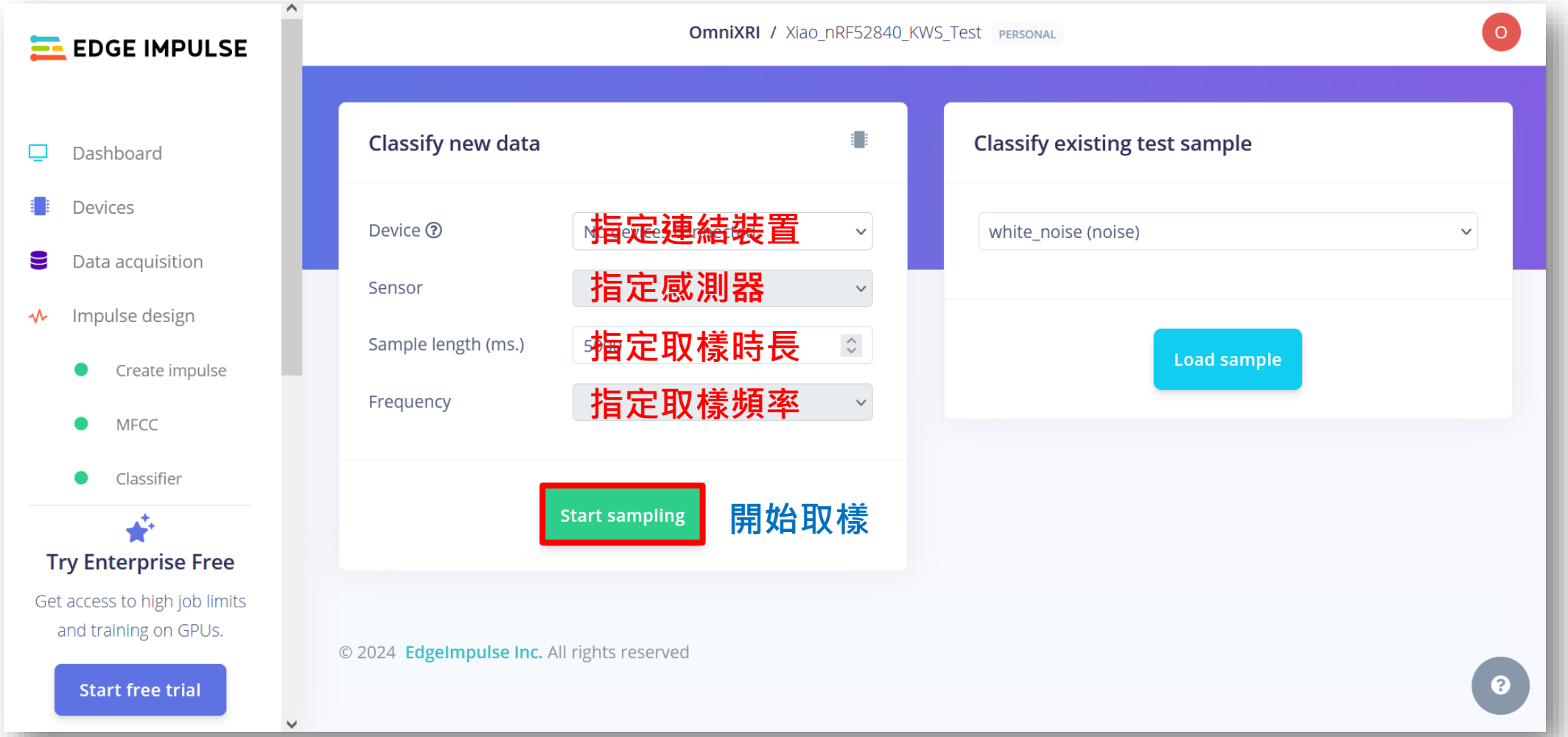
	NEXT	PREV	START	STOP	UNKNOW
NEXT	33.3%	50%	0%	16.7%	0%
PREV	0%	100%	0%	0%	0%
START	0%	0%	100%	0%	0%
STOP	20%	20%	0%	40%	20%
UNKNOW	0%	0%	0%	0%	100%
F1 SCORE	0.44	0.75	1.00	0.50	0.97

30 Epoch 結果

	NEXT	PREV	START	STOP	UNKNOW
NEXT	83.3%	16.7%	0%	0%	0%
PREV	0%	100%	0%	0%	0%
START	0%	0%	100%	0%	0%
STOP	20%	0%	0%	80%	0%
UNKNOW	0%	0%	0%	0%	100%
F1 SCORE	0.83	0.92	1.00	0.89	1.00

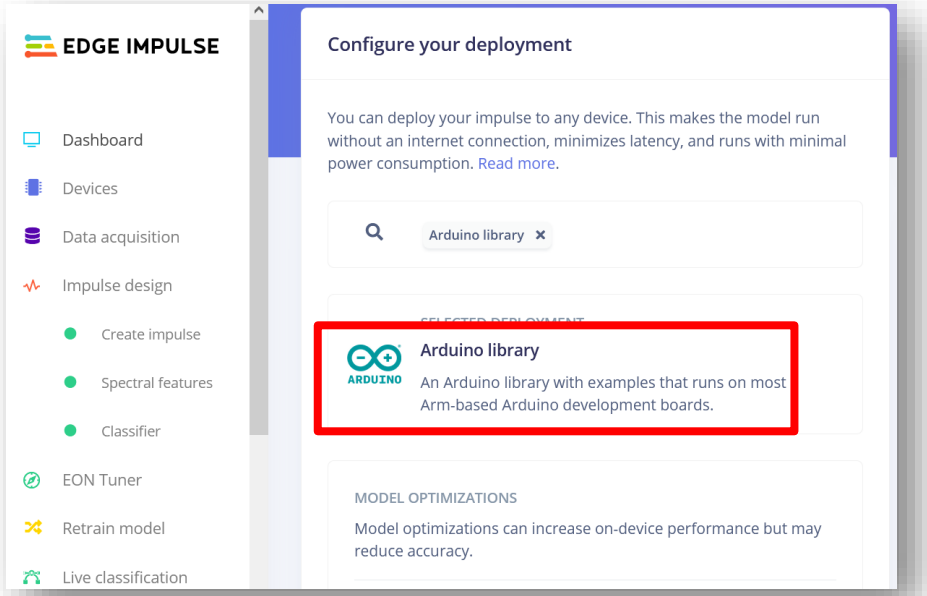
100 Epoch 結果

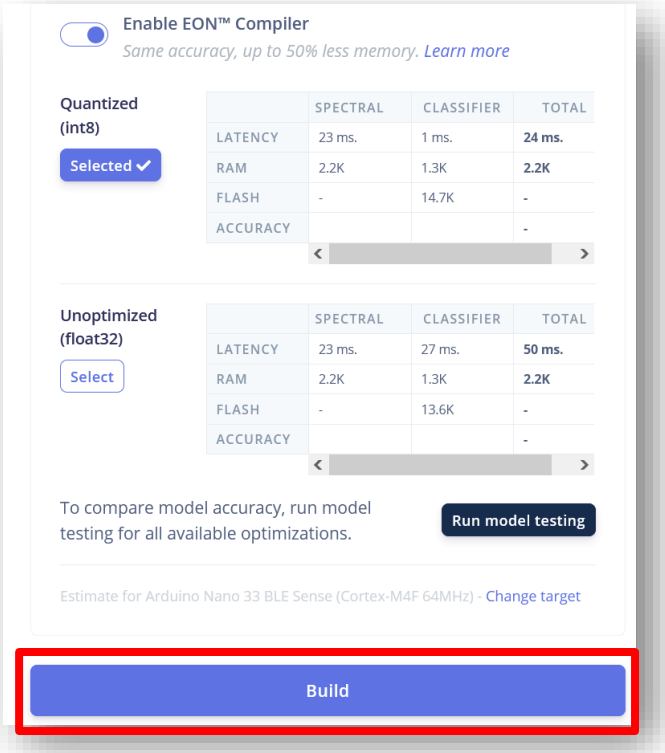
線上測試（從外部裝置取樣）

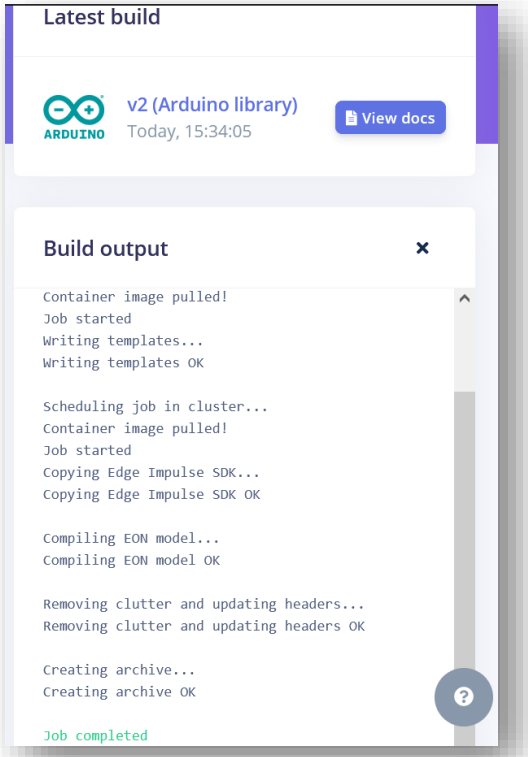


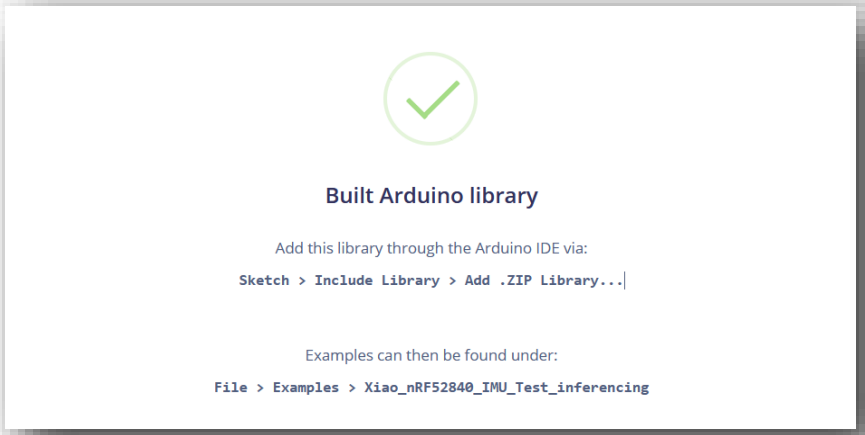
9.4. 模型部署與測試

選擇部署種類及設定參數









以 Arduino 函式庫(*.zip)方式輸出
預設檔名為「ei-專案名稱-arduino-版本序號.zip」

導入 Arduino 函式庫並進行推論測試(1/2)

- 新增函式庫 **Sketch > Include Library > Add .ZIP Library …**
- 新增範例 **File > Examples > ei-專案名稱_inferencing > nano_ble33_sense > nano_ble33_sense_accelerometer（原始範例輸出結果從 COM 輸出文字串）**
- 經編譯會在「**#error "Invalid model for current sensor"**」這列出現「**"invalid model for current sensor"**」，主要原因為Edge Impulse打包時是以Sensor_Fusion方式處理而不是Accelerometer方式造成。
- 點擊選單 **File > Preferences** 找到 Sketchbook Folder，進到該路徑後，進到 \libraries**專案名稱**_inferencing\src\model-parameters\ 開啟 **model_metadata.h** 。找到「 #define EI_CLASSIFIER_SENSOR EI_CLASSIFIER_SENSOR_**FUSION** 」，將定義改成「 EI_CLASSIFIER_SENSOR_**ACCELEROMETER** 」存檔即可。

導入 Arduino 函式庫並進行推論測試(2/2)

- 回到 Arduino 範例後會發現使用的運動感測器不同，所以需將「#include <Arduino_LSM9DS1.h>」註解掉並新增下列程式

```
#include <LSM6DS3.h>
#include <Wire.h>
LSM6DS3 myIMU(I2C_MODE, 0x6A);
```

```
/* Includes ---------------------------------------------
#include <Xiao_nRF52840_IMU_Test_inferencing.h>
//#include <Arduino_LSM9DS1.h> //Click here to get the libr
#include <LSM6DS3.h>
#include <Wire.h>
/* Constant defines -------------------------------------
#define CONVERT_G_TO_MS2    9.80665f
#define MAX_ACCEPTED_RANGE  2.0f        // starting 03/2022

LSM6DS3 myIMU(I2C_MODE, 0x6A);    //I2C device address 0x6A
```

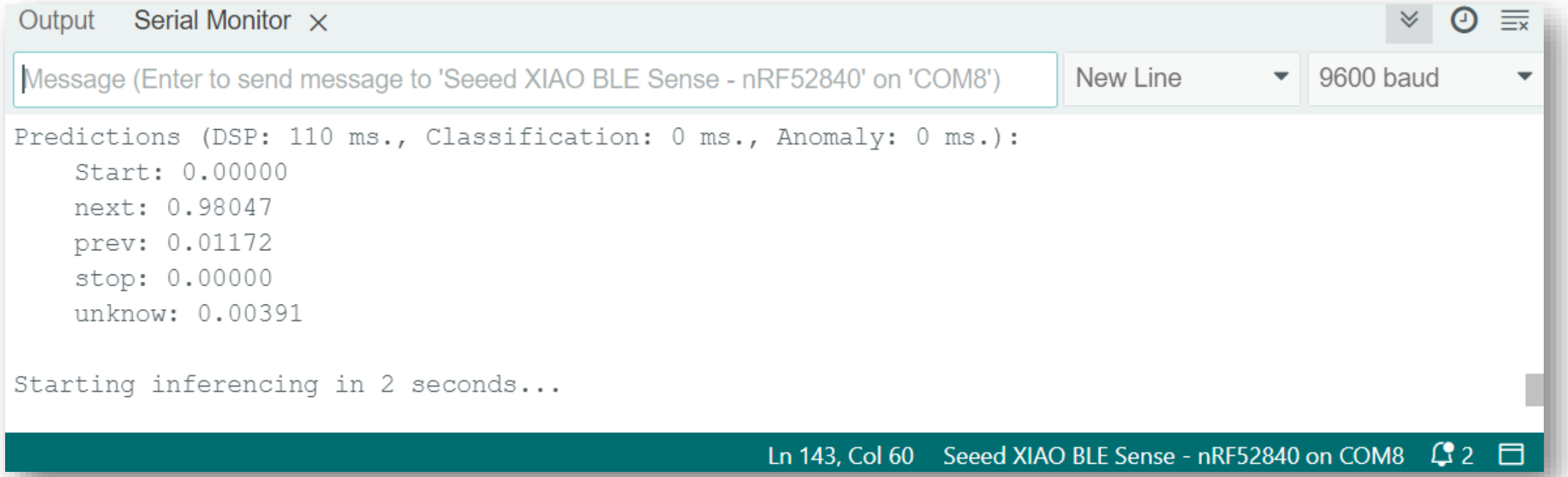
- 在Setup函式中的初始化也要將「if (!IMU.begin()) {」修改為「 if (!**myIMU**.begin()) { 」

- 另外由於讀取IMU運動感測器函式不同因此也需修改如左。

```
//        IMU.readAcceleration(buffer[ix], buffer[ix + 1],
        buffer[ix]     = myIMU.readFloatAccelX();
        buffer[ix + 1] = myIMU.readFloatAccelY();
        buffer[ix + 2] = myIMU.readFloatAccelZ();
```

- 重新編譯前確認開發板是設定在「 **Seeed nRF52 mbed-enabled Boards** > Seeed XIAO BLE Sense - nRF52840 」，不然程式運行後會找不到IMU LSM6DS3。

手勢辨識結果（Arduino部份）



- **推論結果，包括前處理(DSP)、分類及異常時間 (ms)**
- **將五個標籤的置信度顯示出來，1.0為完全正確，0.0為完全不正確，可設置門檻值來限制置信度不高之答案。**
- **No Code型式，範例程式直接編譯上傳，不需改寫任何程式，輸出結果由 COM 接收文字串。**

Arduino 2.0 快速(增量)編譯

非常重要：

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

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

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

參考文獻

- 許哲豪，臺灣科技大學資訊工程系「人工智慧與邊緣運算實務」（2021~2023）

https://omnixri.blogspot.com/p/ntust-edge-ai.html

- 許哲豪，OmniXRI's Edge AI & TinyML 小學堂 Youtube 直播課程總結

https://omnixri.blogspot.com/2024/06/omnixris-edge-ai-tinyml-youtube.html

- 許哲豪，歐尼克斯實境互動工作室系列發文─TinyML(MCU AI)系列

https://hackmd.io/1PK1URhIQ7GutcWgpgsWbg#TinyMLMCU-AI%E7%B3%BB%E5%88%97

- 許哲豪，【課程簡報】20240509_慈濟醫資_穿戴式人工智慧工作坊_利用TinyML技術快速搭建微型智慧應用

https://omnixri.blogspot.com/2024/05/20240509tinyml12.html

- 許哲豪， OmniXRI's Edge AI & TinyML 小學堂 【第13講】實作案例 ─運動辨識（快速指令表）

https://github.com/OmniXRI/Edge_AI_TinyML_Course_2024/blob/main/Ch13_Motion_Recognition/IMU_Quick_Guide.md

延伸閱讀

- 許哲豪，如何讓 Arduino 2.0 快速編譯（增量編譯）

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

- Seeed, Xiao nRF52840 Sense – Embedded ML – Motion Recognition based on Edge Impulse

https://wiki.seeedstudio.com/XIAOEI/

- Github, Seeed-Studio / Seeed_Arduino_LSM6DS3

https://github.com/Seeed-Studio/Seeed_Arduino_LSM6DS3

- Seeed, Xiao nRF52840 Sense – Embedded ML – Getting Started with TensorFlow Lite on Seeed Studio XIAO nRF52840 Sense

https://wiki.seeedstudio.com/XIAO-BLE-Sense-TFLite-Getting-Started/

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

Facebook : Jack Omnixri
FB社團： Edge AI Taiwan邊緣智能交流區
電子信箱：omnixri@gmail.com
部落格：https://omnixri.blogspot.tw
開　源：https://github.com/OmniXRI

YOUTUBE 直播： https://www.youtube.com/@omnixri1784/streams