2022 Synopsys ARC 盃 AIoT 設計應用競賽
提案作品

跑步訓練監控平台

組名

仰望星空

報告人

國立中正大學 通訊工程研究所 - **林子華、 楊博禕**

指導教授

國立中正大學 通訊工程研究所 - **李皇辰 教授**

國立中正大學 運動競技學系暨運動與休閒教育碩士班 - **王順正 教授**

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

作品概述

隨著社會經濟的發展，人們不只追求物質上的滿足，也開始追逐心靈上的富足與完美的體態，越來越多人慢慢養成了運動的習慣。其中，慢跑是一項風行全球的運動，不需要花大錢買裝備，又同時有諸多好處：健康、抒發壓力、成就感等。

然而，不當的步態(gait)卻會導致跑步的穩定度下降，甚至是因為運動傷害而停止這項運動。此外，跑步的特質，例如腳著地與騰空時間等，都是會顯著影響到跑步效率。我們需要一位教練來觀察跑者的步態，適時糾正其姿勢與跑步的策略。

作品概述

或許不是每個人都有能力花錢請教練幫忙矯正跑步的姿勢，於是我們提出一個替代方案：

分析步態時空參數，再請教練依據這些資料給出建議

本研究提出一個基於運動監控的開發平台，主要應用於跑步訓練。本平台包含穿戴式的裝置，用於偵測跑步運動狀態與收集運動資料，還有與AI分析運算平台。 希望透過開發本平台，可以讓跑步運動可以科學化的被監控分析，就好像隨時有一個教練在旁邊幫助，提升跑步運動的效率。

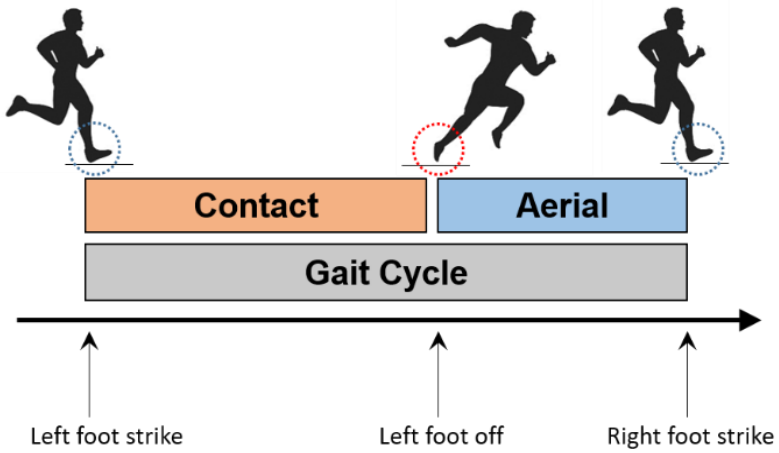
Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

難點與創新

傳統的運動監測設備，有些必須於實驗室等控制環境，使用攝影機、感測器等來監控跑步狀態；這種做法無法直接運用於戶外的跑步運動上。市面上有些公司在開發類似的穿戴式感測器來偵測跑步的步伐頻率，但是無法進行近細緻的分析，也只能用off-line的方式分析資料。

本研究將開發自製的**運動穿戴式裝置**，預計是安裝於鞋子、腰帶，甚至是耳機等位置，收集跑步之動態資訊，來on-line取得三種步態時空參數(spatial-temporal gait parameters)，包含**(1)步頻、(2)騰空時間、(3)著地時間**。這三項參數，將可以有效地評估跑步之品質與效益。



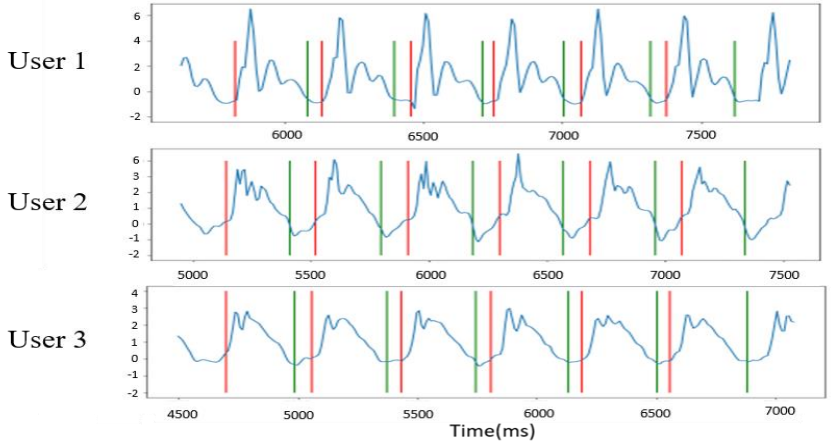
難點與創新

但是，如何於實驗室外準確地量測到步頻、騰空時間與著地時間，是一個非常困難的研究議題。

因為基於每位跑者的跑步習慣略有不同，每個人的體格(男女、老少，高矮胖瘦，是否有接受過專業跑步訓練等)都會影響跑步時的特性。

觀察下圖，並不是所有的標註點都落在局部最大值或最小值，**很難透過統計方法找出著地和離地時間點，因此我們導入深度學習技術。**

我們會用ARC EM9D開發板來開發**AI分析運算平台**之即時AI演算法，來計算運動的參數，把資訊回傳給遠端的教練，來幫助跑者了解自身跑步習慣的缺點。



三位跑者的加速度資料
紅線: 著地時間點
綠線: 離地時間點

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

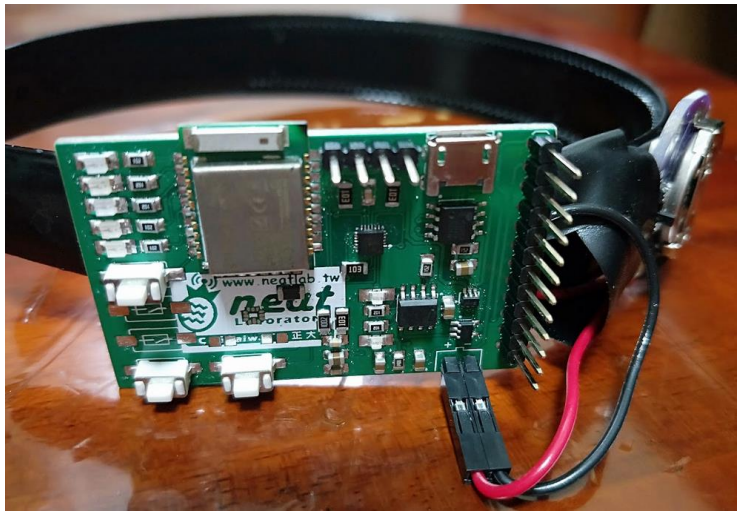
設計與實現-流程

硬體：參賽方提供的ARC EM9D AIoT DK + 實驗室自製的nRF52840。

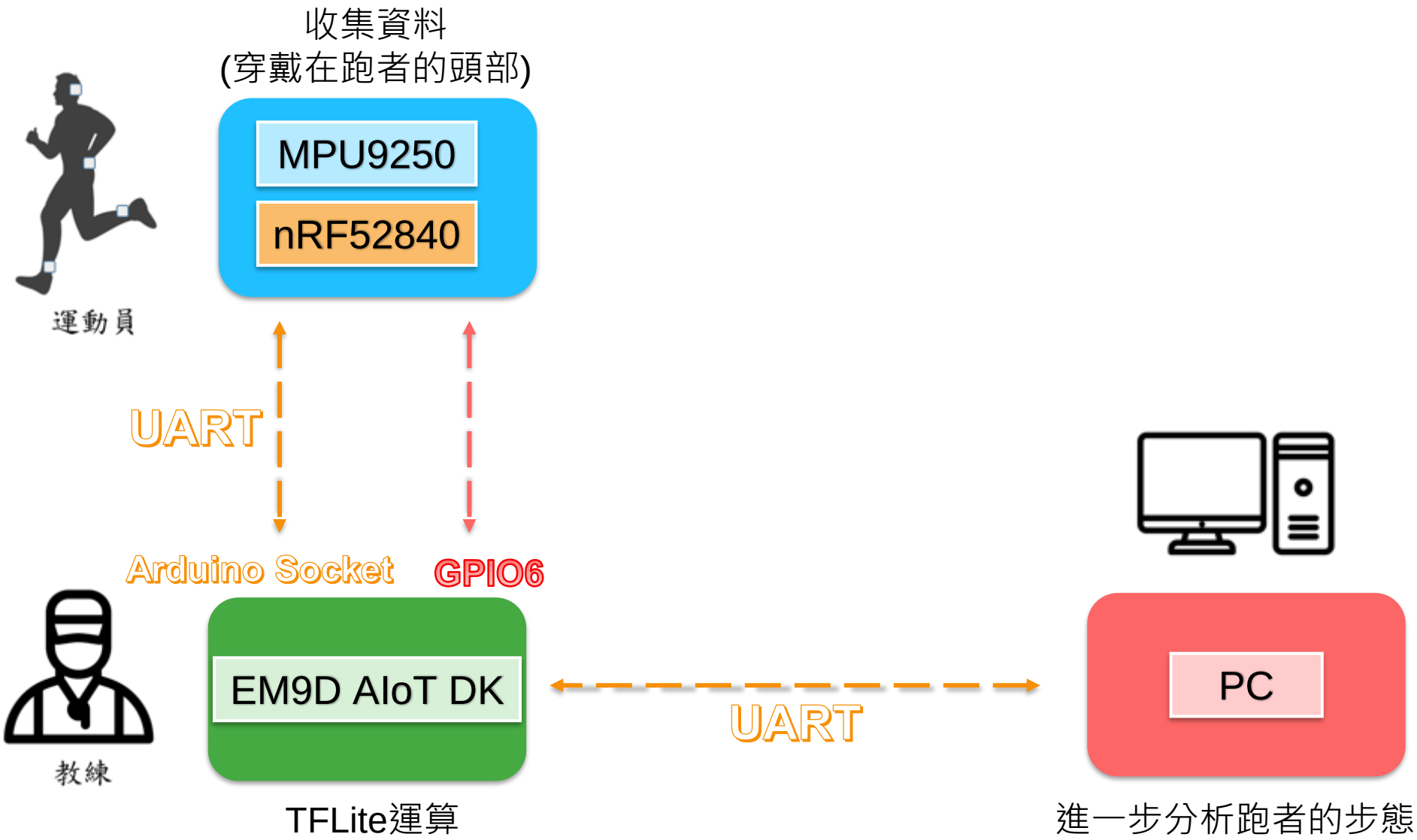
流程：

1) 運動過程收集跑者的加速度資料
2) 使用機器學習模型用於預測觸地時間 & 計算相關步態參數
3) 回報給教練做後續的分析

實驗室自製的nRF52840
上面有MPU9250



設計與實現-硬體架構

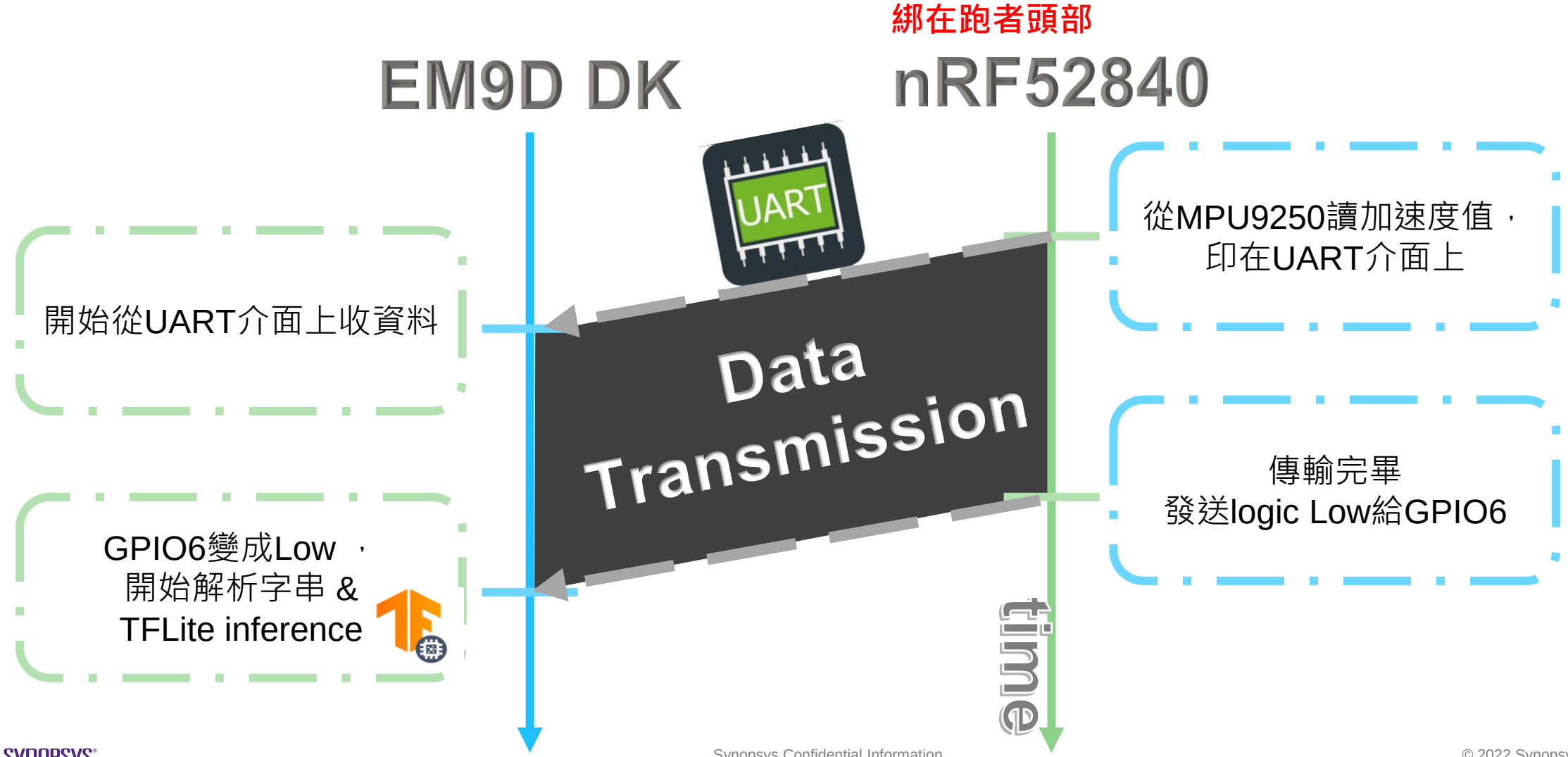


Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

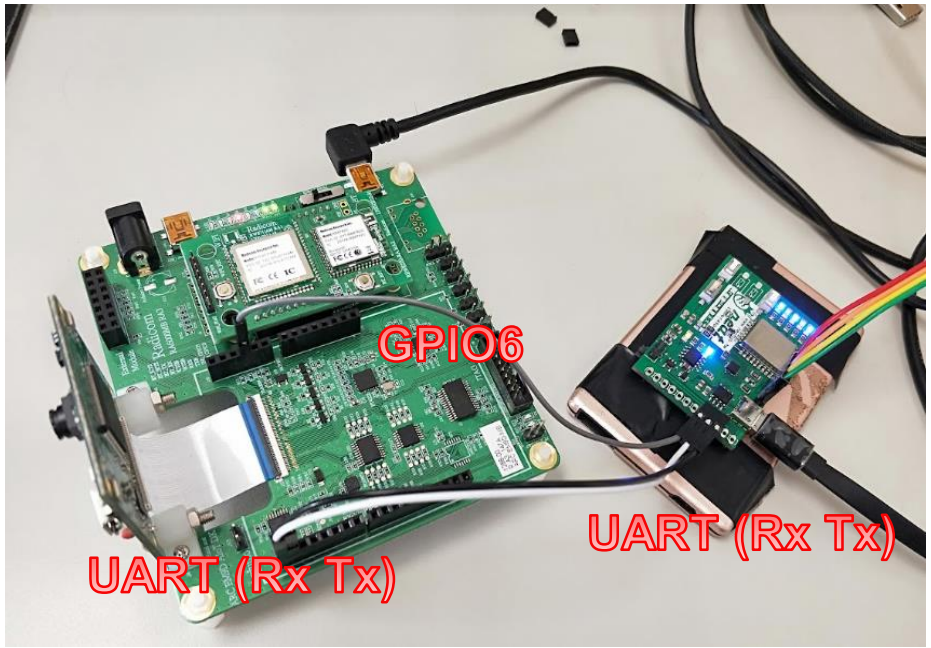
作品進度 - 流程

流程如下，縱軸為時間

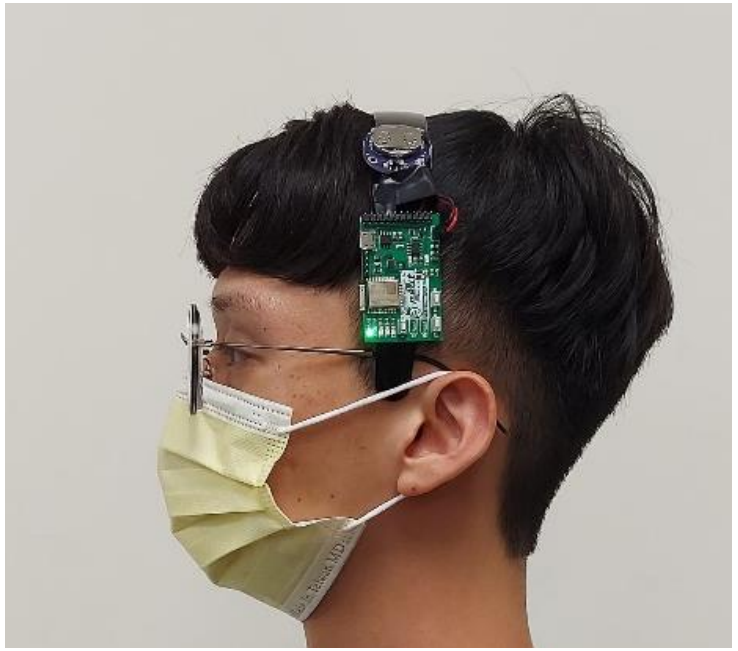


作品進度 - 穿戴

- nRF52840負責資料收集。
- EM9D DK僅用於AI運算Server。



EM9D DK & nRF52840的線路接法



nRF52840 穿戴於頭部

作品進度 - 傳輸介面選擇

雖然EM9D AIoT DK上有加速度計，但是**體積過大，不易配戴在跑者身上。**

於是我們採用實驗室自製的nRF52840 + MPU9250來收取資料，**以UART介面傳輸回EM9D AIoT DK**。

*****原本要使用BLE來傳輸資料，因為EM9D對於BLE支持有限，所以只好先以UART來傳送資料。*****

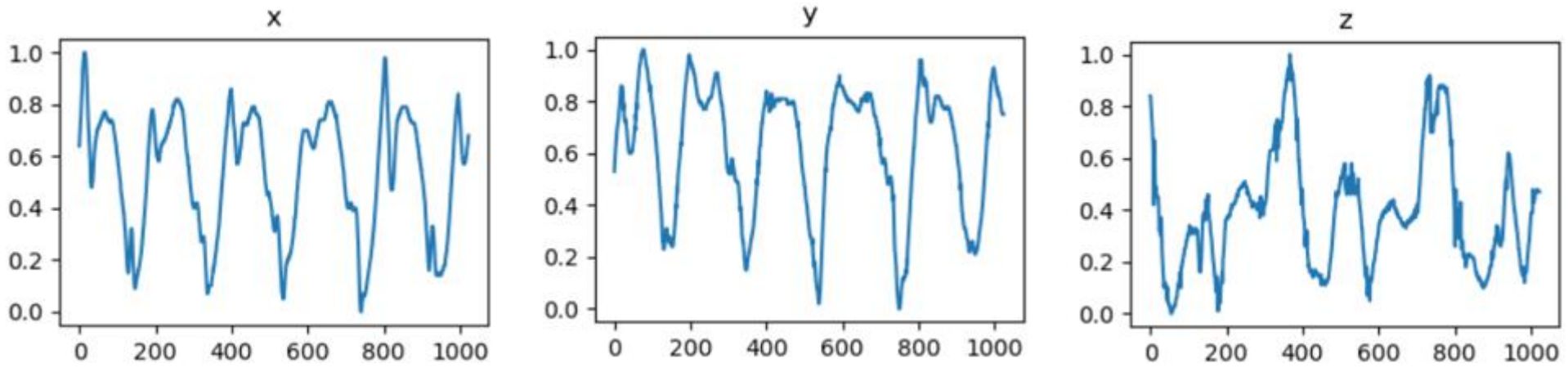
右圖為EM9D AIoT DK端，收到nRF52840的資料再印到PuTTY上的結果。

四個column分別代表：index、x軸加速度、y軸加速度、z軸加速度。

作品進度 – 資料收集

- 我們蒐集了27人的加速度資料
- 等待15秒的熱身完畢之後才開始正式收集
- 持續30秒
- Sample rate: 500 Hz

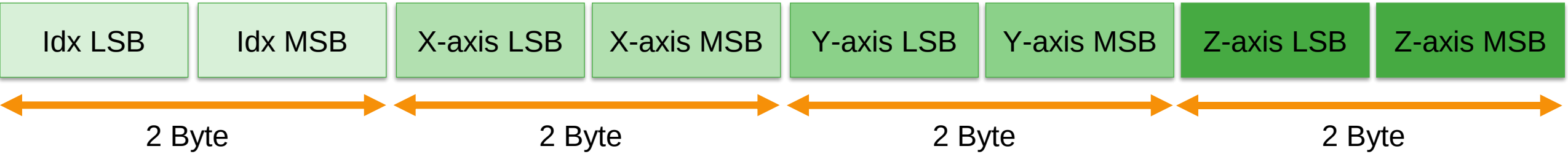
三軸加速度資料
(x軸為時間 y軸為scale後的加速度)



作品進度 – 資料傳輸

- 原先是以字串傳輸，發現占用過多的RAM
  - 格式Idx x y z
  - 156 0.02 -0.99 0.01 ➔ 總共占用16Byte(不含空白)
  - 除了float精準度只能到小數點2位之外，負號還會額外占用一個Byte
  - 實際上佔用更多

- 後來我們把每一個Byte印在UART上，再由EM9D DK去重組
  - 最後的傳輸格式如下
  - 每一筆資料都是8Byte
  - 占用的RAM比原先少了一倍以上

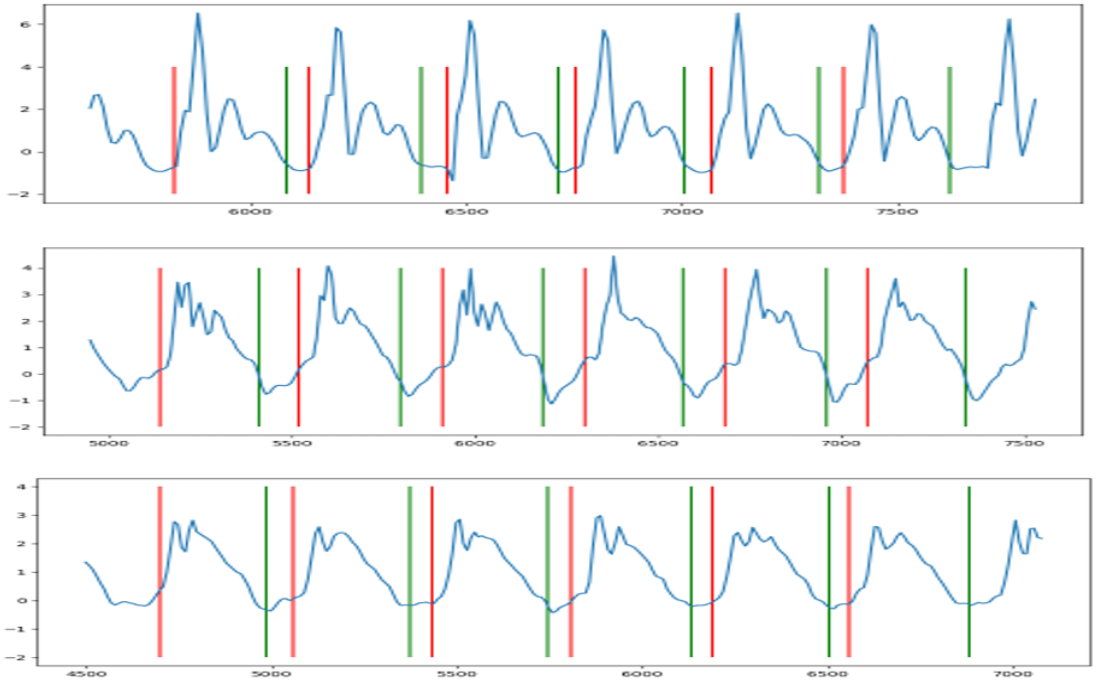


作品進度 - 現象

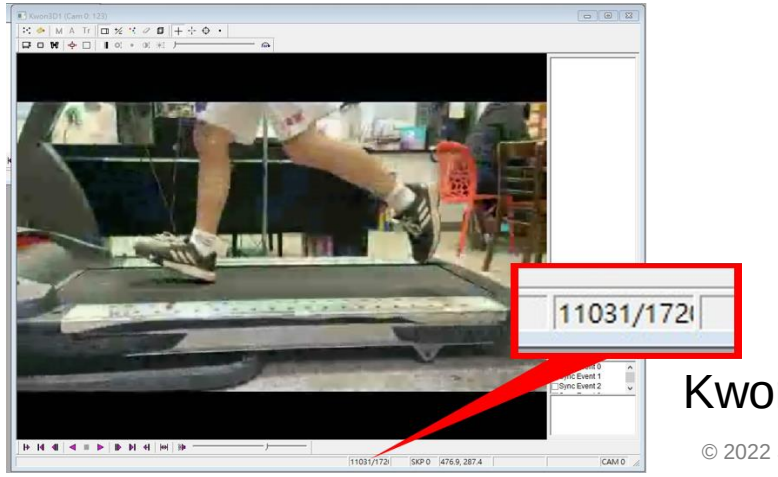
- 量測到三軸加速度的資料之後，
- 我們使用Kwon3D運動分析軟體，慢速攝影拍攝可以得到觸地和離地的時間點

- 觀察右圖，我們**很難用邏輯去判斷觸地和離地的時間點**，因為它們並不是剛好落在局部最大值或最小值。

- 因此我們**採用機器學習的觀念**，來幫我們解決這個問題。



不同人跑步的特徵(紅線表示**觸地**時間點，綠線表示**離地**時間點)，橫軸表示時間，縱軸表示垂直方向之加速度值



Kwon3D分析軟體

作品進度 - Keras模型

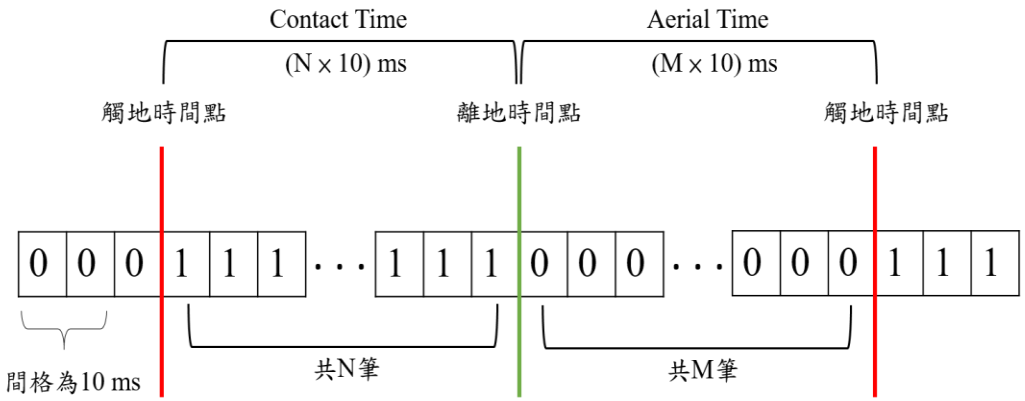
- 首先，我們模型的輸入有3維，共20個瞬間：
  - 分別是x、y、z三軸加速度 &
- 我們選擇了一個簡單的DNN模型，因為LSTM模型中有一些運算函式還TFLite還不支援。
  - 簡單的模型方便TFLite conversion
- 輸出則是兩類
  - Class0：離地
  - Class1：觸地

白話的說：以20個瞬間預測下一個瞬間是觸地還是離地

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	1936
batch_normalization (BatchNo	(None, 16)	64
dense_1 (Dense)	(None, 32)	544
batch_normalization_1 (Batch	(None, 32)	128
dense_2 (Dense)	(None, 16)	528
batch_normalization_2 (Batch	(None, 16)	64
flatten (Flatten)	(None, 16)	0
dense_3 (Dense)	(None, 2)	34

Total params: 3,298
Trainable params: 3,170
Non-trainable params: 128

Keras DNN模型 summary



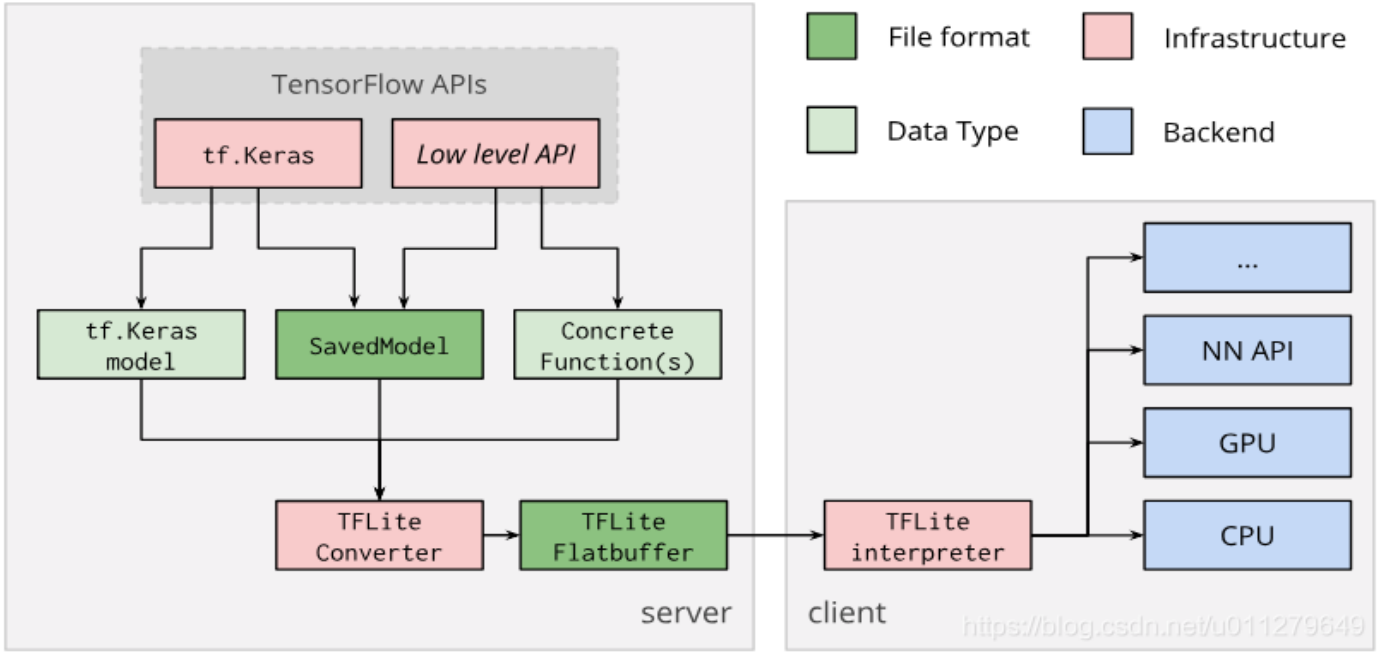
作品進度 - 轉換成TFLite format

- 在inference(終端運算)之前，我們需要將Keras model轉換成TFLite format。
  - 對weight做quantization
  - **節省**記憶體空間 & Flash空間
  - 也提高了**運算的速度**
  - **精度略為下降**

- .tflite file同時包含了weight和model architecture

```
unsigned char generated_quant_model_tflite[] = {
  0x28, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x12, 0x00,
  0x1c, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00, 0x10, 0x00, 0x14, 0x00,
```

model.h中的weight



TFLite 轉換流程

作品進度 - Inference

- 我們可以比較TFLite模型和Keras模型的運算結果。
  - 下圖為10筆資料最後一層的輸出**(class 0 or class 1的confidence)**
  - 因為TFLite模型經過quantization，所以精度會下降一些
  - 但是該判斷為class 0或者class 1，**兩者一致**

TFLite model:
10筆資料predict

```
[[0.99609375 0.        ]
 [0.99609375 0.        ]
 [0.99609375 0.00390625]
 [0.99609375 0.        ]
 [0.99609375 0.        ]
 [0.99609375 0.        ]
 [0.99609375 0.00390625]
 [0.90625    0.09375   ]
 [0.33203125 0.66796875]
 [0.9765625  0.0234375 ]]
```

Keras model:
10筆資料predict

```
([[1.        , 0.00000001],
  [0.99891114, 0.00108892],
  [0.9975666 , 0.0024334 ],
  [0.99987984, 0.0001201 ],
  [0.99999857, 0.00000149],
  [0.99999976, 0.00000021],
  [0.9975127 , 0.0024873 ],
  [0.90363556, 0.09636447],
  [0.34049824, 0.6595018 ],
  [0.9897404 , 0.01025964]],
```

Agenda

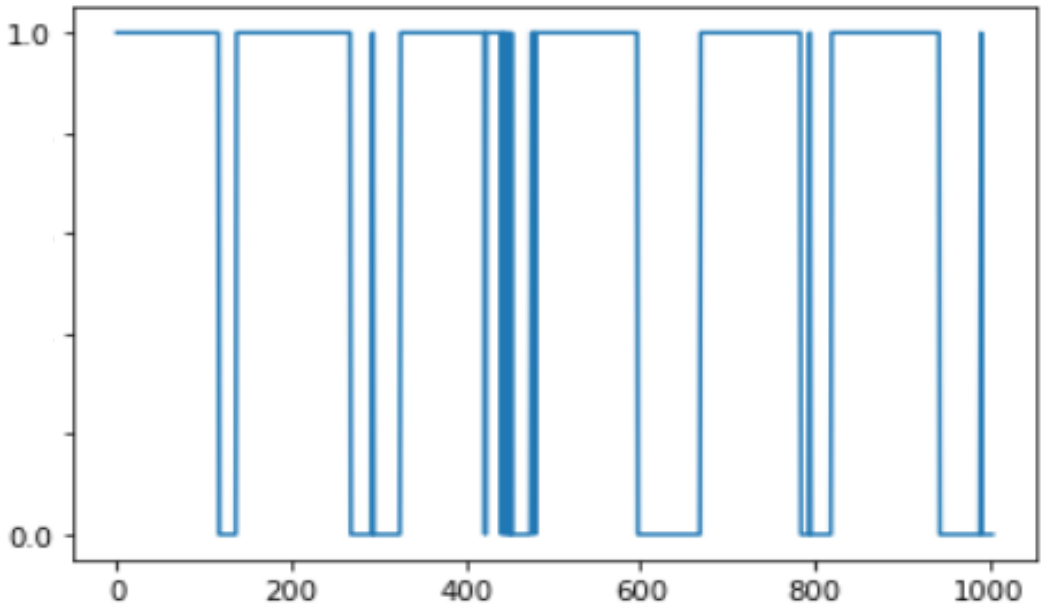
- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

功能展示

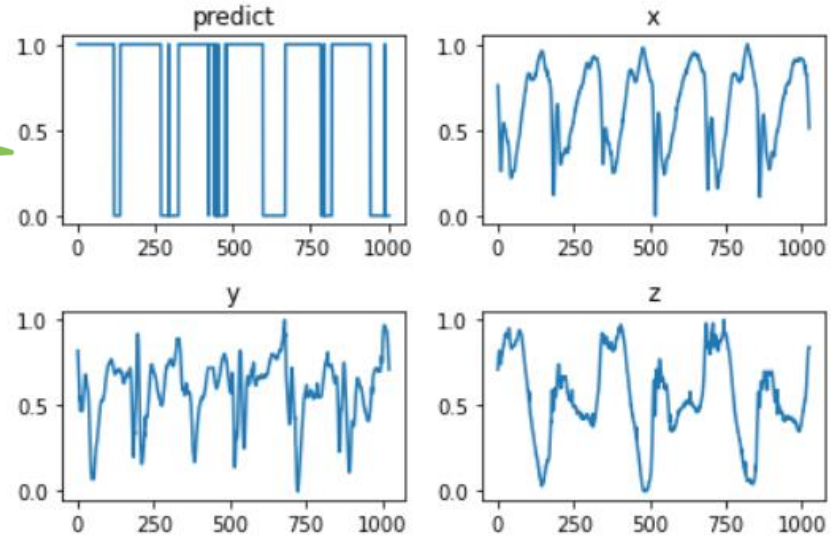
模型預測的結果

- 理想：一個完美的方波
- 現實：有鋸齒的方波

DNN模型預測結果(1=著地 0=離地)



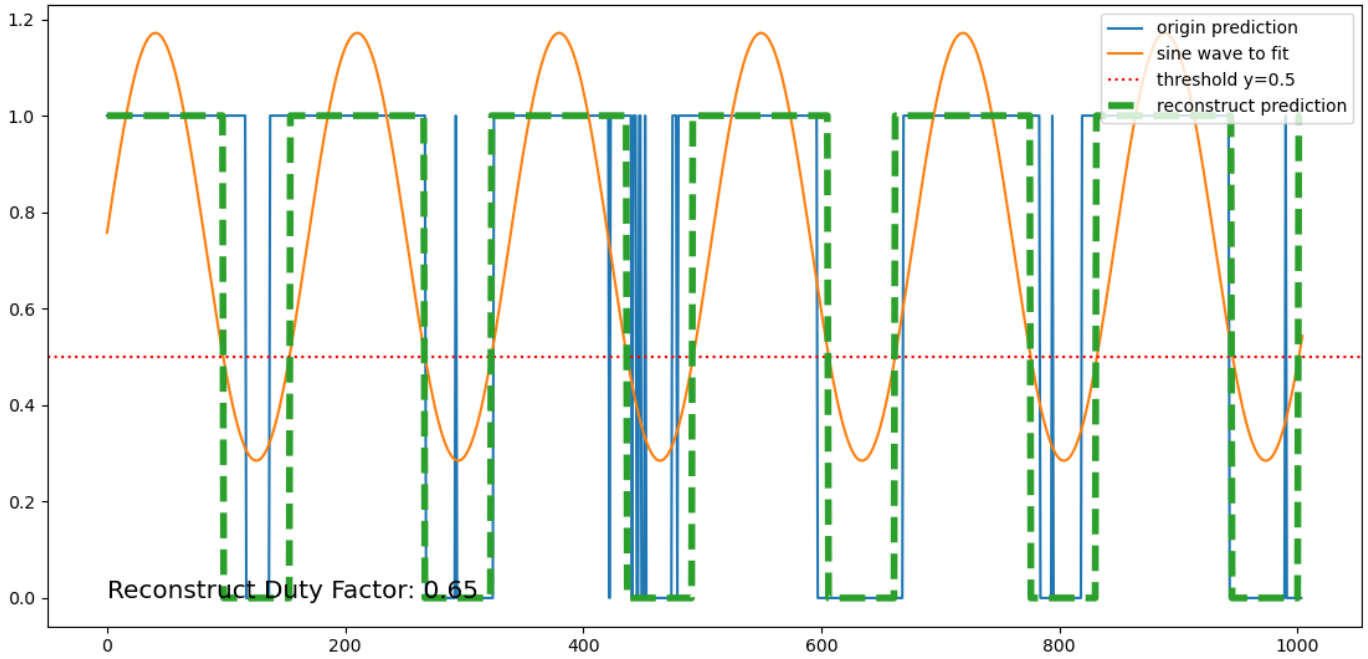
預測結果 & 三軸加速度



功能展示

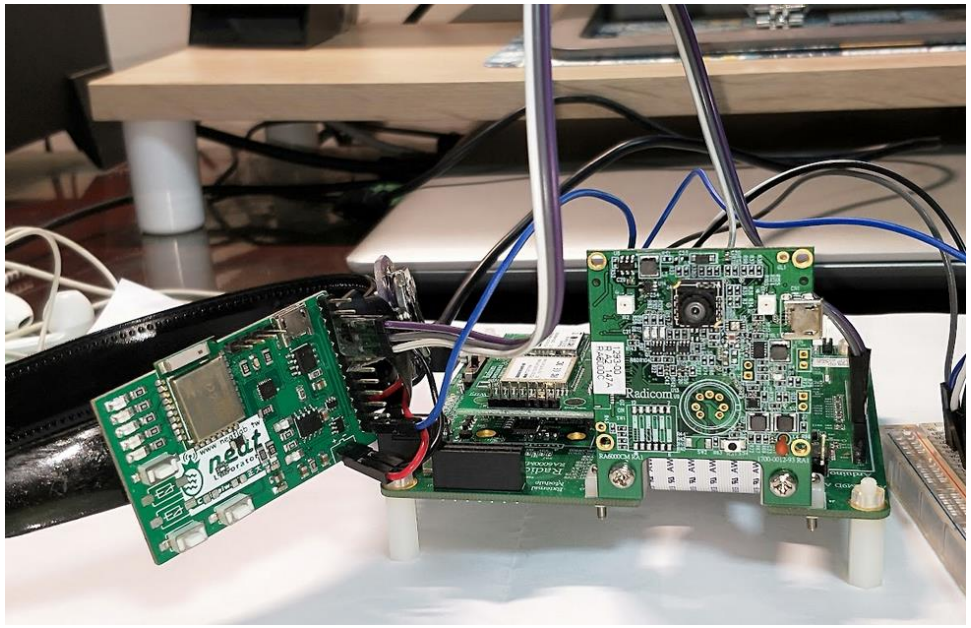
修正預測的結果

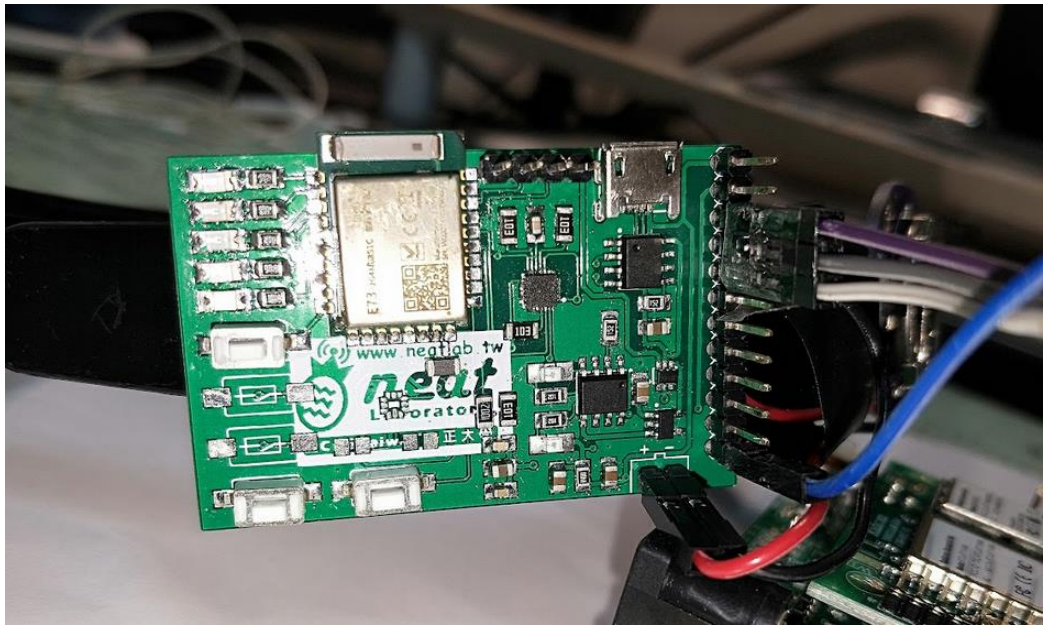
原始預測結果 → 找一個sine波擬合 → 設threshold y=0.5，>0.5變成1，<0.5變成0 → 得到修正後的預測結果 算出著地指數



測試結果

- 作品實體照





髮箍 & 實驗室自製的nRF52840

- 展示影片
  - https://www.youtube.com/watch?v=tPOsxX0Oops

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

總結展望

- 本作品旨在提供一套完整的跑步監測系統，幫助教練分析跑者的步態

- 未來可以
  - 進一步算出跑步著地指數，與教練討論其應用
  - 嘗試移植不同的模型(LSTM、CNN 1D)
  - 增加模型的輸入：身高、體重等
  - 整合其他無線通訊技術達到Real Time monitoring

ARC DESIGN CONTEST 2022

SYNOPSYS®
Silicon to Software™

Thank You