

Term Project Report

電機三 B06901012周奕慧 B06901073羅韻琿

1. Titles

Intelligent Motion Recorder

2. Motivation

穿戴式裝置漸趨普遍, 我們希望透過穿戴式裝置記錄一個人一整天的活動狀況。包括移動、靜止、運動、睡眠等, 以便進行健康管理。

3. Implementation

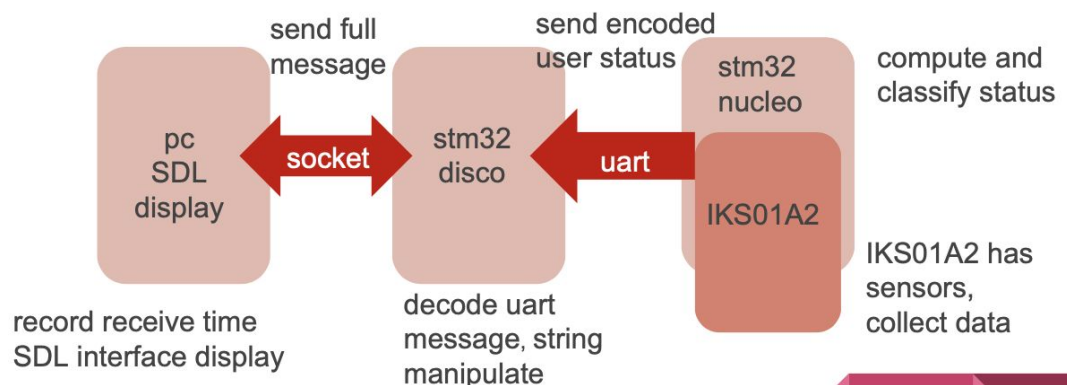
a. Function :

最後實際完成的成品有兩個mode, 分別是日間模式(測量使用者移動狀態, 共8 個狀態, stationary, standing, sitting, lying, walking, running, jogging, biking) 跟夜間模式。夜間模式的結果有分睡著跟還沒睡著兩種。如果還沒睡著的話, 會測未睡著的使用者的移動情形。如果睡著的話就測使用者的翻身行為。將結果用wifi socket傳送到PC。PC會記錄下收到訊息的時間儲存到一個file, 並將結果顯示在使用者介面上。

b. Infrastructure:

主要的溝通架構是STM32 nucleo → STM32 disco → PC server

Infrastructure



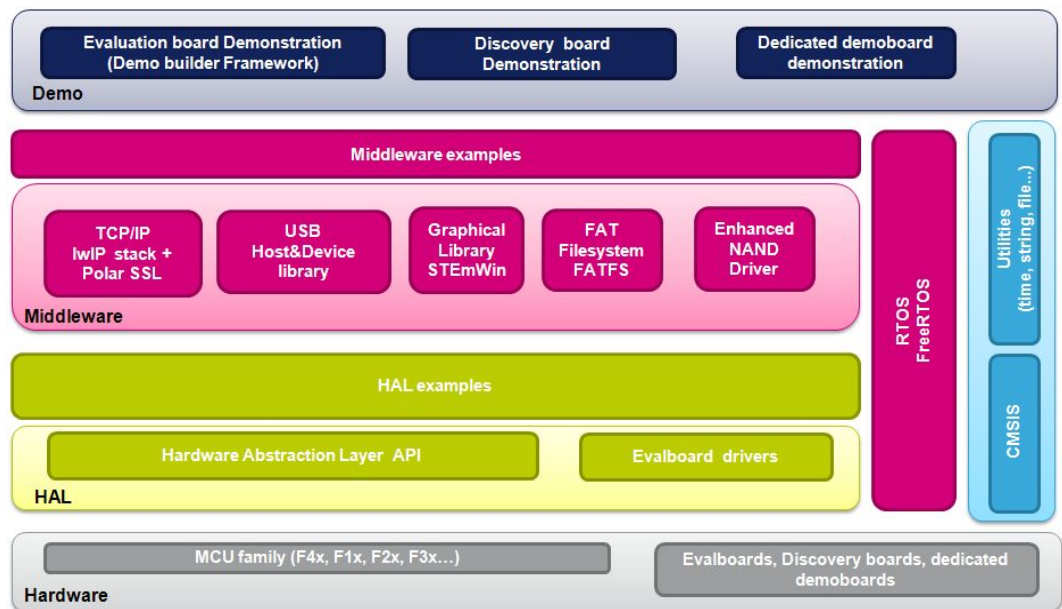
我們使用STM32I476 nucleo 開發板外加IKS01A2疊板進行主要的測量與計算工作。IKS01A2有各種sensor, 我們用到的是LSM6DSL加速度計跟LPS22HB壓力計。IKS01A2負責感測器數據採集, nucleo板子負責處理數據、計算分類使用者狀態, 再將偵測到的狀態用UART傳送給STM32I475 disco 開發板。我們使用nucleo UART3的port傳送資訊

STM32I475 disco 開發板使用UART4的port接收nucleo的訊息, 將訊息解碼, 再用socket 傳送完整字串給PC server。

PC server 會記錄下收到訊息的時間以及訊息內容, 儲存至一個log file, 並將結果display到使用者介面的螢幕上。

c. STM32 nucleo

nucleo 跑的作業系統並不是mbed-os, 他的架構如下



底層是硬體(Hardware), 倒數第二層是HAL(Hardware Abstraction Layer), 再來是Middleware, 最上層就是跑我們的application。另外我們的應用中有用到時間戳記Time Stamp, 需要real time clock,因此我們也有使用RTOS. 我們的開發環境是System Workbench for STM32.

nucleo上跑的演算法是來自Sensor & motion algorithm software pack for STM32Cube.

https://www.st.com/content/ccc/resource/sales_and_marketing/promotional_material/brochure/group0/4d/23/9c/0d/3a/6b/40/ec/ST12274_BRSTMSENS0818_EP/files/ST12274_BRSTMSENS0818_EP.pdf/jcr:content/translations/en.ST12274_BRSTMSENS0818_EP.pdf STM32 Cube 把演算法pre-compile成library 讓人免費下載, 這些library 屬於Middleware的層級。

裡面有非常多package, 包括動作偵測、睡眠偵測、運動強度偵測、健身次數偵測、計步器....。我們測試了多個後留下效果較好的MotionAW activity recognition for wrist(用手腕的活動情形偵測使用者運動狀態, 適合穿戴式裝置) 和 MotionSM sleep monitoring (偵測使用者有沒有睡著)。

我們的Algorithm flow 如下

Algorithm flow

Day mode

- Update motion detection
- Send Uart message

Night mode

- If not sleeping, detect motion
- If sleeping, detect turn over (accelerate z axis value change sign)
- Send Uart message



d. Uart

由於STM32 nucleo 板子上沒有wifi 模組, 所以我們要讓STM32 disco 負責跟PC 進行溝通。STM32 nucleo 跟STM32 disco 兩個板子之間的通訊用UART, 我們實作了單方向的溝通, 由nucleo 傳送資訊給disco Nucleo 使用UART Port3 將訊息傳送出去, Disco 用UART Port4接收訊息。

Uart 溝通面臨許多問題, 最主要的是穩定性低。Uart傳送訊息非常容易掉封包, 因此無法傳送完整的字串。我們改傳送a,b,c,d,e 等表示不同運動狀態。由於運動狀態是連續的, 因此中間掉一兩個封包不會有太大的影響, 因此每更新一次運動狀態就傳送一個封包。對於瞬間性的資訊(比如翻身) 封包遺失的影響會很大, 所以每偵測到一次翻身我們就傳送10個Uart封包。

我們也嘗試傳送過數字, 但是失敗。STM Cube 裡面有一些提供計量資訊的library(比如測健身伏地挺身次數, 計步器, 測睡眠時間)。但因為傳送數字要完整傳送一整個字串, 失敗率實在太高, 因此我們放棄這些功能, 只傳送狀態資訊。

不過偵測睡眠時間我們仍然有找到替代方案, 我們在PC server端紀錄收到每個訊息的時間, 並儲存起來, 因此使用者仍然可以看自己PC紀錄到自己睡著時是幾點幾分, 醒來時又是幾點幾分, 進而推算睡眠時間。

e. **STM32 disco socket client**

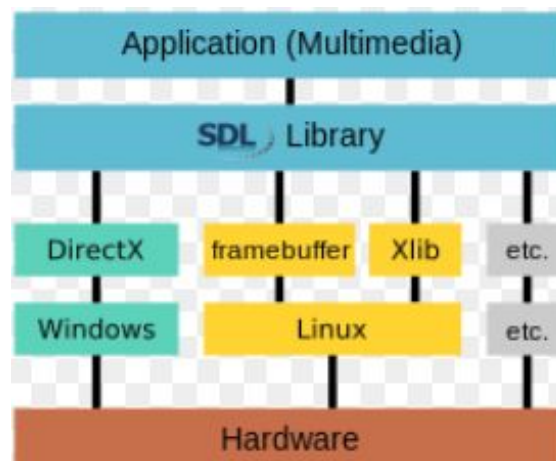
我們將disco和PC都連上手機熱點，以確保兩者的IP網段相同，而且穩定度也會提高。其中，disco作為socket client使用，並指定連線到的PC的IP位址。disco會把剛從nucleo收到的字元解碼成其所對應的狀態，並以字串的形式傳輸出去。

因為socket在傳輸上比UART穩定，所以可以直接傳輸字串。程式碼見Socket/client.cpp。在實作上面，起初有遇到連不上線的情況，估計是因為太頻繁的使用mbed module，通常讓他靜置十幾分鐘，就可以在重新連上線。

f. **PC socket server**

PC作為socket server，會持續listen直到有client連接上，一旦連線建立，PC會傳送一個message給disco，藉由雙向的溝通以確認雙方的連線。接著，PC會輸出disco傳來的訊息並註記現在時刻。

另外，我們藉由SDL (Simple DirectMedia Layer) 做了一個簡單的介面來方便使用者操作，SDL支援Windows, Linux, Mac等平台，以C語言寫成，提供了數種控制圖像、聲音、輸出入的函式，多用於開發遊戲、模擬器、媒體播放器等多媒體應用領域。架構如下圖，只要下載Library即可以開始做開發。



我們將socket server和SDL的程式統一寫在display.cpp，此程式會分析PC收到的message，依照message的內容決定螢幕顯示的照片。編譯方式如下：

```
g++ display.cpp -ISDL2 -ISDL2_image -ISDL2_ttf -o display
```

其中，-ISDL2是要linkSDL的library，而-ISDL2_image和-ISDL2_ttf是針對字串和png格式的圖片。

我們的構想是把PC收到的訊息全部印在畫面上，包含時間和使用者的狀態。但是SDL對於字串和照片上的處理不太一樣，我們可以呈現出時鐘加上背景照片的介面，但是連上socket後會黑屏，目前還不清楚原因。因此我們的替代方法是直接修改照片，讓照片上面顯示文字。

因此，PC一旦收到disco傳來的訊息，就會把畫面換成相對應的照片（該照片上面有文字），terminal同時會記錄下每個時刻的狀態，並存成log檔。

4. Result

a. Moving mode

我們成功偵測到stationary, walking, fast walking, jogging 等狀態。並且用button 切換時十分順暢

b. Sleeping mode

我們能夠順暢的測到no sleeping 時的運動狀態，如果翻轉手也可以立刻測到翻身。另外，當我們靜置STM32 nucleo超過六分鐘時可以測到由no sleeping 變成sleeping。並且no sleeping 最後測到的運動狀態是lying, 十分理想。

5. Reference

Sensor & motion algorithm software pack for STM32Cube Documentation

https://www.st.com/content/ccc/resource/sales_and_marketing/promotional_material/brochure/group0/4d/23/9c/0d/3a/6b/40/ec/ST12274_BRSTMSENS0818_EP/files/ST12274_BRSTMSENS0818_EP.pdf/jcr:content/translations/en.ST12274_BRSTMSENS0818_EP.pdf

Software package source

<https://www.st.com/en/embedded-software/x-cube-mems-xt1.html>

Software package user manual

https://www.st.com/content/ccc/resource/technical/document/user_manual/c8/04/7f/d1/e9/24/47/09/DM00157069.pdf/files/DM00157069.pdf/jcr:content/translations/en.DM00157069.pdf

STM32I476 Nucleo datasheet

https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf

STM32I475 Disco datasheet

https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/b1/b8/7a/f2/f7/8d/4b/6b/DM00347848/files/DM00347848.pdf/jcr:content/translations/en.DM00347848.pdf

SDL tutorial

<https://lazyfoo.net/tutorials/SDL/index.php>

SDL

<https://github.com/NTUEE-ESLab/2019-SmartCushion>