

# ES Lab HW2

Team 10  
EE4  
王敦儒 吳瑋凌

- 所使用的傳感器Modules

- Accelerometer
- Camera
- GPS
- nRF24

- 使用Modules的呈現方法以

在選擇Modules時，我們希望能將資料有趣的呈現在網頁上，因此選擇了一些有數據資料的傳感器，並將他們畫出。由於傳感器的資料是動態得到的(每隔一段時間就會有新資料更新傳入)，所以當伺服器拿到新資料的時候，要將它傳出給客戶端。與作業一不同的地方是：作業一是當使用者按下Refresh按鈕時，瀏覽器會向伺服器拿到最新的資料，而在這次作業我們想到的是：當伺服器端有新資料進來時，就會發送資料給客戶端。實現的方法是用socket.io建立一個雙向溝通的管道，在新資料進來的時候，利用socket.emit發送資料出去，然後在客戶端用d3將點資料繪成線圖形呈現。

d3繪製直線的需要資料格式為一矩陣裡包著含有'x', 'y'座標值的物件，而利用svg.line()取得x,y座標並且interpolate決定繪出來的線軌跡。網頁設計上是將其用表格呈現，三個標題與其三個呈現資料的圖表，其大小均是長寬各250px。

目前使用的情境是：利用兩個tessel板子，其中之一(A)負責收集資料，另一個(B)負責傳遞資料。A上面有Camera(Port B), GPS(Port C), Accelerometer(Port D)三種傳感器，再加上nrf24(Port A)晶片；B上面則有nrf24(Port C)晶片，並需要用到wifi。當A上面收到新資料時，會透過nrf傳送資料至B上的nrf，在透過B上的wifi連線，將B當作中繼站，傳送至遠端伺服器，再經由瀏覽器便可以讀取資訊。GPS和Accelerometer是自動不斷的接收資料，Camera則是必須按下A板子上的Config按鈕方可拍照。

程式碼分為三個部份：rx.js, tx.js, server/

需先將tx.js燒錄在A板(A:nRF24, B:Cam, C:GPS, D:Accel)，將rx.js燒錄在B板(C:nRF24)，並將B板連上網路。瀏覽器的部份需打開220.134.54.147:3000(不確定會不會一直可以連線)

若要在本機上開啟server則需改動rx.js內部的host ip

## 1. Accelerometer

最一開始的打算是要將加速器的x,y,z座標值傳到瀏覽器，畫出3D圖，這樣我們可以手持module畫圖，然後看螢幕上呈現出我們在空氣中畫的圖形。但上網google了一下，發現要在短時間了解並用d3繪製出立體圖形對我來說有些困難，於是只好放棄一維的呈現，只傳x,y的座標資料給client。

## 2. GPS

GPS最直接想到就是要和地圖結合，所以我們想在網頁上顯示出目前的位置，並隨著感應器移動而移動。tessel的GPS模組似乎不太穩定，常常會收不到訊號(可能是在室內?)

## 3. Camera

用Camera Module拍照後，把影像的路徑傳給瀏覽器，再將其填入表格中box的那一欄，如此可將影像呈现在網頁上。照片照下來的檔案有點大，加上是raw file因此如果透過nrf24傳送至B板需要非常久(真的要非常久)

## 4. nRF24

最後一個傳感器選擇用nRF24，將兩個tessel板資料互傳，再透過wifi上傳至遠端伺服器。tessel的nrf24模組速度不是很快，完全無法跟上accelerometer的最慢取樣速度，所以資料會一直累積，而且在A板子移動過後一陣子才會在B板子看到變化

### ● 及撰寫時所遇的問題與其解決方法

- 這次遇到的問題有在伺服器emit資料物件格式給客戶端的時候，不曉得其該如何傳入，還有要如何網頁上呈現我們所想要的照片與線條。解決方法是網路上搜尋關鍵字並參考他人實作經驗來修正。
- 依照node-nrf24的Document所描述，nrf24的rx端可以同時開啟最多6個pipe在listening mode，但是實際使用後發現只要開到兩個pipe以上，前面的pipe都會停止運作，經過一番搜尋，發現node-nrf24的舊版本的確有這個問題，作者在其網站上已經說明解決方法，並在新版解決。但npm install 所得到的nrf24 module並不是最新版，不過使用作者提供的方法便可解決。不過這次實驗的大部分時間幾乎都在解決這個問題……

參考使用畫面



