# Sensor Data Project Report
Jacques Wagstaff

## Project Summary

In this project we combine the sensor data of a mobile phone with data extracted from a weather forecast. The idea is to creat an activity tracking system using the phone's sensors. The system also gives activity suggestions and varying warnings depending on the local weather and activity duration. For example, the system suggests to the user to go wind surfing if the wind conditions are good, or to go for a walk if the user has been sat down for too long. The system also gives a warning if the user spent too much time in the sun. This system could be particularly useful for holidaying users.

In this system the local weather data is collected given the GPS data from a smartphone. This weather data collection occurs on a periodic basis. Data from the phone's sensors is continuously being uploaded to the Dweet.io website. This data can then be read into Node-RED and processed accordingly. The processed data is then visualised using the Node-RED dashboard.

## Setting-up

1. I set-up a local instance of Node-RED on my PC, and learnt how to do basic flows.

1. I set-up my mobile phone so that its sensors can be used as a data source. I set-up the 'Dweet My Phone' application on my phone. This sends sensor data every (0.5-5) seconds to Dweet.io and is saved for 24hrs on their server.

2. On Node-RED I installed the Dweetio node. This listens to messages from Dweet.io, and using a unique identifier (ModelTest01) I can collect the data from my phone's sensors.

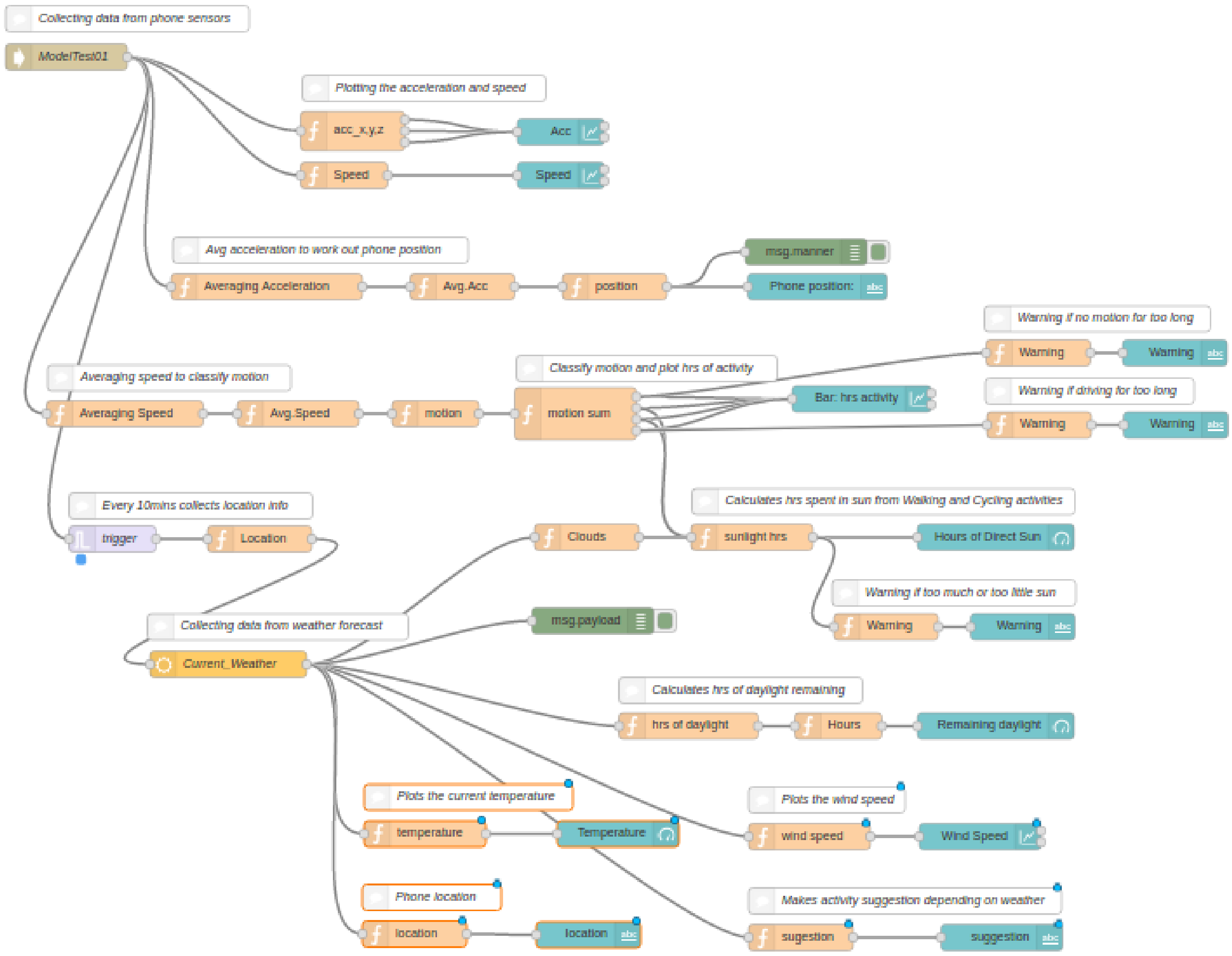
3. I decided to use the Node-RED dashboard package for visualisations, which I installed.

4. I set-up an account for OpenWeatherMap and obtained an API key. I installed the openweathermap node on Node-RED, this polls openweathermap.org for current weather data when an input is received, such as a given location or time.

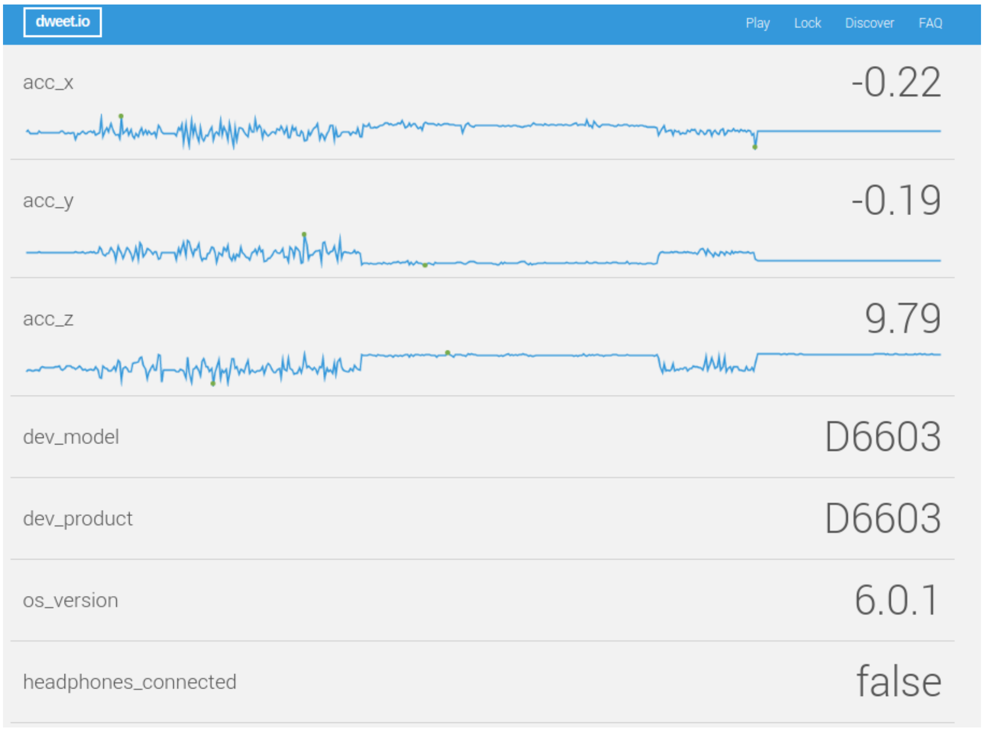
5. I build the system to collect data periodically. The data is processed by a number of different functions, and then visualised by different Node-RED dashboard nodes.

## The System

This is an image of the final full system that I created with the Node-RED tool.


Collecting data from phone sensors
ModelTest01
Plotting the acceleration and speed
acc_x,y,z
Acc
Speed
Speed
Avg acceleration to work out phone position
Averaging Acceleration
Avg.Acc
position
msg.manner
Phone position:
Warning if no motion for too long
Warning
Warning
Averaging speed to classify motion
Averaging Speed
Avg.Speed
motion
Classify motion and plot hrs of activity
motion sum
Bar: hrs activity
Warning if driving for too long
Warning
Warning
Every 10mins collects location info
trigger
Location
Calculates hrs spent in sun from Walking and Cycling activities
Clouds
sunlight hrs
Hours of Direct Sun
Warning if too much or too little sun
Warning
Warning
Collecting data from weather forecast
msg.payload
Current_Weather
Calculates hrs of daylight remaining
hrs of daylight
Hours
Remaining daylight
Plots the current temperature
temperature
Temperature
Plots the wind speed
wind speed
Wind Speed
Phone location
location
location
Makes activity suggestion depending on weather
sugestion
suggestion

Here is an example of the Dweetio output:



acc_x -0.22

acc_y -0.19

acc_z 9.79

dev_model D6603

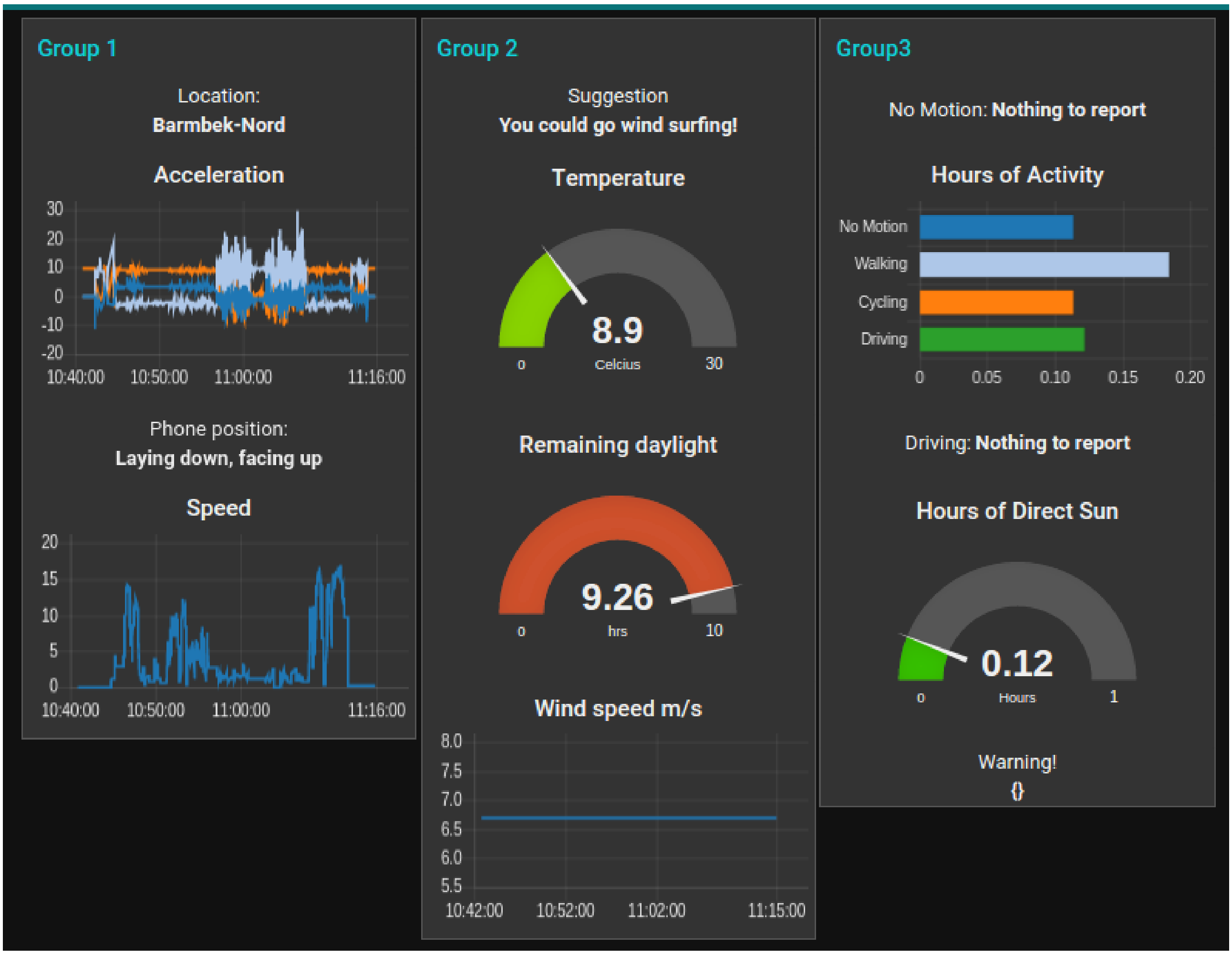
dev_product D6603

os_version 6.0.1

headphones_connected false

## Example Data

This example data was collected from a short drive into town and back.



Group 1

Location:
Barmbek-Nord

Phone position:
Laying down, facing up

Group 2

Suggestion
You could go wind surfing!

Group3

No Motion: Nothing to report

Driving: Nothing to report

Warning!
{}

## The Dashboard

### Group 1

Displaying the location of the phone:
The phone’s GPS signal is fed-into the OpenWeatherMap node, which then gives the appropriate current location name.

Plotting the Acceleration and Speed:
The values from the accelerometers in the phone, in the x,y,z directions, are plotted. I also plotted the Speed values obtained by the GPS.

Displaying the phone position:
Here we calculate the averaged acceleration (z-direction) to work out the phone’s position (laying down, facing up or down, or unknown position). The idea here is to use the acceleration due to gravity to work out the laying position of the device. For this the acceleration is averaged over a chosen period of time (15s), this is to smooth out the variations a bit. Depending on the averaged

acceleration, we can determine if the device is laying face-up or face-down. The position of the phone is then displayed on the dashboard.

**Group2**

### Displaying a Suggestion:

This notification makes use of the current weather data to suggest an activity. If the wind speed is large enough, the system will suggest “You could go wind surfing” (this could be for a sports/activity app). Other suggestions depending on the weather are going to the gym if the weather is bad, or going for a walk if the weather is good.

### Displaying the current temperature and the hours of daylight remaining:

We visualise the current temperature, obtained from the weather data, with a gauge. We also calculate, from the ‘sunset’ time data, the hours of daylight remaining for the current day. This is also shown in a gauge.

### Plotting the wind speed:

We visualise the change in wind speed, obtained from the weather data, in a plot. This data is collected on a periodic basis.

**Group3**

### Hours of Activity:

We calculate the averaged GPS speed over a chosen period of time. Then we classify the type of activity (No Motion, Walking, Cycling, Driving) by the averaged speed. For example, if the averaged speed is below a small value, we assume that the activity is “No Motion”. Over the course of the day, all activities are summed up. The total hours of each activities are then shown in a bar chart.

### Hours of Direct Sun:

Here we assume that if the activity is either Walking or Cycling that the user is outside. We add up all the time the user spends outside throughout the day. Also from the periodically collected weather data, the “clouds” data is obtained for each chunk of activity time. This gives the percentage cloud coverage. From these two data sources we calculate the time spend in direct sunlight (this is a very simple model). This is displayed in a gauge.

### Warnings:

Here we give three warnings depending on hours of activity and hours of direct sun. If the activity “No Motion” adds up to more than 1 hour, a warning appears telling the user to “go for a walk”. If the activity “Driving” adds up to more than 2 hours, a warning appears telling the user to “take a break”. And finally, if the hours of direct sun is very low a warning appears telling the user to “get some sun”. However, if it adds up to more than 2 hours a warning appears telling the user to “get in the shade”.

## Final Thoughts

Due to the short time I’ve spent on this project so far, the data analysis was not very sophisticated. A large proportion of the time I spent setting-up and making sure that the system works. Ideally, to determine a user’s activity accurately, we would need to collect a lot of data with the activities known as a training set for a machine learning algorithm to find a good model. The very simplistic model I used is of course not very accurate, but I showed that the system works in principle. I started thinking about the patterns in the phone’s acceleration data as well, which can give insights into the phone’s position and of course therefore the user’s activity. This would be an interesting route to investigate further.

The model used for calculating the hours of direct sun is also very simplistic. I thought about calculating something more complex like a UV index or so, to give a better measure, this could be done in follow-up work.

In summary: I started this project to learn the different platforms/software/programming languages needed to work with data collected directly from sensors. I managed to set up a system that collects data from various incoming sources, processes that data to gain some insights, and implement a visualisation solution to show the result.