Damaged Goods

Spec Version 1.6

Document Scope

Purpose: This document describes the functional and physical requirements for a product consisting of electronic hardware, embedded firmware and PC based software deemed to be fit for the purpose below. It will be revised over the semester and updates posted for clarity, cost or component availability reasons.

Summary: Your team has been commissioned to produce a system for the monitoring of sensitive goods while in transit. A portable logger will record GPS derived location data, accelerometer based motion data and temperature (at a minimum). An upload feature will allow capture of the data over any internet connected device. Specifically, for this project a PC based software system will display this data overlayed on Google Maps with visualisation of the variables a key feature.

Background: Goods are increasingly produced remotely from their point of use. We use many means of transporting goods but pay a cost in terms of damage or spoilage while in transit. A good example is the supply of vaccines to remote locations. Precise temperature control in the laboratory and storage centre is useless if the vaccine is then transported for hours or days in a hot vehicle. There is often no means of detecting the spoilage and the vaccine is delivered in good faith, but it will be wasted effort. The same or very similar scenarios arise for foodstuffs and precious goods. Additionally, physical impact can damage poorly packed goods with legal difficulty in ascribing blame. For example - where did the damage occur when did it occur, and what was the magnitude of the impulse?
A small, portable, battery powered device which travels with the goods is one solution to this set of problems.

Your team is charged with developing, building and demonstrating such a system.

The two sub-systems, the **portable logger** and the **display and analysis software** for this project will be discussed in turn.

Project Requirements

(A) Portable logger
A logging device which records temperature and 3-axis acceleration is to be designed. It will also acquire and record GPS coordinates where coverage exists. This data stream will be put into non-volatile memory (SD card) on the logger. After a transit event has occurred, the SD card will be removed and contents uploaded to a predetermined file server available to all teams. The specifications for the data accuracy, sample time and format are given in this document. Additionally, physical size, battery life and user interface are also given. The device will require embedded microcontroller and the target IC will be from the TI family below.
A development kit will be made available to you which will allow you to test code and functionality before committing to your target IC. Basic specifications for this component are given below in Table 1.

Table 1. Minimum specification of portable logger

Parameter	Minimum Specification
Sample Rate	1 sample per minute*
PCB Size Envelope (Max)	70 x 55 x 25mm (L x W x H) No prescribed enclosure. This is only for your PCB, not the stellaris board. GPS does not need to be included in this limit.
Battery life	Rechargeable single cell lithium / lithium-polymer. Device must acquire and record for 48 hours assuming a GPS lock is available the entire time. Recharge via mini/micro-USB jack.
Battery Size (Physical)	Batteries must conform to a separate size limit of 70 x 55 x 25mm.
GPS module	Prescribed GPS module provided (NEO-6M)
Acceleration	Accuracy: 12 bit minimum resolution (3 axis) +/- 10G, analog or digital parts acceptable
Temperature	3 significant digits data to be in degrees Celsius. Range -20 -> 85C✦ Thermistor, diode sensor only. **No** digital sensor IC to be used. Needs to be accurate to 1C.
Target microcontroller	Tiva C Series by TI
SD card	Micro SD (SDHC or SD)
Target file server	(TBA)
User Interface	USB socket (mini B or micro) for charging, on-off switch, single tri-colour LED, any labelling appropriate for the purpose.

Drop Test	Device must withstand 2m drop test (after product demonstration)

⁺ all parts must be specified to work within this range

* actual sample may be an average reading from any number of samples taken in that minute

In the case that there is no GPS lock:
- You should still perform an acquisition once per minute
- Your software should leave the `latitude` and `longitude` keys out of your sample in the YAML file (see section C)
- Any samples which are missing the `latitude` and `longitude` keys do not need to be displayed as points on the map and may be ignored as far as any mapping is concerned. However, they do need to be processed in any threshold tests.
- If the YAML file contains no latitude and longitude information, simply display a blank map or no map and only perform the good/bad test as outlined in section B.

In the case that there is no GPS lock **and** no time information from the GPS:
- You should still perform an acquisition once per minute (or as close to one minute as possible)
- Your software should leave the `latitude` and `longitude` keys out of your sample in the YAML file, as well as the `time` information (see section C)
- Any samples which are missing the position/time keys do not need to be displayed as points on the map and may be ignored as far as any mapping is concerned. However, they do need to be processed in any threshold tests.
- If the YAML file contains no position or time information, simply display a blank map or no map and only perform the good/bad test as outlined in section B. If the good/bad test fails for a point that has no time or position information, a special notification (message box, blinking label, etc) should notify the user of this fact.

(B) Display and Analysis Software

Basic Software

The software will acquire data from the named server, process the data to permit display of the route, times, accelerometer data and temperature overlayed on the GPS correct mapping display.

The language and development environment is not restricted. The end product must function on the lab imaged 47-210 PCs but may be developed and demonstrated on a PC or laptop of your choice.

The overlayed route will be suitably encoded to show the amplitude of a user defined variable (accelerometer or temperature). In the crudest sense, it will show where on the route a user-defined threshold was exceeded. The thresholds (maximum and minimum) will be

adjustable by the user by slider bar and text box inputs. Acceleration only needs to have a maximum threshold, not a minimum.

Your software should display a simple good/bad boolean (Yes/No, True/False, green light/red light, etc) stating whether the goods have been outside of the threshold or not during the entire journey (in a single YAML file).

It is expected that displayed route colour will encode the amplitude or the chosen variable, but other graphic methods may be made available (eg. isotherms, bar chart)

(C) Online data format

All times are to be specified using ISO 8601 date and time format, and including time zone information (e.g 2014-03-02T23:20:19+00:00); http://en.wikipedia.org/wiki/ISO_8601

Data will be stored in a plain text file using the YAML format: http://en.wikipedia.org/wiki/YAML

File Format Description

Sensor	YAML Tag	YAML Type	Units
Latitude	`latitude`	`Element`	N/A
Longitude	`longitude`	`Element`	N/A
Temperature	`temperature`	`Element`	Celsius
Acceleration	`acceleration`	`List [x, y, z]`	m/s^2

An example file can be found here: http://goo.gl/y7OH5N. For formatting, all files are to use four spaces as the indent. The debug section of the file can contain anything you wish, but it must be valid YAML. You may safely ignore this section in your software. All sensors should record information to three significant digits. Lat/Long should have six figures after the decimal point.

(D) The cloud

The online upload/download store of cloud data has been provided by the EAIT IT group. Each group has a small amount of space (50-100MB) on the server `moss.labs.eait.uq.edu.au`. The server can be accessed via SSH/SCP/SFTP with your student credentials.

All of your uploaded data files are to be stored in `/home/groups/engg4810g/engg4810g`**xx,** where **xx** is your team number. You also have

read access to the entire `/home/groups/engg4810g` directory so that you can import data from your colleagues' repositories. Some notes:

- You only need to parse files with the extensions ".yml" or ".yaml"
- Do not assume that data is properly formatted by other teams. Your software should be robust enough to ignore malformed data files.
- You need to set up public key authentication (http://goo.gl/OO7clg) so that you/we do not need to continuously need to type in your student password. You should make it easy to configure a username and SSH key in your software so that we have the ability to change it on demo day. Please do not submit any keys, bring them to demo day instead.
- Please do not clone data from other teams into your own repository. We will be checking.
- Your software should check at least once every 5 minutes for updated files in the entire cloud directory (including all subdirectories).

As a courtesy, these folders have been mapped to G drive on the lab computers for development purposes. However, you are still required to access the files via SCP/SFTP in your software deliverable.

(E) Device data format

When you use your device, it will fill up the SD card with data. The format of the data is completely up to you. When you take the device to your computer and plug in the SD card, you should be able to open your software, browse to the log files on the SD card, import them and convert them to YAML for transmission to the cloud. Your software does not need to be able to automatically read data off the SD card upon insertion.

Grade Uncapping Features

6
Two additional sensors will be incorporated into the logger and data stream and display software. The sensors will be chosen from the list below:

Sensor	YAML Tag	YAML Type	Units
Absolute pressure	`pressure`	`Element`	kPa
Average sound level (SPL A weighted)	`sound`	`Element`	dBA
Humidity (relative)	`humidity`	`Element`	%
Luminosity	`luminosity`	`Element`	lux
Magnetic field	`magnetic_field`	`List [x, y, z]`	uT

UV light level (any lambda between 250nm-450nm)	uv	Element	mW/cm^2

The logger device will "wake on movement" i.e the acquisition and recording will be dormant until a movement of predetermined intensity commences operation. It should not perform any time-based sampling while in this mode. Your device should return to the lowest power state after 2 minutes or after receiving a GPS lock (whichever is sooner). If you do not receive a GPS lock, the sample does not need to contain latitude and longitude information. If the device is in this mode but never activated by movement, there is no need to write a file to the SD card.

The mechanism for enabling this mode is up to you.

Software will provide a visualisation tool for multiple journeys. This must show a heat/density map overlay which describes how close selected journeys are to exceeding of a set of constraints. You will need to be able to select a folder on moss and your heat/density map should be calculated using all recordings within that folder. For example take the image below.
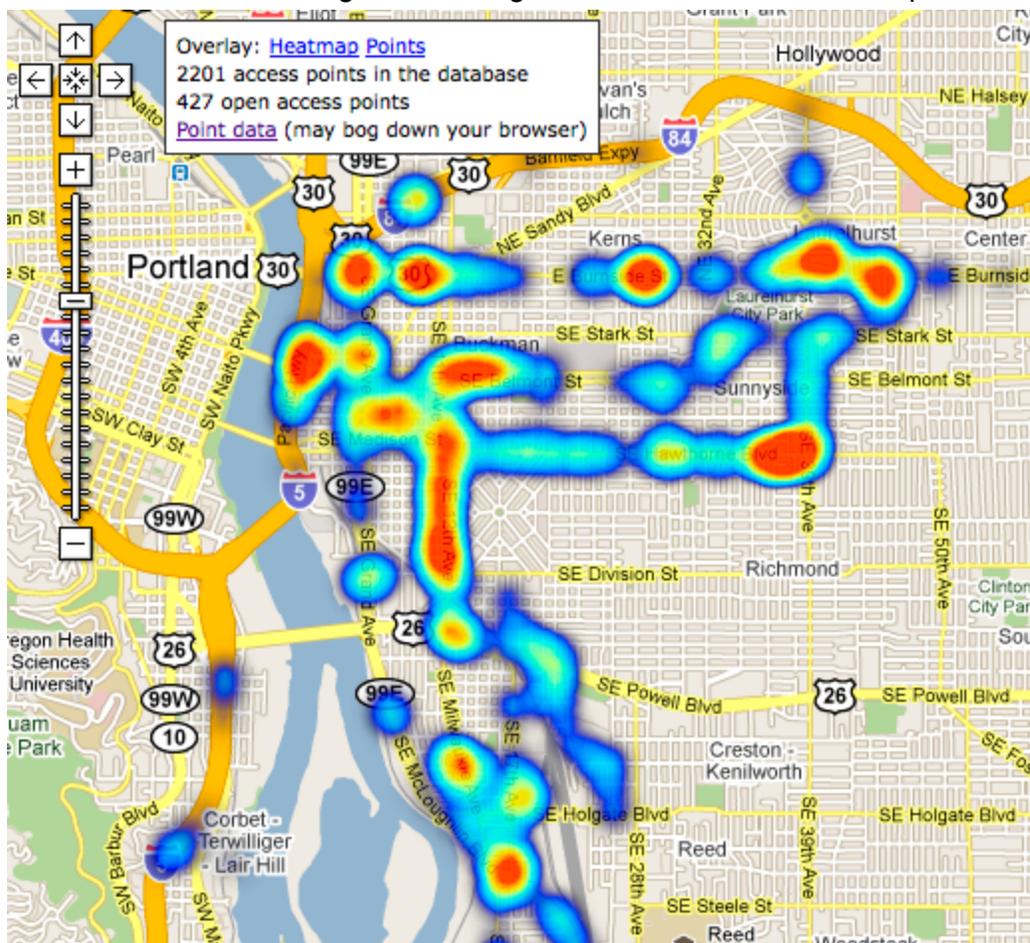
Lets say that 500 journeys have been used. When the package moves through the areas in red, it is often close to going out of bounds (in one or more constraint i.e temperature in some areas and acceleration in others). You must have at least 255 different levels of measurement in your visualisation. The value you use on the heatmap should be the difference between whichever constraint is closest to it's boundary and the boundary itself, measured as a percentage. For example if temperature is 99 degrees with a maximum of 100 and minimum of 0 and the acceleration is 1.5G with a maximum of 3G and minimum of 0 you should mark that sample as 1% because the temperature is within 1% of the upper boundary considering its entire range. In this way a higher value on the heatmap will be better you can imagine this as the 'safeness' rating of these points. Heatmaps should be interpolated over the visible map area. The boundary for each sensor seen in the recordings should be adjustable on the same page. To account for the varying sample densities on your map you should break the visible area into 100x100 boxes and in each box only use the smallest/average value. Whether the smallest or average value is used should be a user selectable option. Once you have done this for every box you should interpolate between boxes using a suitable smoothing function.

7

Features compliant to the 6 requirements listed above are to be implemented.
In addition, a configuration file will be written to the SD card prior to use of the logger. This will allow a user selected preferred route to be uploaded to the logger and an audible and visual alarm will operate when the logger deviates from the desired course by an error margin of no less than three times the accuracy of the GPS module.
This will require a software user interface and real time comparison of GPS derived position and desired position by the portable logger.
A high temporal resolution mode for all or a portion of the route will also be selectable. This mode will place the logger into a one sample per second rate for all sensors except the GPS; the GPS should remain at one sample per minute and the most recent GPS sample used as the position for all subsequent measurements where the GPS is not active. The minimum lifespan of the device is to remain the same as the one sample per minute implementation.

Suppliers
You can order through the following suppliers via ETSG:
- http://au.element14.com
- http://australia.rs-online.com
- https://www.soanarplus.com
- http://littlebirdelectronics.com/ (which stocks parts from Sparkfun and Pololu)

If you order parts from outside the normal ETSG process, under no circumstances will you be reimbursed for any parts from these suppliers.

However, you can include parts in your final design from the following suppliers as long as **<u>all original receipts are kept and submitted with your product (photocopies are not acceptable):</u>**

- http://www.digikey.com/
- http://au.mouser.com/
- http://www.hobbyking.com/

Appendix A - Approved Software Libraries:

To add to this list please post a question on Casper using the course tag:

'software approval'

Given that this list will change more often than the specification, we have decided to move this into a separate document which will be updated live. Please visit the link below:

<p align="center">http://goo.gl/3lzquh</p>