SCHOOL OF ELECTRICAL ENGINEERING
AND TELECOMMUNICATION

Haze Watch: An Android Interface for the Haze Watch Pollution Monitoring System

by

Vishnu Unnikrishnan

Thesis submitted as a requirement for the degree
Bachelor of Engineering (Electrical Engineering)

Submitted: 3rd June 2013
Supervisor: Vijay Sivaraman
Student ID: z3242321
Topic ID: VR 35

Thesis title: Haze Watch: An Android Interface. **Topic number: VR 35**

Student Name: Vishnu Unnikrishnan **Student ID: z3242321**

A. **Problem statement**

Air pollution is a critical issue that faces us as society today. It has been linked to
2.3% of all deaths in Australia and it alone costs the NSW government $4.7 billion
(26%) in healthcare costs alone. Although the costs of air pollution in the social,
economic and environmental spheres are well known, the government's monitoring
strategy in the greater Sydney region consists of 15 monitoring sites for an area of
12 000km2. Haze Watch aims to address this problem by crowd sourcing pollution
information. This is done by distributing an inexpensive pollution monitoring device
which sends pollution data to a mobile phone which uploads it to a remote server.

B. **Objective**

The objective of this thesis to design and build the smart phone application which will
Interface between the pollution monitor and the remote server. It is also to begin the
Field testing process and to analysis the quality of sub systems and to provide
Suggestions on improvements.

C. **My solution**

Mobile Application with UI development
Mobile Application with software development
Field Trials to establish Quality
Analysis of device
Analysis of application
Suggestions for improvement

D. **Contributions** (at most one per line, most important first)

A new intuitive user interface for mobile application
Two tunnel loss strategies to handle lost GPS signal
A network loss strategy to handle lost network signal
Field trial map planning and run, and analysis.
Analysis of device battery life and device
Analysis of Application location services
Analysis of Smart phone battery life
Analysis of effect on data from above issues

E. **Suggestions for future work**

Revision of Mobile device
Upgrades to application future proof application
Further field trials

While I may have benefited from discussion with other people, I certify that this thesis is entirely my own work, except where appropriately documented acknowledgements are included.

Signature: Date: 06 / 06 / 2013

Thesis Pointers

List relevant page numbers in the column on the left. Be precise and selective: Don't list all pages of your thesis!

8	Problem Statement
8	Objective

Theory (up to 5 most relevant ideas)

9	Air pollution
11	Monitoring
12	Haze Watch
13	Mobile development
15	Quality Management

Method of solution (up to 5 most relevant points)

20	Mobile Application with UI development
23,28,29,31	Mobile Application with Software development
35,37,38,41,43	Field Trials to establish Quality
46	Analysis of device
48	Analysis of application
56	Suggestions for improvement

Contributions

20-22	A new intuitive user interface for mobile application
27-28	Two tunnel loss strategies to handle lost GPS signal
33	A network loss strategy to handle lost network signal
35-45	Field trial map planning and run, and analysis.
46-47	Analysis of device battery life and device
50-51	Analysis of Application location services
49-50	Analysis of Smart phone battery life
53-55	Analysis of effect on data from above issues

My work

23,25,32	System block diagrams/algorithms/equations solved
20, 23, 35, 46	Description of assessment criteria used
35-45, 26, 29, 34	Description of procedure (e.g. for experiments)

Results

26, 29, 34, 35,37,40, 42, 45	Succinct presentation of results
26, 29, 34, 38,42,45, 46-55	Analysis
45, 48,49,55, 34,29,26	Significance of results

Conclusion

57	Statement of whether the outcomes met the objectives
56	Suggestions for future research

Literature: (up to 5 most important references)

4, 8, 11	World Health Organisation, (2008)
4, 8, 11	NSW Department of Health (2009)
20-22	Android Developers (2013)
29-30	Judy Liu (2012)
12	Matthew Kelly (2012)

Abstract

Air pollution is a significant problem all around the world, over 26% of the NSW health budget is spent on air pollution related diseases and 2% of all deaths in Australia[1] have been linked with diseases of factors related to Air pollution. This combined with the fact that the government has 15 monitoring stations for an area of over 12,000 km^2[2] shows that Air pollution is a serious problem that has not been seriously dealt with by the government or other parties. The Haze Watch project aims to rectify this issue by crowd sourcing pollution data. This is done through by distributing cheap exposure measurement devices to the community. These devices communicate with a phone and this phone uploads data to the HazeWatch server.

My thesis is composed of three sections. The first was to build the mobile application which would collect, calibrate and transmit data from the device to the server. The second to run field trials using the system and to see how well it worked and how accurate the system is, and finally I was to analyse the system for quality factors and to suggest improvements to the system.

I have successfully built the android application which communicates with both the device over bluetooth and to the server over the network connection. This application also provides timestamps and geotags in all situations and can handle various changes in the network state. The field trials using the device and application have shown a strong correlation between it and the commercial meter and has shown quite significantly that the major cause of air pollution and in this case carbon monoxide is traffic congestion. Finally the quality factors of this system have also been evaluated. The device can run for 56 hours before batteries need to be changed the worst case time between data gather point and location gather point is 40 seconds, and further quality factors can be found in the corresponding section. Improvements to this system can include creating

I have met the objectives that have been set out in my thesis. The application runs successfully, I have field tested this system extensively and I have analysed the system to find where its limits and where it excels. I have also suggested improvements to the system as a whole.

Acknowledgements:

I'd like to thank all the members of the Haze Watch project. I would like to thank Jason and Oscar who have contributed substantially so that we could test our system.

I would also like to specially thank my supervisor Dr Vijay Sivaraman who's guidance and support helped make this thesis possible. I wish him and the Haze Watch team the best of luck.

Contents

1 Introduction:

Clean air is a critical natural resource that is fundamental to maintaining a safe and healthy life for not just humans but for every living creature. Unfortunately due to the industrialised nature of society air pollution is a significant issue around the world. In Australia alone between 100-1500 deaths are directly linked to air pollution every year[3], and the NSW department of health estimates that it spends over $4.7 billion or 26% of its annual budget on treating air pollution related diseases and other factors[1].

1.1 Problem Statement:

Although the costs of air pollution in the social, economic and environmental spheres are well known, the government's monitoring strategy in the greater Sydney region consists of 15 monitoring sites for an area of 12 000km^2[2]. Haze Watch aims to address this problem by crowd sourcing pollution information. This is done by distributing an inexpensive pollution monitoring device which would send pollution data to a mobile phone which uploads it to a remote server.

1.2 Thesis Objectives:

The objectives of my thesis are three fold.

- Design a Mobile Application:
  The HazeWatch project requires a mobile phone to interface the pollution device to the remote server. The mobile application must translate and calibrate the pollution data received and then timestamp and geotag the data before uploading it to the server. All of these functions must work accurately in all reasonable situations.

- Run Field Trials using the System:
  By building the mobile application the system will be complete and it becomes necessary to field test the system. This will show how accurate the system is as well demonstrate how well the system handles changing environmental conditions and from these trials we will be able to identify system wide issues.

- Analyse System Quality and suggest improvements:
  The final objective of my thesis is to analyse the quality of the system and its individual sub systems. By conducting experiments designed to test the limits of these sub systems we will be able to identify specific issues that may not otherwise evident. These issues can then be resolved by the improvements then suggested.

2 Background:

The background will cover four major areas. There will be a look at air pollution, the types, its causes and effects, followed by a look at current monitoring strategy and the background of the Haze Watch project. Finally there will be a look at the mobile platform that has been chosen for application development and finally an overview of modern quality management processes.

2.1 Air Pollution:

Air pollution is the occurrence of gases, particulates or other matter in the atmosphere in concentration that is harmful to humans and other living organisms, or in concentrations that can natural landforms or buildings[4]. Air pollution is a serious world wide issue that is set to worsen into the future unless steps are taken to resolve the issue.

2.1.1 Causes of Air Pollution:

Air Pollution is caused or worsened by a variety of factors both natural and man made. The natural factors include volcanic eruptions and bush fires as a primary source, while terrain plays a secondary role in worsening air pollution. Human causes of air pollution are a result of road transport , factories and other industrial processes.

Figure 1: Sources of Pollution (a.Volcanoes[5], b.Bush Fires[6] c.Factories[7], d.Cars[8])

2.1.2 Types of Air Pollutants:

There are many types of air pollutants. The Haze Watch project at the moment measures two types, Carbon monoxide and Nitrogen Dioxide. Both of these are gaseous pollutants and are produced as a by product of combustion. Therefore they are produced by all the major sources listed above.

Carbon Monoxide (CO): Carbon Monoxide is a colourless and odourless byproduct of incomplete combustion. Since combustion is a fundamental reaction undergone that is required for most sources of industrial activity. As such it is a common pollutant and is produced by automobiles, factories and from natural sources such as bushfires.

Carbon Monoxide in high concentrations is extremely toxic for human beings and other complex lifeforms as it combines with hemoglobin in the blood and the resulting product carboxyhemoglobin is unable to transport oxygen to cells around the body and this can result in death. Concentrations around 600ppm can result in coma, seizures and death.

Nitrogen Dioxide (NO2): Nitrogen Dioxide is a reddish brown toxic gas that has a sharp bitter taste. It is a byproduct of internal combustion and as mentioned earlier is caused by automobile use.

Similarly to Carbon Monoxide Nitrogen dioxide is a toxic gas that at low concentrations is an irritant and at high concentrations can result in death. Long term exposure to moderate levels of Nitrogen dioxide can have physiological effects such as decreased lung function and acute or chronic bronchitis.

AQI	NITROGEN DIOXIDE	
	Sensitive Groups	General Population
VERY GOOD 0-33	Ideal conditions for outdoor activities	Ideal conditions for outdoor activities
GOOD 34-66	Ideal conditions for outdoor activities	Ideal conditions for outdoor activities
FAIR 67-99	No health impacts expected when air quality is in this range	Can enjoy outdoor activities
POOR 100-149	People with asthma should watch for symptoms.	No need to modify your usual outdoor activities, unless you experience symptoms.
VERY POOR 150-200	People with asthma should watch for symptoms. If you have symptoms you should rest and use your reliever medicine and/or seek medical advice.	Anyone experiencing discomfort such as coughing or throat irritation should consider reducing physical exertion outdoors or rescheduling strenuous activities to periods when the index is lower.
HAZARDOUS 200+	Avoid outdoor exertion and stay inside as much as possible. If you have symptoms you should rest and take your reliever medicine.If symptoms persist, seek medical advice.	Everyone should avoid outdoor exertion and stay inside as much as possible.

AQI	CARBON MONOXIDE	
	Sensitive Groups	General Population
VERY GOOD 0-33	Ideal conditions for outdoor activities	Ideal conditions for outdoor activities
GOOD 34-66	Ideal conditions for outdoor activities	Ideal conditions for outdoor activities
FAIR 67-99	No health impacts are expected when air quality is in this range.	Can enjoy outdoor activities
POOR 100-149	People with heart disease, such as angina should reduce heavy exertion and avoid sources such as heavy traffic.	No need to modify your usual outdoor activities, unless you experience symptoms.
VERY POOR 150-200	People with heart disease, such as angina should reduce exertion and avoid sources such as heavy traffic. If you have symptoms you should rest and use your reliever medicine. If symptoms persist seek medical advice.	No need to modify your usual outdoor activities, unless you experience symptoms.
HAZARDOUS 200+	People with heart disease, such as angina should avoid exertion and sources such as heavy traffic.	Everyone else should reduce heavy exertion.

Figure 2: AQI Tables and Effects[2]

2.1.3 Effects of Air Pollution on Humans:

The effects of air pollution are wide and varied. However its effects on humans need to be seriously considered. According to the World Health Organisation 1497[3] deaths or 2.3% of all deaths in the year 2012 was linked to Air pollution related illnesses. These illnesses include, lung cancer and cardiopulmonary disease. Pollutants also aggravate diseases like Asthma and can worsen other respiratory problems. This trend is even worse in other countries as they rapidly industrialise.

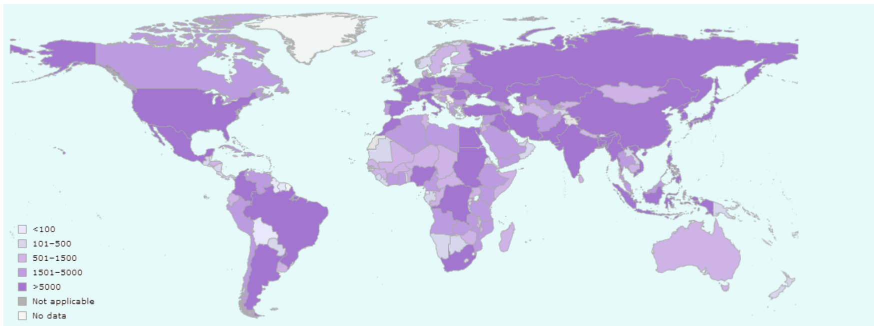


Figure 3: World Wide Pollution Deaths[3]

Air pollution can also have a negative effect on man made structures. For example excess carbon dioxide is filtered from the atmosphere through the formation of acid rain which dissolves calcium carbonate based structures. These include marble buildings like the Taj Mahal and limestone surfaced building like the pyramids of Giza.

2.2 Monitoring Strategies:

2.2.1 Current Strategy

The current monitoring strategy in the Sydney Region consists of 15 monitoring stations that cover 12000 square kilometers. They measure all of the above mentioned pollutants. The major issue with this measurement strategy is that it is inaccurate due to the poor spatial data collected.

Figure 4: Current Measurement Locations[2]

This means that to cover the entire region complex mathematical techniques like interpolation have to be used which can be inaccurate due to the different topographical and man made conditions. These can include the presence of hills, valleys, tunnels and airports. Further the measure used in Sydney only updates the information over an 8 hour period, leading to there not only being a lack of spatial data collection but also a lack of temporal data collection.

2.2.2 Haze Watch:

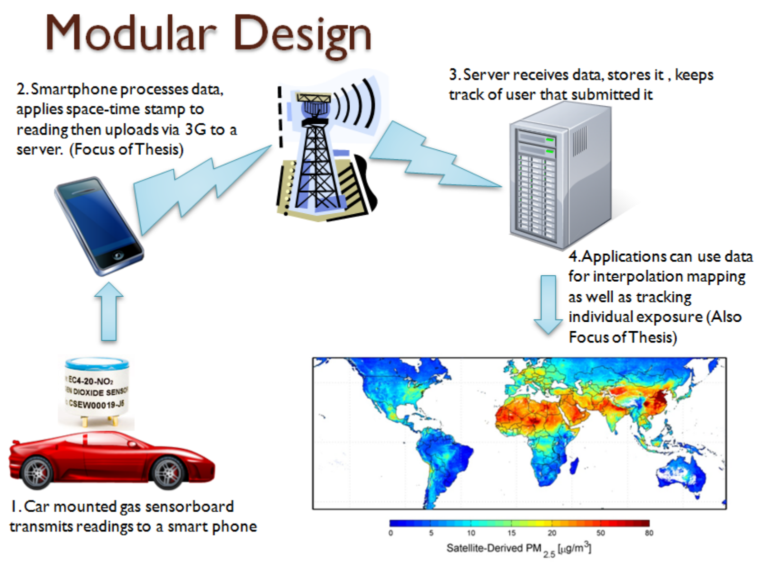
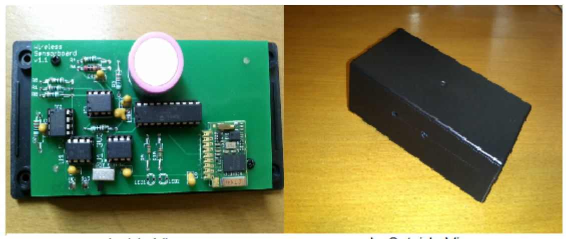


Figure 5: Haze Watch System Overview[15]

The Haze Watch project began in 2010 with the aim to collect and provide localised and temporally accurate pollution data. This project consists of three modular stages, a measuring device, a mobile application that communicates between the device through bluetooth and to the server over a network connection. The final stage is the server, which collects, processes and analyses the data.

Bluetooth Device: The bluetooth device has gone through two revisions. The first model, used contained a metal oxide sensor and measured three gases, carbon monoxide, nitrogen dioxide and ozone. The device contained a basic microprocessor and a bluetooth module. This bluetooth module was used to minimise the cost of the device as it would be able to communicate to the server through a mobile application and the mobile network.

The second revision of the device was developed because the original device was not very sensitive. This new device used electro-chemical sensors which was not only more sensitive but had a linear calibration equation. This device used a pic microntroller to communicate between the sensor and the bluetooth module for the same reason as the original device.



a. Inside View b. Outside View

Figure 6: Bluetooth Device

Mobile Application: The mobile application is the first part of this thesis and interfaces between the bluetooth module and a web server. The application can access the location services, the timing functions, and cellular radios present on the device. By using these functions present on the mobile phone the cost of the device can be minimised because these services would not be needed on the said device. This mobile application was also required to translate the raw data into information.

As the device entered into its second revision this mobile application also had to be rewritten to communicate using an entirely new communications scheme as well as to create new strategies for handling the different situations that have arisen. This rewrite as mentioned forms the first part of this thesis, and analysing the quality of the application forms part of the second part of this thesis.

Server: The server collects the data from all of the devices through the mobile application. The server then analyses this data and provides a web interface which can serve pollution information. This includes a map of pollution information over time. The server will also provide the calibration constants of a device to the phone if the device id is provided.

2.3 Mobile Development Environment:

There are two major competitors in the smart phone industry. These are Apple with its Ios based devices and Google with Android. Therefore due to limited resources a choice had to be made upon which environment would host the newly developed mobile application which would interface the pollution device to the server. For the following reasons outlined below the Android operating system was selected.

- Majority market share:
  Android platform has a 63% share of the smart phone market and this is expected to grow as more people switch over to smart phones.[9]

- Wider Socio-Economic spread:
  Android runs on low-cost smart phones as well as expensive phones and as such is expected to include a wider demographic of people.This leads to an even spread of data gathering locations as different socio-economic groups live in different areas.

- Lack of Access to Apple's bluetooth stack:
  Apple's bluetooth application programming interface (api) is not publicly available and permission to access the api has to be received from Apple before developing ios applications using bluetooth.

Android programs are developed using the Eclipse IDE. The Android operating systems supported by this application are Android 4.0 (Ice Cream Sandwich and up) because the bluetooth stack on Android 2.3.3 (Gingerbread) and below do not support the serial port device bluetooth profile used by this device.

2.3.1 Previous Haze Watch Applications:

Two previous applications were developed at an earlier point in time for the old Haze Watch project. These applications worked with a previous version of the device and was not compatible with the latest device. The first application is very simple and is unintuitive as it requires the user to enter mac addresses of devices not found on the list. The second application uses a different method, and selects the serial port device from the bluetooth discovery list. This is problematic when there is more than one device.

The lack of compatibility and other significant short comings has required a complete redesign of the application.

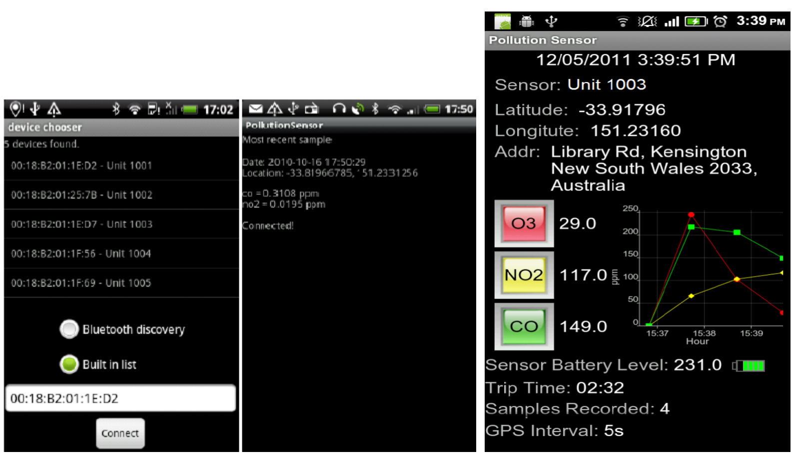


Figure 7: Nikolaus Youdale's[10] and Dawei Lu's[11] Android Applications

2.4 Quality Management:

Quality management is an integral part of any project. A product is considered to be of good quality when it is successful. However success isn't defined by completion. It is an ongoing process that only ends when a product's life cycle is over. This means that that the traditional measures such as completion within the required parameters are not valid identifiers of success.

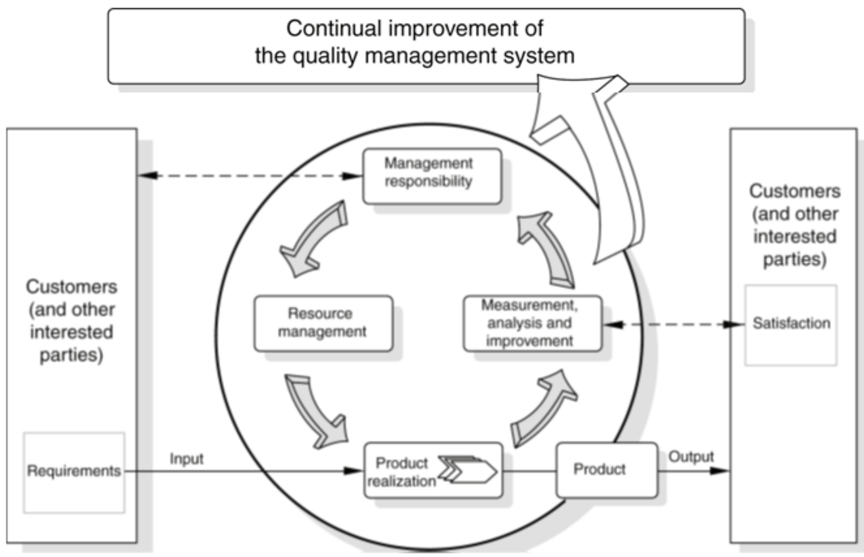


Figure 8: Quality Management Product Life cycle[12]

The above figure relates to the ISO 9000, which is a quality management for a process based system. The ISO 9000 standard was created to help organisations meet the expectations of stakeholders. These stakeholders include not only consumers but also society, regulatory authorities, and other groups such as suppliers. By using this standard for analysis of the system, we can establish that the system is functioning accurately, efficiently and as per the users needs.

As shown in the diagram above, customers or in this specific case the users of the system are present on both sides of the diagram which suggests that rather than being passive consumers they are actively involved in the design and development of the product. The input from users is fed into the product creation life cycle, where engineers implement suggestions as well as the hard requirements into the product. This product is then used by the consumers and their continued suggestions are fed back into the system for future product revisions.

The product life cycle is also of interest. It is a circular process rather than the traditional linear process of design and realisation. In this system management assigns resources to realise the product and although a product will be delivered at this stage, there is another process which is to measure, analyse and improve the product, and the results are delivered to management who are then empowered to assign what resources needed for the future revision. To implement this cycle the most important factor is to measure and analyse the how the system performs.

In the Haze Watch project the analysis will be applied to the device, the mobile application, and the data that has been gathered. By breaking down the project into these three areas it will be easy to establish where the specific issues that exist lie and where the successes are and more importantly how these success can be leveraged as well how the issues inherent in the system can be resolved.

3 User Interface Development:

The user interface is an important part of any mobile application as its the first thing a user will see and interact with in an application. This means that a user interface must be comply with the standards of the platform it runs on, it must not be intrusive and it must be intuitive. The Haze Watch data collection application aims to achieve all three of these objectives.

3.1 Initial Designs:

The first stage of the design process was to come up with a design that met the above three criteria. The first design that was designed is outlined below.

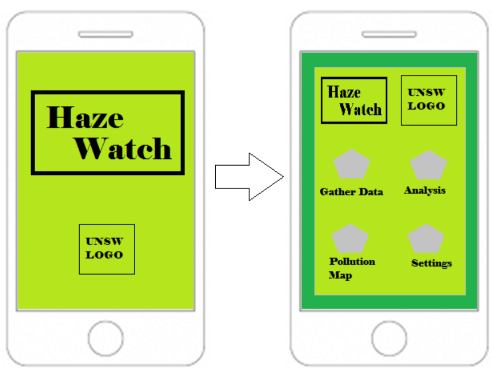


Figure 9: Initial Android Application User Interface

In this design a home screen would initially open to greet the user, and it would hide the start up activity (in next section). When the activity completed it would open to the second screen in the image above and four options would be presented to the user. These are gather data, analysis, pollution map and a settings button. This design aimed to maximise efficiency by making available to the user all the functions that they might have needed. It was also a simple layout with bright colours and featured the logos of both the Haze Watch project and UNSW.

Before this design was implemented it was decided that providing a driver with access to a map was not necessary, it added extra complexity which would not be used. Further to this the Android best practices guide disapproved of using a home screen. It recommended that users be able to access the UI functions of the application on launch and any background tasks should be conducted seamlessly and without affecting user behaviour.

The redesign is shown below.

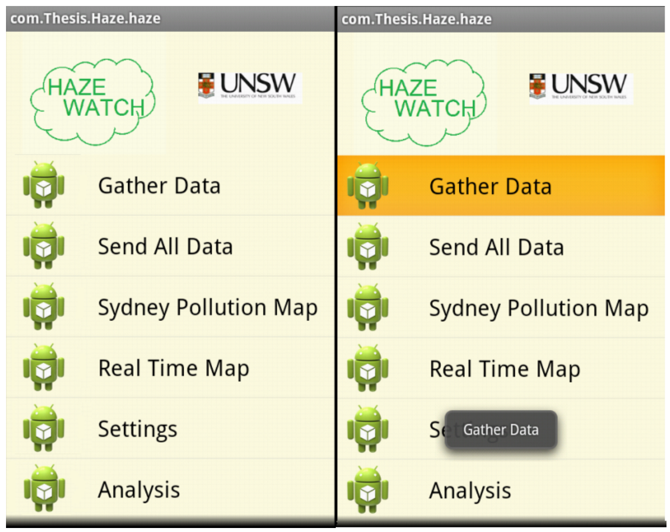


Figure 10: Second Android Application User Interface

The second design seen above was an attempt to fix the issues that were identified with the initial design. The first step was to meet the requirements of the Android best practices guide. Under this system a list view was used to give the options previously represented in buttons. When an item was clicked it would lead the user to a page which would display the relevant information.

This application is intuitive as on a touch it highlights the selected item and on click it launches a floating label with the button that has been selected. The application is also very easy to use as well because it only required a single click to launch the application. The issue with this application was it was deemed to have too many functions and another revision of the application was proposed. The above design was demonstrated in the Thesis A presentation.

The third design although not pictured took a highly minimalist approach. In this design a drop down list provided the options of which device to choose and the information was displayed after hitting connect. The idea behind this was for users to only see the application when they started the data gathering service. This application although simple, non-intrusive and easy to use, was not considered user friendly or because there was no way to stop data gathering, it was also considered to be non-intuitive.

Further comments from users called for a more intuitive and user friendly application, which provide a real time data feed rather than just being non-intrusive. This resulted in the final design presented below.

3.2 Final Design:

The final design was an evolution of all of the above designs. It combined the intuitiveness of the second revision, while adding the non-intrusiveness of the third design and it added more information without creating a sense of information overload. This design also complied with the newest set of android design guidelines which was released after the old interface was designed. The design is shown below.

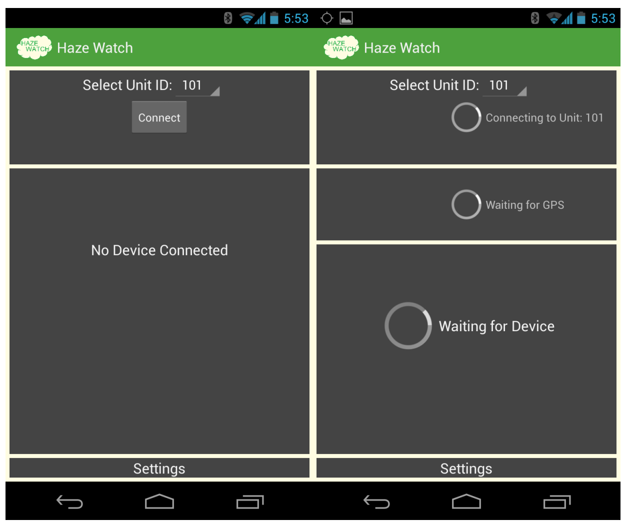


Figure 11: Final Android Application User Interface (a, b)

In the figure above, the first image (a) is the first thing the users see. The user has three options, to select the device id from a drop down list, a connect button so the user can connect the application to the device. The final option is the settings button. This button is placed in a way which will prevent inexperienced users from changing the default settings.

Part b in the above figure, shows the screen after the button has been pressed. It informs the user that it is waiting for device connection, current GPS coordinates, and finally device information. The user is constantly kept aware of what is happening.

In the below figure part d is shown if a bluetooth device can't be found. The UI addition is a floating text box which informs the user that device detection failed. When device detection succeeds part c is seen. This displays the calibration data for the device, the time, the location data and the location address. It also displays cards with pollution measurement. They are colour coded with blue indicating low values, green indicating medium values and red indicating high values. The latest value is also shown in a larger sized text.

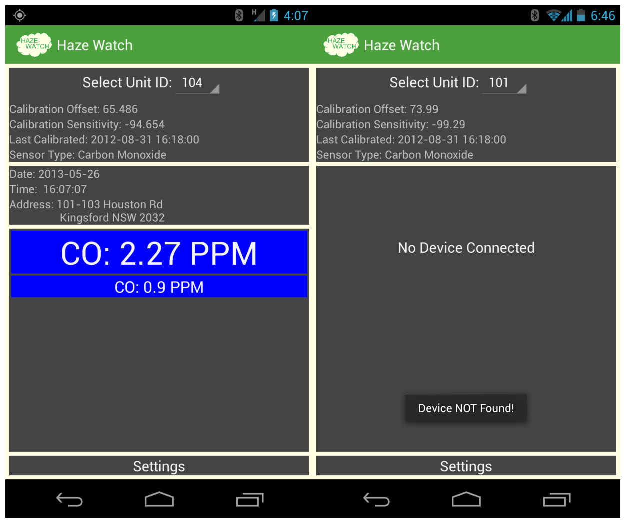


Figure 12: Final Android Application User Interface (c, d)

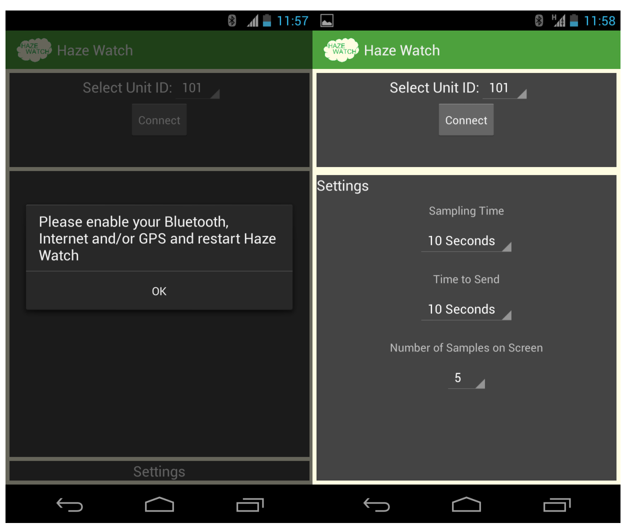


Figure 13: Final Android Application User Interface (e, f)

Part e shows what happens when a user tries to access the application without enabling bluetooth, a network connection or a gps connection. This is to prevent useless data from being collected. Part f is what happens when the settings button is pressed, it creates a pop up window which shows the three setting that can be adjusted by the user.

This has demonstrated the major UI choices made, in the next few sections the application UI will be analysed to see how they meet the criteria that was set out above.

3.3 Final Design User Interface Analysis:

3.3.1 Compliance to Guidelines:

There is no set of concrete rules that must be adhered to in terms of UI design, that is mandated by any organisation. However Google provide a set of Android development UI guidelines[13] that suggest the use of certain concepts when designing the user interface of an application. The major suggestions include not using an explicit menu button, using standard themes and icons and the back button must end an application, while the home must pause the application.

- Menu Button:
  The application I have developed does not make use of a menu button whether software or otherwise. However it explicitly uses an Action Bar which has been considered a more user friendly replacement for the menu button. It should be noted that this feature is only available in Android 4.0 and upwards.

- Standard Themes and Icons:
  The current application it uses a modern android holographic theme as the base for development. This puts it within the recommended thematic constructs of the Android operating system. To continue to make sure this application meets future changes, the application uses a cards based user interface with all data delivered in contextually similar cards. This is similar to Google's most actively developed applications like Google Now and it will ensure that the current trend towards a card based interface will be supported by this application.

- Standard Button Function:
  The device level back button is required to move back one activity or to kill a process if it is the base activity of that process. This is achieved in my application because a pop-up occurs automatically asking the user if they want to kill the process and the background service. The Home button has not been overridden and it pauses the user interface but not the background service. This is required for the correct functioning of the application.

3.3.2 Intrusiveness:

Intrusiveness is a qualitative measure about how much a user is annoyed by a design. As this measure varies by user it is challenging and impossible to provide a concrete scale to measure this concept. Hence the measure must be compared to a set of criteria. The first criterion is how often does the user have to interact with the application, the second is how many menus are present, and finally if the application needs constant attention from the user.

- User Interaction:
  The user in this application is only required to select the device id and press the connect button, all of the other options are preset and hidden away. This 2 step process is the minimum interaction the user is required to perform. This is superior to previous approaches which required manually entering the bluetooth MAC address into the application in the past application, or searching the bluetooth discovery list for a bluetooth device.

- Number of Menus:
  In this application there is only one menu which is the settings menu which has been preset and hidden away. This contrasts with the earlier versions and designs which have many menu options to select and further sub venues to access. The lack of menus in my application reduces the time the user has to spend with the application and as such does not feel that the application intrudes upon them.

- User Attention:
  This is different from the first criterion because the first requires interaction, this one only refers to attention.This application was built in a robust fashion so the user does not need to pay attention to the application while it is in operation. The application will provide alerts to the user on top of any other application that the user runs in a minimalistic way. This is done through floating text boxes that remain on screen for 5 seconds.

3.3.3 Intuitiveness:

Like intrusiveness, Intuitiveness is a complex and qualitative measure. However there were some basic ideas that exist in design for something to be intuitive. These are the application should follow the standards already established by other applications, It should have clear information in regards to processes and whether the users need it, frequently used information should be at the top and there should be a logical flow of information and terminology should be consistent.

- Established Standards:
  As mentioned earlier this application does meet and follow the guidelines established by Google.

- Clear and Concise Information:
  This application presents floating text boxes when something happens in the background service that the user should be aware of, for example if there is a disconnection of the device before stopping the service or if no device could be found. In the main screen itself, the user is presented with information both verbally and visually by the presence of progress bars and an explanation on why the progress bars are running.

- Logical flow of Information:
  In terms of relevant user information, it is delivered in a considered and useful manner. The calibration data is hidden away from the user when the service is running as it is hardly ever required. However the location and time data which is more important than the exact value of the pollution data is shown at above the pollution data. With that same thought in mind it can be seen that the although the exact value isn't needed an indication is needed. In this application the colour coded values blue for low, green for medium and red for high removed the need to read the exact values although these values are still provided. Continuing this notion of providing relevant information, the top most value is the most recent and this is emphasised by the large size of the text distinguishing it from the other values.

- Consistent Terminology:
  The terminology is also consistent as there is nothing that increases complexity. The terms unit id and settings are the only two words that the user interacts with. The user id has a matching term on the device itself, while settings follows its own definition.

3.3.4 User Feedback:

The greatest test for any application's user interface is to see how well it interacts with its users. Currently four people have used the application during field tests. Two of these people are external to the current Haze Watch project. The feedback was generally positive and there was no issue with how the application operated. Training the user to use the application was also an easy process as only one button is required to be pressed and again no complaints where received.

This application was designed on the basis of user feedback for earlier iterations of the application. This feedback consisted of increasing the number of progress bars and adding a way to completely stop the background process. These ideas from the users was implemented into the final design. No further improvements have been suggested at this time.

4 Mobile Application Development:

Developing the user interface is only one part of an mobile application. The majority of the work is conducted behind the scenes inside the application which provides the information for the user display as well as receiving data from teh device and translating it and then uploading it to the server. The following flow chart outlines how the application functions as well showing the major blocks of the application design process.

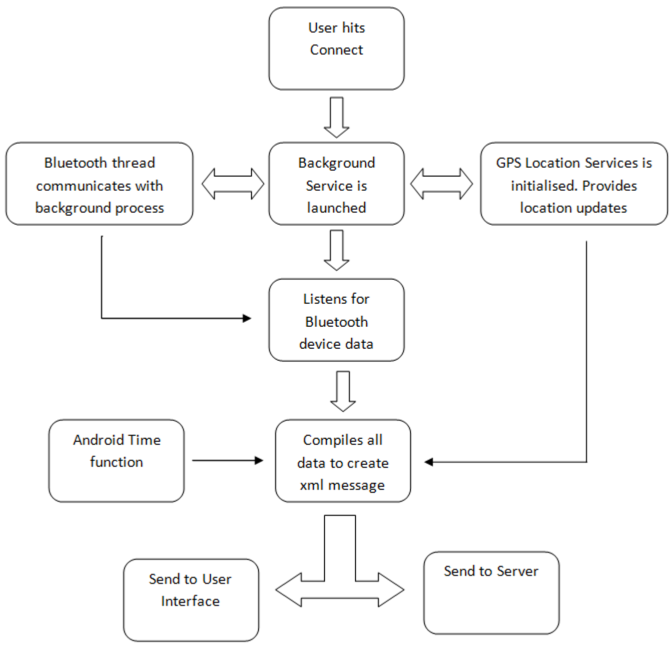


Figure 14: Flow chart of Mobile Application

In the image above, all of the two dimensional arrows indicate the creation of a new thread or process that runs asynchronously while the line arrows indicates information flow.

It can be seen from the chart there are effectively four areas of interest. These are the bluetooth interface, the translation and calibration of data, the location and time stamping of the data, and finally the upload to both the server and the user interface.

4.1 Bluetooth Interface:

The communication platform between the device and the mobile device is over Bluetooth. Bluetooth was chosen because it provides a one to one link between the device and the smart phone without interfering with the network connectivity, and also because it was the cheapest wireless communications platform that is widely available. As there is no established standard protocol for Bluetooth packet structure, one had to be designed for this project. This structure had to pass all the required data from the device to the application. The design constraints imposed upon the packet structure and the communication method were:

- There was to be only one way communication, from the device to the mobile application.
- The protocol needed to be extendable to handle more than one sensor in future implementations.
- Must have some method to identify the version number of the packet structure and the identification of the device.
- The packet structure must be simple and easy to implement in the micro-controller of the device.

4.1.1 Bluetooth Packet Structure:

The Bluetooth packet structure is shown below.

Byte	1	2	3	4	5			
Header:	Flag		Version #	Unit ID				
Message:						Gas Type	Data	

Figure 15: Bluetooth Packet Structure

This packet structure addresses all of the above criteria.

4.1.2 Error Checking:

The major issue with the first constraint that there should be only one way communication is that it limits the ability to implement common protocols like TCP over bluetooth which reduce errors. This meant implementing a method which dropped packets when errors are picked up. This occurs in this application by:

- Two byte start flag (Bytes 1 and 2) are always set to 0xFF and 0x00, and this combination will never occur in that order in the packet again. If it is picked up before end of the packet the packet should be dropped.
- Compare the unit id to the id that has been selected by the user, if there is a mist match drop the packet.

It should be noted that no cyclic check sum for error checking was implemented as one of the requirements was for simplicity in the bluetooth device and adding a cyclic checksum would not meet the requirements for simplicity.

4.1.3 Future Proofing:

The third byte contains the version id of the protocol. This is checked because there are two versions of the protocol, the older one does not contain this byte of information and therefore needs a separate function to handle it. By integrating this byte of data into the system it allows for the system to be future proofed and prevent further hard coding. The next two bytes make up the unit id which is unique for every device and allows up to 65,000 devices. This allows the protocol to be future proofed.

4.1.4 Message Handling:

The body of the message contains one byte to signal the type of gas and two bytes for the pollution value. This value is a 10 bit number and is divided such that the first 8 bits are found in the first byte and the last two are found in the second data byte. To add more gases to this tool all that has to be done is to repeat the body segment.

As the receiving platform was Android a quirk of the java platform has to be noted. In java, all bytes are signed and this is a significant issues when trying to deal with bytes of data and conversion to integers will see this sign to flow through to the integers. To avoid this the msb (the first byte) and lsb (the second byte) must be bitwise anded with 0xFF and then cast to an integer. This integer value is then for the msb bitshifted to the left by 2 and the lsb is bitshifted to the right by 6. These numbers are then summed for the final result of a ten bit number expressed as an integer (4 bytes). The process is shown below.

Algorithm 1 Recombining a ten bit number in Java

```
private int results(byte msb, byte lsb){        /*Results combines the the divided 8 and 2 bit number into a 10 bit number*/
    int a = 0;
    int b = 0;
    int result = 0;

    a = msb&0xFF;                               //&0xFF ensures that the int is not created
                                                //with leading 1s and retains the same value as the msb.

    a = a<<2;

    b = lsb&0xFF;
    b = b>>6;

    result = a+b;
    return result;
}
```

Beyond this the bluetooth connection is established on a separate thread using a handler for interrupts. This allows for a logical flow of the code and also prevents conflicts to access the data received by the mobile phone. A feature of modern smart phones is that most processes including bluetooth services are paused when the screen is turned off. To avoid this wakelock permissions to the CPU are provided to the application. This prevents the phone from pausing this service and allows for the application to continue operating while minimising the major cause of battery drain, the screen.

4.1.5 Testing and Implementation:

This protocol was implemented successfully, and the results of test can be seen in the figure below.

[][-1][0 101 -128][-64][-1][0 101 -128 64][-1][0 101 -128 64][-1 0][101 127 -128][-1 0][101 -128 64][-1 0][101 -128 0][-1][0
101 -128 0][-1][0 101 -128 -64][-1][0 101 -128 -64][-1 0][101 -128 -128][-1 0][101 -128 -64][-1 0][101 -128 -128][-1][0 101
-128 -64][-1][0 101 -128 64][-1][0 101 -128 -128][-1 0][101 -128 -64][-1 0][101 -128 64][-1 0][101 -128 -128][-1][0 101
-128 -128][-1][0 101 -128 -64][-1][0 101 -128 0][-1 0][101 -128 64][-1 0][101 -128 64][-1 0][101 -128 64][-1][0 101 -128
-128][-1][0 101 -128 -64][-1][0 101 -128 -128][-1 0][101 127 -64][-1 0][101 -128 -128][-1 0][101 -128 64][-1][0 101 -128
64][-1][0 101 -128 -128][-1][0 101 -128 -128][-1 0][101 -128 -128][-1 0][101 127 -64][-1][0 101 -128][-128][-1][0 101 -128
64][0][0 101 -128 64][-1 0][101 -128 0][-1 0][101 -128 64][-1 0][101 -128 64][-1][0 101 -128][64][-1][0 101 -128 -128][-1
][0 101 -128 -64][101 -128][101 -128 -64][-1 0][101 -128 -128][101 -128][101 -128 -128][-1][0 101 -128][-64][-1][0 101
-128 -64][-1][0 101 -128 -128][-1 0][101 -128 -128][-1 0][101 -128 64][-1 0][101 -128 -128][-1][0 101 -128 64][-1][0 101
-128 64][-1][0 101 -128 -128][-1 0][101 -128 -128][-1 0][101 -128 -64][-1 0][101 127 -64][-1][0 101 -128 -128][-1][0 101
-128 -128][-1][0 101 -128 -128][-1 0][101 -128 -64][-1 0][101 -128 -128][-1 0][101 -128 64][-1][0 101 127 -64][-1][0 101
-128 64][-1][0 101][-128 64][-1 0][101 -128 0][-1 0][101 -128 -128][-1 0][101 -128 -128][-1][0 101 -128 -128][-1][0 101
-128 64][-1][0 101 -128 -128][-1 0][101 -128 -128][-1 0][101 -128 64][-1 0][101 -128 -128][-1][0 101 -128 64][-1][0 101
-128 64][-1][0 101 127 -64][-1 0][101 -128 -128][-1 0][101 -128 64][-1][0 101 -128][64][-1][0 101 -128 64][-1][0 101 -128
-128][0][0 101 -128 64][-1 0][101 -128 0][-1 0][101 -128 -128][-1][0 101 -128][-128][-1][0 101 -128 64][-1][0 101 -128 -64
][-1 0][101 -128 64][-1 0][101 -128 -128][-1 0][101 127 -64][-1][0 101 -128][-64][-1][0 101 -128 64][-1][0 101 -128 -128
][-1 0][101 -128 -64][-1 0][101 -128 -64][-1 0][101 -128 -128][-1][0 101 -128][-128][-1][0 101 -128 -128][-1][0 101 -128 64
][-1 0][101 -128 64][-1 0][101 -128 -128][-1 0][101 -128 64][-1][0 101 -128 -128][-1][0 101 -128 -128][-1][0 101 -128 -128
][-1 0][101 -128 -128][-1 0][101 127 -64][-1 0][101 -128 -64][-1][0 101 -128 -128][-1][0 101 -128 -64][-1][0 101 -128 -64
][-1 0][101 -128 -128][-1 0][101 -128 -128][-1 0][101 -128 0][-1][0 101 -128 -128][-1][0 101 -128 -128][-1][0 101 -128 -128
][-1 0][101 -128 -128][-1 0][101 -128 64][-1 0][101 -128 -128][-1][0 101 127 -64][-1][0 101 -128 -128][-1][0 101 -128 64
][-1 0][101 -128 64][-1 0][101 -128 -128][-1 0][101 -128 -64][-1][0 101 -128 -128][-1][0 101 -128 0][-1][0 101 -128 64][-1 0
][101 -128 -64][-1 0][101 -128 -64][-1][0 101 -128][-64][-1][0 101 -128 -128][-1][0 101 -128 -128][-1 0][101 -127 0][-1 0

Figure 16: Print Out of Packets

In the above image the brackets are placed around the packets as they are received from the Bluetooth buffer, these packets are based on an older system than the one currently implemented. The above figure shows a numerical printout of the bytes. The negative values are a result of the bytes being signed in Java. This means that a byte with a value of 255 is equal to -1. Beyond this issue the collection of data shows that a bluetooth connection has been established and data is being received.

The issue is that the Bluetooth packets (shown by the square brackets) being received are not the same as the standard above. This is because the packet structure defined is layered on top of the bluetooth communications protocol packets and this will break up the Haze Watch packets to fit into the packets sent by the Bluetooth module. This means that the application must treat the incoming data as a stream rather than just checking that a packet came in, and it must recombine the received packets. When this is done it can be seen that packets resemble the standard established earlier.

The major issues picked up during this tests where that the Android 2.3.3 and below do not connect to the device because the bluetooth class id is set as a serial port device, hence Android version 3.0 and up had to be used. The second issue is that the device had connectivity problems that occurred in a random fashion. This is a quality issue and can be found in the analysis of device quality section of this report.

4.2 Timestamps and GeoTags:

Timestamps and GeoTags are required to make the pollution data collected by the device useful to server as it will allow it to identify and analyse the data with respect to location and time. The requirements for these two things are listed below.

- The timing and location services must be accurate
- These two services must operate at all times.
- The User must be provided with the location and time data.

The time stamping is a simple procedure as there are basic functions available as part of the java libraries and it does not require a network connection or any specialised hardware. On the other hand there are three methods for geotagging. These are using the inbuilt GPS, using the location services provided by the mobile network connection and finally using the wifi location services. The chosen option was the GPS as accuracy is critical for geotagging and a hard requirement. However there are shortcomings with this method chief of which is that when there is no signal there will be no location provided, for example in a tunnel. It was therefore necessary to develop an algorithm to handle this issue. As part of this thesis two systems were developed and are outlined in the two sub sections below.

4.2.1 Direction and Speed based location detection:

The first method that was implemented was direction and speed based positioning. This system used one minutes worth of past data to calculate the speed at which the vehicle was moving and also the direction it was going in and interpolated that information in a linear fashion when the GPS signal was lost. The result of this algorithm functioning in the Cook's River and M5 tunnel is below.

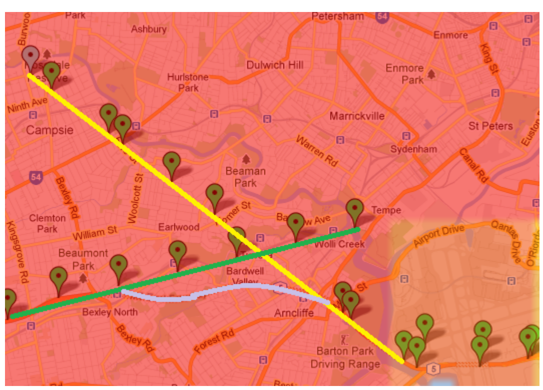


Figure 17: Direction and Speed Based positioning in the M5 tunnel

As seen in the above figure this method is not very accurate. The light blue line is the M5 tunnel while the yellow line is the approximation when approaching the tunnel from the city, while green is approaching tunnel from the south west. As it shows this algorithm fails to be accurate when there is a sharp deviation before the entrance or if there is a sudden change in the speed from outside the tunnel to inside the tunnel. The final issue with this strategy was that if the speed changed in the tunnel and the car spent more time and consequently more data points were gathered these points would continue to extend beyond the limits of the tunnel. These shortcomings made this strategy untenable as it failed to satisfy the requirements to be accurate.

4.2.2 Endpoint based linear interpolation:

The strategy that was implemented was the endpoint based linear interpolation. This strategy involved recording the start points and every value gathered in the tunnel until the endpoint of the tunnel is reached. At this point the difference between the start point and endpoints is taken and divided by the number of observations. This gives the average distance traveled between each observation and allows for a linear approximation of the tunnel which is limited by the tunnel endpoints and thereby provides a better approximation than the previous method.

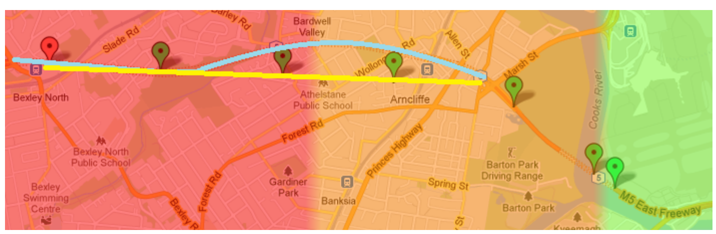


Figure 18: Endpoint based positioning in the M5 tunnel

The issues with this method again is that it is not perfectly accurate but for the purposes of this application it is accurate enough, secondly this modifies the notion that this application is a pass through between the device and the server, since initially the idea was for this application to be a dumb repeater. Finally the implementation of this system is more complex than the earlier system.

However as accuracy was more important for this component this algorithm was implemented.

4.2.3 Location Look Up:

Although the GPS co-ordinates provided by the above sections are understandable by machines and those who have experience with maps, it is by no stretch of the imagination easy for human use. Hence an address reverse look up task was implemented to show the user the address of the location from where the data was collected. This is a Google specific function which uses the Google maps api. This is seen in the earlier figure where the address is provided rather than the the gps co-ordinates. This also satisfies the third requirement as well as satisfying the earlier mention notion of intuitiveness.

4.2.4 Testing and Implementation:

The results of the implementation of the two strategies can be seen above for location detection. Testing the implementation of timestamps was a simple process as the date and time value can be retrieved as a string from the date function. This test was successful.

To test the GPS tunnel implementation without running field trials, external software called FakeGPS was used. This software as the name suggests provided false gps co-ordinates and by using these signals it was possible to test the implementations.

4.3 Calibration and Translation:

The pollution meter provides uncalibrated values and the server expects calibrated values, hence it falls to the mobile application to calibrate the values. The requirements for this section are set out below.

- The two byte result from the bluetooth module must be converted to a floating point value that measures the gas in parts per million.
- The data must be accurate.
- The calibration data may change and this should not require the application to be changed.

4.3.1 Calibration Equations

The pollution meter transmits a 10 bit integer that represents a quantised voltage value and before it can be used in any meaningful way into needs to be turned back into the voltage value (Vout) and then into the parts per million value through the calibration equations. These equations are shown below.

Algorithm 2 Calculating Vout

$$Vout = \left(\frac{x}{X} * Vref\right)$$

Where:
Vout is the Output Voltage.
Vref = 5V a constant.
x is the 10 bit number.
X = 1024 a constant that is the max value of x

In the calibration equations below there are two constants Sg and A. Sg indicates the sensitivity of the of the gas sensor while A is the offset. These values where found through experimentation by Judy and the table provided is the results of her work[14].

Algorithm 3 Calibration Equation

$$G(ppm) = \frac{1}{Sg}\left(\frac{Vout - Vref}{Rgain} - A\right)$$

Where:
Sg is the sensitivity of the gas sensor.
Vout is the Output Voltage.
Vref = 5V a constant.
Rgain = 100K.
A is a standard offset.

In the table below, the Sg and A values have been given, however it is expected that these values will change over time and as such the calibration numbers will change and these should be used as a guide only. In the application these values are not hard coded rather they are downloaded from the server when the application is started this way the no change to the application is needed when the calibration values change.

Sensor #	Sensitivity (Sg)	Offset (A)
CO 1	73.99	-99.29
CO 2	68.33	-69.541
CO 3	75.61	-79.126
CO 4	72.24	-88.667
CO 5	65.486	-94.654
CO 6	79.641	-85.714
CO 7	72.333	-105.94
NO2 1	290.58	-6.5014
NO2 2	335.39	48.77
NO2 3	459.12	21.915
NO2 4	374.94	60.384
NO2 5	608.99	204.55
NO2 6	432.63	46.98
NO2 7	567.25	206.98

Table 1: Calibration Constants

4.3.2 Implementation:

The implementation of the calibration and translation functions have successfully implemented in the application and they meet the above criteria.

- Conversion of Bytes:
  As mentioned in the bluetooth the bytes have been converted into a 10 bit integer. This 10 bit integer is a value between 0 and 1024. Due to the design of the bluetooth module the integer actually falls in a range between 512-1024, and for CO 512 is equal to 0 ppm, while for NO2 1024 is equal to 0. This means that the function had to be flexible to deal with both cases. The above calibration equations where implemented and an issue that arose was that the calibration constants need to be multiplied by 10E-9.

- Accuracy:
  The electrochemical sensors are analogue and continually vary its current depending on the level of pollution and the microcontroller samples the sensor at 10 Hz and transmits the data over bluetooth at the same rate. The issue is that this may have momentary spikes at a point in time which may be an inaccurate reading. To resolve this and improve accuracy, the values are averaged over a ten second period which consists of a 100 samples. This smooths out any momentary changes.

- Application Flexibility:
  The calibration constants are stored on a server and at application start these values are downloaded from the server. Therefore the values can change in the server and these changes will flow to every single application ever deployed. This allows for flexibility.

4.4 Upload to Server:

The final thing the application must do is to upload the information collected to the server. The requirements for this section are set out below.

- Uploads must occur regularly.

- Uploads must be in the standard XML format so as to be accepted by the server.

- Method to handle network loss.

4.4.1 Regular Uploads:

The application uploads the data every 10 second through an asynchronous task. This is because the upload process can take a long time depending on the type of mobile network available, as it requires a http connection to be established. Furthermore by doing it asynchronously it avoids other processes from being interrupted such as the aggregation of pollutant values. The algorithm used to send data is rather simple and shown below.

Algorithm 4 Asynchronous Sending Algorithm

```
private class sendfiles extends AsyncTask<String, Void, String> {

    protected String doInBackground(String... urls) {              /*Send Files Function*/

        String result = null;
        String res;
        HttpClient client = new DefaultHttpClient();                    //Initialise APACHE http
        HttpPost request = new HttpPost(URL);
        HttpResponse response;

        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2); //Set Value name and data to send
        nameValuePairs.add(new BasicNameValuePair("content", xmlstring));

        try {
            request.setEntity(new UrlEncodedFormEntity(nameValuePairs));      //try sending
            response = client.execute(request);
            Log.wtf("Sent Message",xmlstring);
            BufferedReader br = new BufferedReader(new InputStreamReader(
                    response.getEntity().getContent()));
            while((res = br.readLine() )!= null){

                Log.wtf("msg", res);
            }

        } catch (ClientProtocolException e) {                              //Catch failure

            e.printStackTrace();
        } catch (IOException e) {

            e.printStackTrace();
        }

        if(netsave){                                                       //Send saved data from network loss
            read_send(SVDT, 1);
            netsave = false;
            setFile = true;
        }

        if(tunnel_save){                                                   //Send data from GPS loss
            tunnel_save = false;
            read_send(NGDT, 2);
        }

        return result;
    }

    protected void onProgressUpdate() {
    }
```

4.4.2 XML format:

The second requirement is that the data must be transferred in XML format. This is an established standard for data transfer over networks and was the only way to provide the server with data. This was established with the XML library that is available in the standard Android libraries. The output of the XML function is a string that looks like the figure below. The XML contains all of the relevant data including the device id (group id), the pollution data, the location and timestamp, and the gas type.

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <info></info>
  <groupid>108</groupid>
  <samples>
    <sample>
      <datetime>2010-05-24 18:00:00</datetime>
      <latitude>-33.7972222222222</latitude>
      <longitude>150.765833333333</longitude>
      <no2>2.0</no2>
    </sample>
  </samples>
</data>
</xml>
```

Figure 19: Endpoint based positioning in the M5 tunnel

4.4.3 Network Loss:

The final requirement is to handle network loss. This is an essential requirement because the network can drop out at various locations due to coverage issues. This problem is extremely significant in rural locations or in areas of low coverage such as indoors or with certain telecoms. This is a significant issue as seen when over 20 minutes worth of data was lost when the mobile network dropped out.

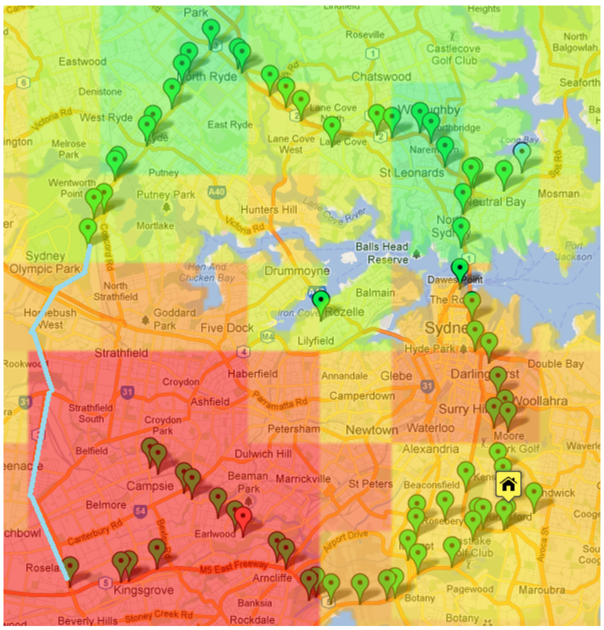


Figure 20: Missing Data and first tunnel algorithm

The strategy used is to check if the network is available before sending the data and to send it if it is available. If it is not available the data is saved to an internal file and sent when the network connection returns. This system has performed very well and there has been no issues once the system has been implemented. The successful system implementation is shown below.

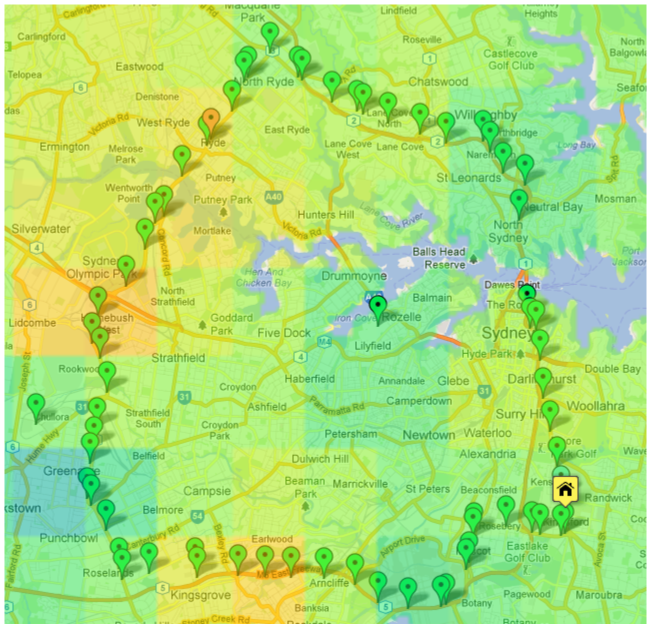
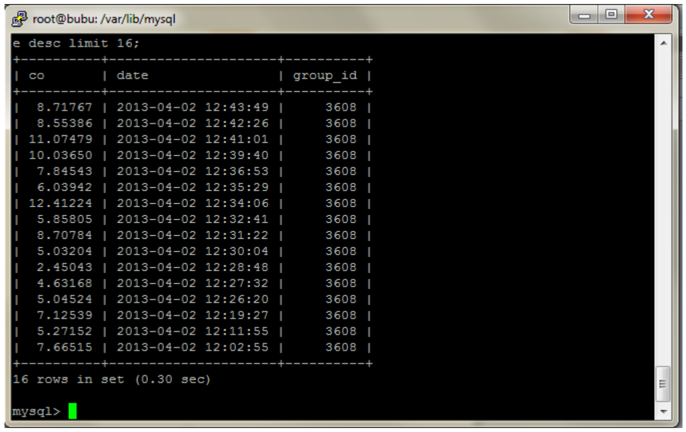


Figure 21: Missing Data and first tunnel algorithm

4.4.4 Testing:

The application has successfully uploaded data to the server. This could be tested by remotely logging into the server and looking at the database. By doing this the exact data uploaded could be seen and any missing data sets could be identified. After repeated testing it was seen the application functioned well even when tested for more than an hour or so. An example of the test is seen in the figure below.



```
root@bubu: /var/lib/mysql
e desc limit 16;
+----------+---------------------+----------+
| co       | date                | group_id |
+----------+---------------------+----------+
|  8.71767 | 2013-04-02 12:43:49 |     3608 |
|  8.55386 | 2013-04-02 12:42:26 |     3608 |
| 11.07479 | 2013-04-02 12:41:01 |     3608 |
| 10.03650 | 2013-04-02 12:39:40 |     3608 |
|  7.84543 | 2013-04-02 12:36:53 |     3608 |
|  6.03942 | 2013-04-02 12:35:29 |     3608 |
| 12.41224 | 2013-04-02 12:34:06 |     3608 |
|  5.85805 | 2013-04-02 12:32:41 |     3608 |
|  8.70784 | 2013-04-02 12:31:22 |     3608 |
|  5.03204 | 2013-04-02 12:30:04 |     3608 |
|  2.45043 | 2013-04-02 12:28:48 |     3608 |
|  4.63168 | 2013-04-02 12:27:32 |     3608 |
|  5.04524 | 2013-04-02 12:26:20 |     3608 |
|  7.12539 | 2013-04-02 12:19:27 |     3608 |
|  5.27152 | 2013-04-02 12:11:55 |     3608 |
|  7.66515 | 2013-04-02 12:02:55 |     3608 |
+----------+---------------------+----------+
16 rows in set (0.30 sec)

mysql>
```

Figure 22: Server Uploads

5 Field Trials:

Although each component of the system has been demonstrated to work, the system needs to be tested to see if it meets the objectives of the haze watch project. As mentioned in the background the objectives of the Haze Watch project was to crowd source pollution data, by providing cheap devices to the general public. However before data can be collected from the general public, tests have to be run to establish if the system can suitably show that it meets a set standard in the following criteria.

- Accuracy
  Accuracy is measured against a commercial meter which is guaranteed to be accurate in relations to concentrations above 5ppm.

- Robustness
  The robustness of the entire system needs to be seen and whether it can handle different weather conditions as well as other factors.

- Usefulness
  The third criteria is to see if the collected data is useful, otherwise users will have no incentive to use this product.

5.1 Trial 1: Bunnings Run:

The first full test for the system was a trial run from Holsworthy to Bunnings in Bankstown on the 2nd of April 2013. It was a test to see if the system was working. The result of the test if the basic location services were working and to see how the device and the application functioned during use. The route and a visual representation of the results is shown below.

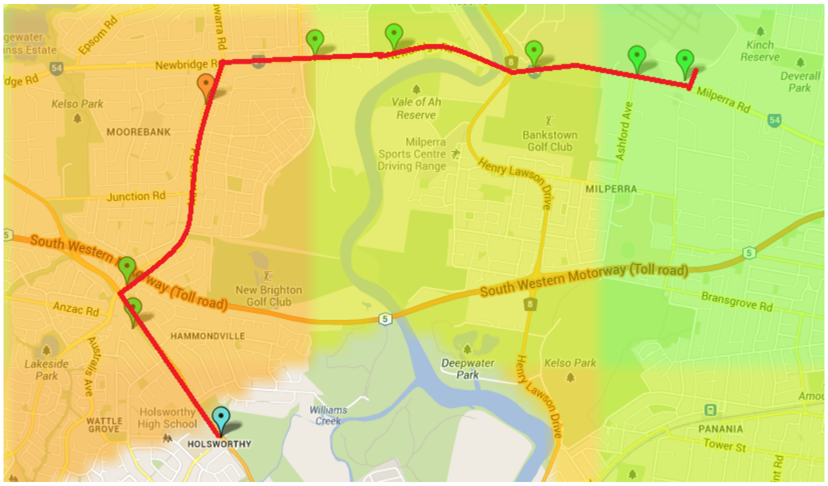


Figure 23: Trial 1: Holsworthy to Bunnings

5.1.1 Observations:

The observations drawn from this test are below:

- The maximum time between readings should be around 10 seconds. This is due to the fact that the vehicle covers significant distance in that time and not enough readings can be taken for the system to meet the requirements for accuracy.
- The positioning and timestamps for non-tunnel situations work correctly and data was uploaded to the server.
- There is a period where data has not being gathered, this is due to the device pausing the background service.
- The data values are unusually high.

5.1.2 Analysis:

- Accuracy:
  The values are unusually high because the device had not been primed for 8 hours before the field trial, hence it does not effect accuracy. However the location and time values are correct and the values being received by the server are the same values that are being sent by the phone indicating that the system works as expected.
- Robustness:
  The system failed the robustness test because it stopped gathering data during a period of time when the phone screen was off. The reason for this is the Android operating system pauses applications unless they have

permission to run. This CPU wakelock permission was provided and the system functions as expected.

- Usefulness:
  This criteria was not being tested for, however since the device was not primed for the required 8 hours the pollution data cannot be used.

5.2 Trial 2: Liverpool Run:

The second major trial was undertaken after the device was primed for 8 hours, on the 2nd of May 2012. It ran from Holsworthy to Westfields at Liverpool. The aim of this test was to confirm the previous test analysis regarding maximum time limits as well as location data and also if the values could be useful. The figure below shows the route and CO visualisation. The route is in black, and the return route is in red where it deviates from the route. This test occurred to address the issues in the former test as well as a trial run to the first official test.

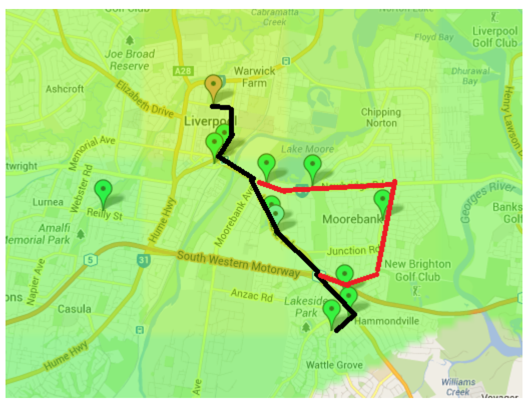


Figure 24: Trial 2: Holsworthy to Liverpool Westfields

5.2.1 Observations:

- Data seems to be missing over significant sections of the route, with map only showing 15 points. However more than 15 values available on server, as seen in figure below. This suggests an issue with the server from this test.

572923	NULL	2013-05-02 10:12:40	3608	3.09075	-33.948085	150.947868
572922	NULL	2013-05-02 10:12:29	3608	2.82656	-33.948000	150.947886
572921	NULL	2013-05-02 10:12:19	3608	2.99836	-33.948127	150.947791
572920	NULL	2013-05-02 10:12:08	3608	2.94556	-33.947499	150.947351
572919	NULL	2013-05-02 10:11:57	3608	2.78036	-33.947954	150.946798
572918	NULL	2013-05-02 10:11:47	3608	2.43082	-33.949038	150.946666
572917	NULL	2013-05-02 10:11:36	3608	3.59889	-33.949824	150.946588
572916	NULL	2013-05-02 10:11:26	3608	3.44051	-33.949976	150.944999
572915	NULL	2013-05-02 10:11:16	3608	1.75747	-33.949124	150.944863
572914	NULL	2013-05-02 10:11:05	3608	1.13010	-33.948461	150.944000
572913	NULL	2013-05-02 10:10:55	3608	1.38153	-33.947820	150.945357
572912	NULL	2013-05-02 10:10:45	3608	1.77749	-33.946966	150.945967
572911	NULL	2013-05-02 10:10:35	3608	2.77398	-33.946027	150.945990
572910	NULL	2013-05-02 10:10:24	3608	2.32523	-33.945129	150.946940
572909	NULL	2013-05-02 10:10:13	3608	2.62220	-33.945126	150.946938
572908	NULL	2013-05-02 10:10:03	3608	3.47351	-33.945210	150.946901
572907	NULL	2013-05-02 10:09:53	3608	3.67809	-33.944868	150.947159
572906	NULL	2013-05-02 10:09:43	3608	3.25573	-33.943649	150.948265
572905	NULL	2013-05-02 10:09:32	3608	3.36132	-33.943031	150.949950
572904	NULL	2013-05-02 10:09:21	3608	4.02785	-33.942545	150.951819
572903	NULL	2013-05-02 10:09:10	3608	4.52939	-33.941179	150.952479
572902	NULL	2013-05-02 10:09:00	3608	3.51310	-33.939996	150.952774
572901	NULL	2013-05-02 10:08:49	3608	3.66489	-33.938640	150.953004
572900	NULL	2013-05-02 10:08:39	3608	3.41411	-33.937086	150.952894
572899	NULL	2013-05-02 10:08:28	3608	4.27862	-33.935725	150.953248
572898	NULL	2013-05-02 10:08:18	3608	4.99134	-33.934177	150.953725
572897	NULL	2013-05-02 10:08:07	3608	3.38772	-33.933770	150.953253
572896	NULL	2013-05-02 10:07:57	3608	4.06084	-33.933766	150.953247
572895	NULL	2013-05-02 10:07:46	3608	4.39741	-33.933769	150.953251
572894	NULL	2013-05-02 10:07:36	3608	4.56239	-33.933771	150.953254
572893	NULL	2013-05-02 10:07:26	3608	4.89895	-33.933763	150.953257
572892	NULL	2013-05-02 10:07:16	3608	4.37761	-33.933759	150.953249
572891	NULL	2013-05-02 10:07:05	3608	4.02125	-33.933761	150.953242
572890	NULL	2013-05-02 10:06:55	3608	4.21263	-33.933769	150.953252
572889	NULL	2013-05-02 10:06:45	3608	4.65478	-33.933772	150.953243
572888	NULL	2013-05-02 10:06:34	3608	4.72077	-33.933754	150.953410
572887	NULL	2013-05-02 10:06:23	3608	3.86947	-33.933777	150.952145
572886	NULL	2013-05-02 10:06:13	3608	4.03445	-33.933804	150.951087
572885	NULL	2013-05-02 10:06:02	3608	3.41411	-33.933739	150.949729
572884	NULL	2013-05-02 10:05:52	3608	3.62507	-33.933878	150.948632
572883	NULL	2013-05-02 10:05:41	3608	7.26810	-33.933645	150.948367
572882	NULL	2013-05-02 10:05:30	3608	5.27511	-33.932997	150.948595
572881	NULL	2013-05-02 10:05:20	3608	9.22809	-33.932068	150.948715
572880	NULL	2013-05-02 10:05:10	3608	7.22190	-33.931504	150.948585
572879	NULL	2013-05-02 10:05:00	3608	3.49331	-33.931119	150.948280
572878	NULL	2013-05-02 10:04:49	3608	5.14973	-33.930135	150.948206
572877	NULL	2013-05-02 10:04:38	3608	6.09342	-33.929293	150.948091
572876	NULL	2013-05-02 10:04:28	3608	7.55847	-33.929170	150.947919
572875	NULL	2013-05-02 10:04:17	3608	11.77541	-33.929169	150.947918
572874	NULL	2013-05-02 10:04:06	3608	7.89503	-33.929182	150.947906
572873	NULL	2013-05-02 10:03:56	3608	3.98825	-33.929163	150.948038
572872	NULL	2013-05-02 10:03:46	3608	3.71108	-33.929127	150.947490
572871	NULL	2013-05-02 10:03:35	3608	3.84307	-33.929183	150.945586
572870	NULL	2013-05-02 10:03:25	3608	3.94206	-33.929293	150.941660
572869	NULL	2013-05-02 10:03:14	3608	5.26191	-33.929823	150.939932
572868	NULL	2013-05-02 10:03:04	3608	7.12291	-33.930339	150.937907
572867	NULL	2013-05-02 10:02:53	3608	4.68778	-33.929913	150.935853
572866	NULL	2013-05-02 10:02:43	3608	5.02434	-33.929143	150.933861
572865	NULL	2013-05-02 10:02:32	3608	6.00763	-33.928569	150.932295
572864	NULL	2013-05-02 10:02:22	3608	6.03403	-33.928040	150.930831
572863	NULL	2013-05-02 10:02:12	3608	5.94164	-33.928034	150.930829
572862	NULL	2013-05-02 10:02:01	3608	5.84265	-33.928114	150.931037
572861	NULL	2013-05-02 10:01:50	3608	6.05383	-33.927550	150.929561

Figure 25: Trial 2: Holsworthy to Liverpool Westfields

- Location and timing work correctly.
- No location services in Westfield's parking lot.

5.2.2 Analysis:

- Accuracy:
  The only values missing are on the map and not in the server so the only issue is with the visualisation tools, more tests need to be run to see why this issue occurred. Other than that the 10 second limit is providing accurate readings. Higher readings are seen in enclosed spaces. The major

issue with accuracy is to do with the lack of accuracy in the location services in the enclosed spaces. Therefore in this case the system has failed in the accuracy tests.

- Robustness
  The system passed the robustness test in this situation because it did not fail to upload data when the screen was off. The only issue was with the pollution visualisation tool which did not function as expected. As a peripheral tool of the Haze Watch project it does not indicate a failure of the system. This is further enhanced by the fact that the server has received all of the uploaded data.

- Usefulness
  The data as the system has been primed can be useful to users, except in tunnels. As this criteria was not explicitly tested for no opinion will be given for this test on whether the objective was achieved.

5.3 Trial 3: UNSW to Port Botany circle:

The third field test was the first official test as conducted by the current Haze Watch team. It was conducted on the 3rd of May 2013. The method of the test was to have the subject carry a yellow monitor and drive on the red route while a second person the monitor holder would drive on a second route. The aim of this test was to demonstrate that pollution values would be similar across similar roads.

The two routes and the pollution visualisation is shown below.

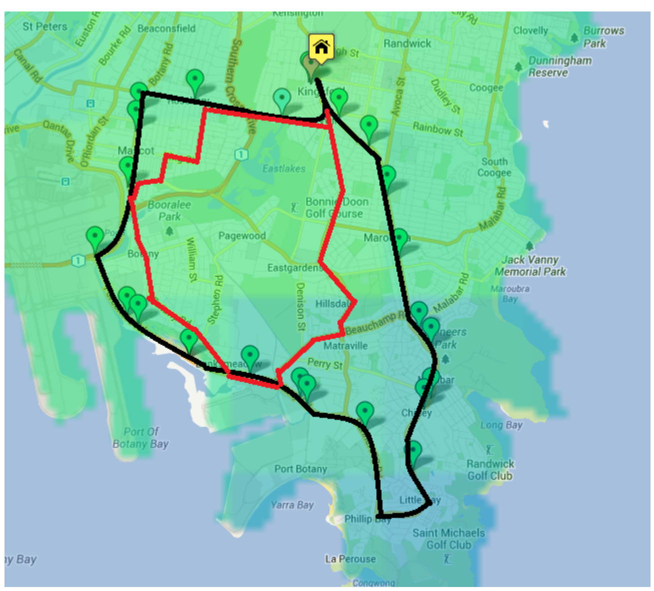


Figure 26: Trial 3: Circle from UNSW to Port Botany

5.3.1 Observations:

The results to this test was graphed as the device had been primed for the eight hour period. The graph is shown below.

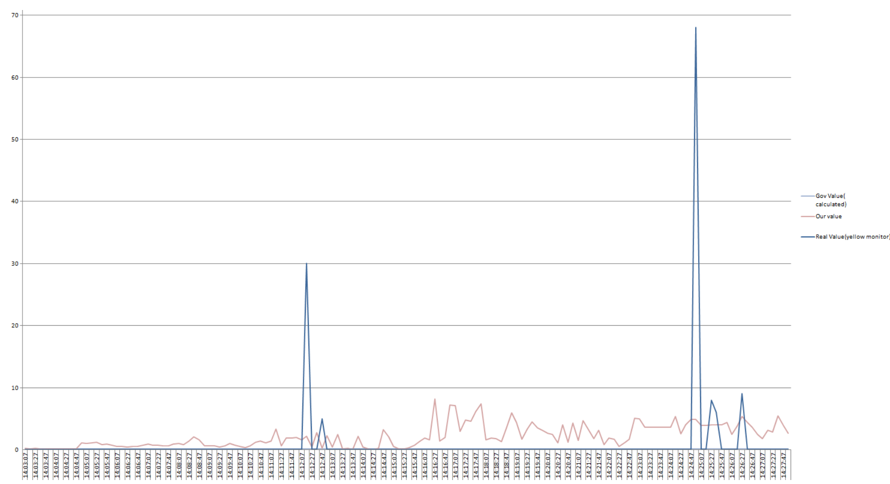


Figure 27: Trial 3: Graph of UNSW to Port Botany circle route

These observations drawn from these results include:

- Low Pollution values around the Port Botany area, with notable exceptions at the Airport and at the cross street between Foreshore Road and General Homes Drive.
- Two sharp spikes indicating that the subject was exposed to a heavily polluting vehicle.
- No correlation between the two data sets.

5.3.2 Analysis:

- Accuracy:
  The pollution values recorded at the points are most likely correct and the location data has functioned perfectly. Hence accuracy is satisfied.
- Robustness:
  No data was lost and there was no issue with the system or the peripheral services which meant the issue in field test 2 could not be recreated.
- Usefulness:
  The data suggests that the original aim of predicting pollution on a close by road by knowing the value of a road is incorrect. It suggests that pollution on roads is highly localised. This is backed up the two spikes

that occurred while following polluting vehicles. However it could also mean that the device isn't calibrated properly, this will be checked in the next field test. This means that the data is useless for this purpose hence the usefulness test has not been successful.

5.4 Trial 4: Sydney Loop:

The fourth field test was a loop around Sydney where both the commercial meter and the bluetooth device was on the same car. This test was conducted on the 17th of May 2013. The aim of this test was to show that the Haze Watch device was accurate and matched the commercial meter. It was also to test the tunnel location strategy. The result for the pollution visualisation is below.

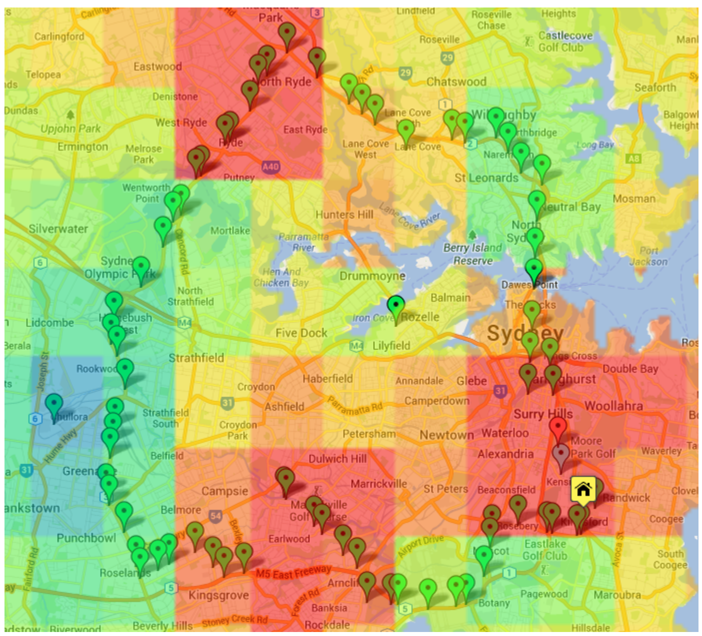


Figure 28: Trial 4: Pollution Visualisation of the Greater Sydney drive

The pollution visualisation doesn't show the values of the commercial meter hence the graph below was generated to show the values of both the commercial meter and the values from the server.

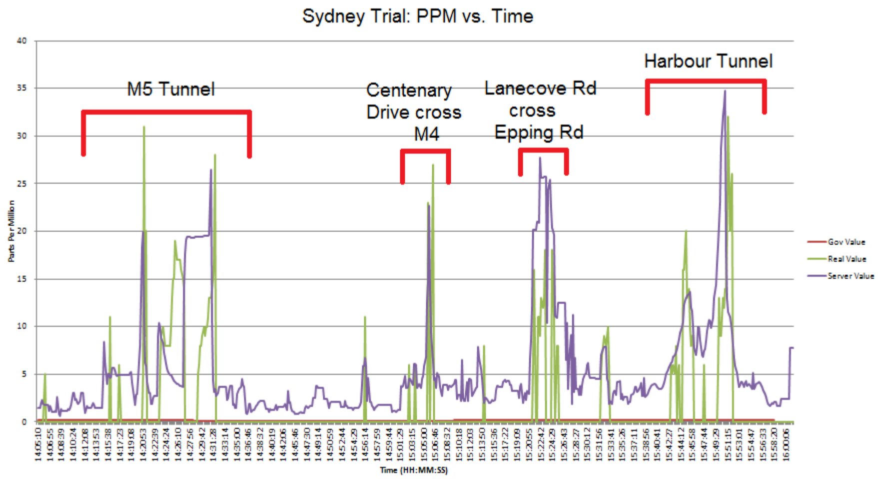


Figure 29: Trial 4: Graph of the Sydney drive

5.4.1 Observations:

The observations from the above two figures indicate:

- The location system in tunnels is very inaccurate. The only reason issues where avoided was due to the size of the route.
- The system has worked in all situations, even in places where previously there was a network failure.
- There is a very good correlation between the commercial meter and the bluetooth device

5.4.2 Analysis:

- Accuracy:
  The tunnel algorithm was inaccurate and therefore the second strategy has to be implemented. However since both the device and the commercial meter at the same location at the same time location is not important to the test and teh data needs only be mapped by time alone. The data correlation indicates that the meter is working correctly. Furthermore form the data it can be seen that pollution is not only high inside tunnels but also on congested roads suggesting that it may be possible to predict pollution based on congestion. However it may also indicate that pollution could be predicted from temporal displacement.

- Robustness:
  This criteria has been completely satisfied, as the system is completely working.

- Usefulness:
  The data is suggested to be highly accurate and it is useful. However more tests need to be done if it is to be seen if it can be used for prediction.

5.5 Trial 5: Temporal Displacement around Sydney:

The fifth and final trial conducted during my time with the Haze Watch project was to see if a temporal prediction could be made. This involved driving with the bluetooth meter around the same route an hour before the subject carrying the commercial meter and an hour after the subject. By doing this it was hoped that there would be possible to show a relationship between CO and time. This test was carried out on the 26th of May 2013 and involved all of the members of the Haze Watch team. The two runs have their pollution visualisation maps and the corresponding graphs below. In the third graph the purple line is the prediction and the green line is the commercial meter value.

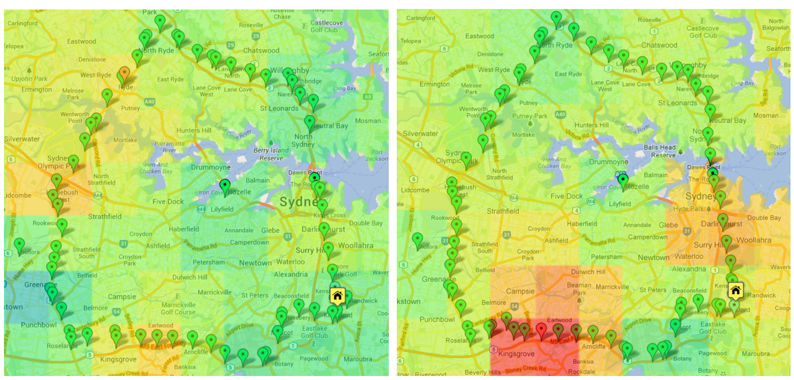


Figure 30: Trial 5: Temporal Trial (a. 1st Round, b. 3rd Round)

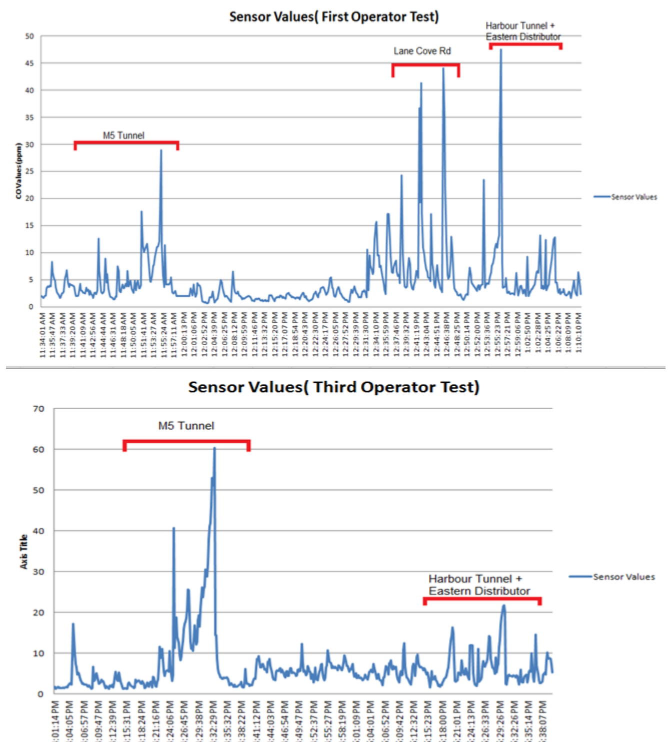


Figure 31: Trial 5: Temporal Trial Graphs (a. 1st Round, b. 3rd Round)

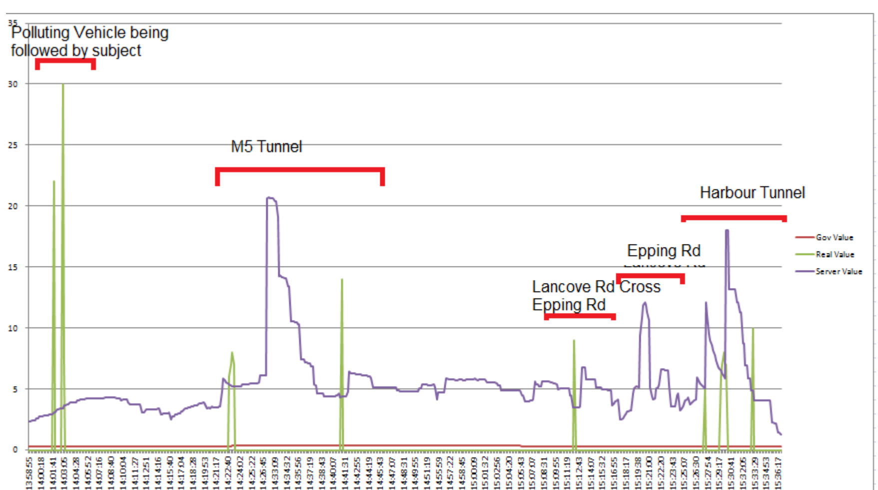


Figure 32: Trial 5: Temporal Graphical Prediction and subject run

5.5.1 Observations:

The observations that can be gleaned from all of the above maps and graphs are:

- All of the points are very accurate, and the tunnel location problem has been resolved
- The pollution values change significantly at different times.
- There is little correlation between the subject data and the trial runs as evidenced by the third graph, other than in tunnels.
- Traffic congestion was different at different times, the start of the subject run was similar to the 1st run conditions, while the latter half was similar to the second run.

5.5.2 Analysis:

- Accuracy:
  As seen from the third graph temporal prediction is not very accurate. This suggests that pollution is not a function of time. However the correlation noticed in trial four with regard to congestion occurs here as well. For instance in the first run the M5 tunnel pollution values are much closer to the subject run. These times also had similar congestion levels as well, similarly the Epping road and the harbour tunnel congestion was similar over the second run and the subject run. The pollution levels are correspondingly similar. Hence it suggests that traffic congestion may be the major cause of pollution.
- Robustness:
  This system ran perfectly the tunnel estimates are accurate and there was no failures in uploads.
- Usability:
  The system is usable for people who carry devices at this point in time.

5.6 Field Trials Conclusion:

From the field trials conducted thus far, it can be concluded that pollution can be predicted based on time on a very inaccurate basis. However the thought has arisen that it may not be the time that is most important but the level of congestion on the road. With this factor in mind it may be possible to predict pollution levels at similar congestion levels more accurately. This hypothesis needs to be tested in further field trials, to establish if this is the case although all indications point to it being so.

The system otherwise has shown to perform in real world conditions and is currently a useful tool to measure the current levels of pollution at a location. As such this system has met the requirements that were originally set out for success of being accurate, robust and useful.

6 Quality Analysis:

It has been shown that the system as a whole is successful. However a system wide analysis does not reveal issues with the individual components of the said system as other components of the system can cover up the weakness of a component. Hence the quality of each component of the system must be looked at to find where it weaknesses are. By doing this we will be able to see when something is wrong and where something will go wrong before it does. This will errors and the negative externalities that arise when things go wrong.

The components that will be looked at in this system is are bluetooth device, the mobile application, and the data collected. By analysing this three key components we will be able to see how things can be improved as the Haze Watch project progresses.

6.1 Device quality Analysis:

The device has three major areas where quality analysis needs to be conducted in. These are as follows:

- The Battery Life:
  Battery Life is a major concern as this device is being handed to the public and it will be of great convenience if they have to constantly change the battery.
- Device Priming:
  The device priming is to test when the device sensors settle down to an average value.
- Bluetooth Connectivity:
  The bluetooth connection is critical to the application and it will be analysed in detail.

6.1.1 Battery Life:

The battery life of the device is one of the most important measures possible. It will give us concrete time limits for how long the device can be used for and allow future planning for device use. To test this devices where given fresh batteries and connected to android devices to measure pollution. At intervals the battery voltages and time where recorded. This can be seen in the figure below.

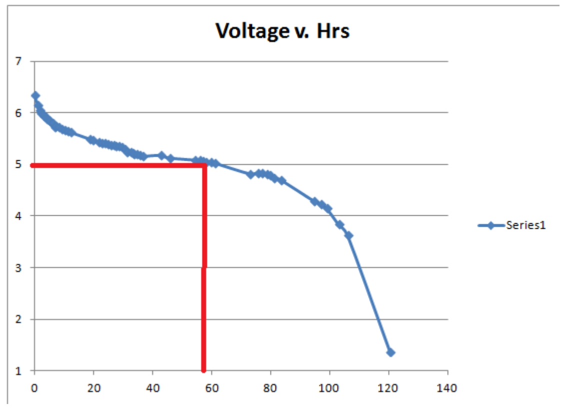


Figure 33: Battery Voltage vs. Time

In the figure a red line is drawn at the 5V mark which corresponds to about 56 hours. This is regarded as the battery life because after this point the voltage received by the electrochemical sensor will fall in line with the changing battery voltage this will cause the value outputted by the sensor to change as the reference voltage is now no longer constant. As such there will be an error in the readings after the 56 hour mark. This is a fault with the device that needs to be addressed.

6.1.2 Device Priming:

The electrochemical sensor takes time to warm up and it varies by how long the device was inactive for. The graph below demonstrates this.

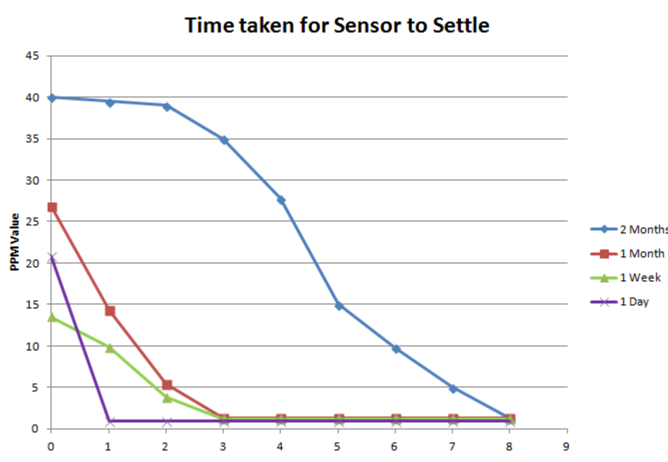


Figure 34: Battery Voltage vs. Time

From the graph it can be seen that leaving the device inactive for a month makes the sensor lose its sensitivity and ultimately has a worse case settling time of 8 hours. Continued use means that the time taken for the sensor to settle actually falls. This suggests that the device should continually be used otherwise it will take a long time until it becomes usable.

To resolve this issue the designer of the circuit, Matthew Kelly had left the analogue circuit to be always on if the batteries where connected. This would have been fine however there was an issue in which the digital circuit would draw current and keep itself on. This resulted in unusual behaviour by the bluetooth module and battery life was severely impacted as seen above.

6.1.3 Bluetooth Connectivity:

Bluetooth connectivity is critical for the device to function correctly as its required to interface to the smart phone application to the server. However the issue is that the device does not occasionally connect to the smart phone. Why this issue arise is unknown, however the hypothesis is that the bluetooth module has not been connected properly. Advice from a former Haze Watch member has been provided who suggested to reconnect the connections as per his drawings. This must be done in the next revision of the device.

Until that time, analysis and experimentation has shown that the following methods will help resolve the issue.

- Turn the device on and jiggle the battery for a temporary disconnect.
- Turn the Android bluetooth connection on and off
- Wait for a few seconds or until the red LED turns off

These methods are all empirical in nature. Appendix A contains a copy of the email sent by James Carrapetta which outlines the probable issue.

6.2 Application Analysis:

The application which was developed as a part of this thesis needs to also be analysed because it is a critical component of the system. The Application as mentioned earlier has a few key areas that need to be explored, these are as below.

- The Battery life of the smart phone
  The battery life of the smart phone is very important as the user will most likely use the smart phone for other purposes than just as an interconnect between the device and the server.
- Time between Location and Value Updates:
  It is important to know the time difference between location and value updates because this affects accuracy, a large difference means that the location at which the reading is taken is completely different to where it really occurred.

- Accuracy of location:
  The final stage looks at the accuracy of location, mostly when the GPS signal is lost or not available.

6.2.1 Battery Life:

The battery life of a smart phone while running the Haze Watch application varies with the type of the type of smartphone used. The two types tested where Google's Nexus 4 and Samsung's Galaxy S3. The S3 has a larger battery and a longer battery life than the Nexus 4, when not running any application however both contain 2100 mAH batteries.

The test of the Nexus 4 battery life was part of the fourth field trial highlighted in this document. To get accurate battery statistics an application called better battery stats was used. This when combined with the onboard usage analyser helped show the values. The graph below shows how both phones performed while running this application. This test did not consider the user using any other application while the Haze Watch process was running.

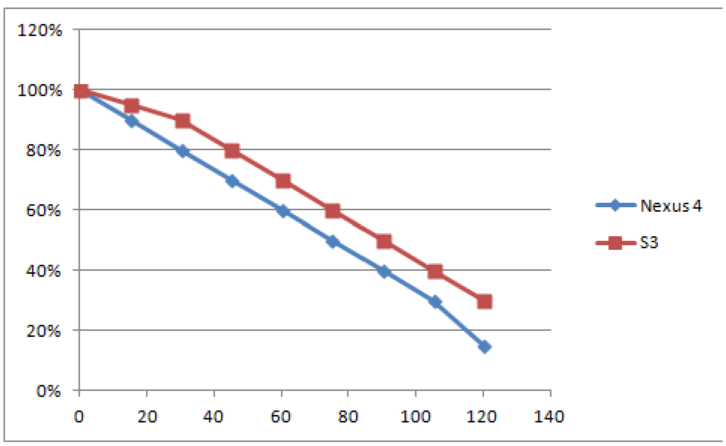


Figure 35: Smart phone battery life

The bottom axis is time while the left axis is percentage battery life. In this test it can be seen that the Nexus 4 falls to 10% in 2 hours while the S3 falls to 15% in the same period of time. This is unfortunately very battery heavy. The major drain if the screen is not on the usage of the GPS and mobile network connection.

An option to minimise this would be by implementing timed sending whereby data is saved and sent at only one point in time. This would effectively reduce the network connectivity required for the application to run. The other is to implement another location strategy, and will be discussed in following sections.

As the device is mounted on a car and the smart phone would be in the car, the best way to resolve the 2hr battery life issue is to plug the phone into the

12V DC output of the car through a mobile phone charger. This will prevent the phone running out of battery and keeps it charged while receiving information. This solution was implemented by the Haze Watch team when running the other field trials.

6.2.2 Time between Location and Value Updates:

This is a measure of how long it between the pollution value updating and the location value updating. This is significant because as mentioned earlier the difference if significant means that the pollution values are recorded in the wrong areas and one of the objectives of the Haze Watch project is to provide accurate data. What was found however was that that the initial co-ordinate discovery time is always significantly different to every other case. As such two graphs have been used to show this factor.

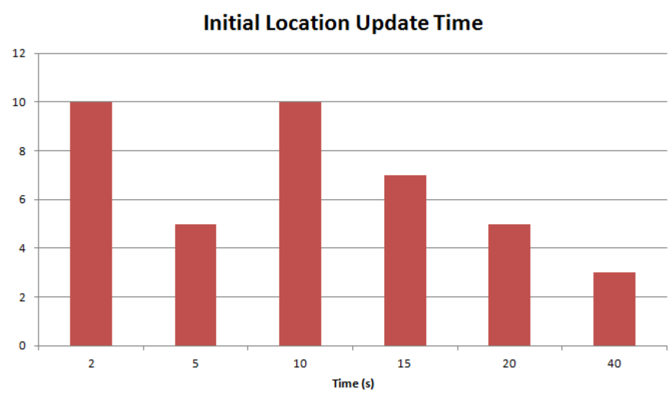


Figure 36: Initial Update Time

The initial update time is concentrated around the 10-15 second mark, this is expected as the GPS listener update every 10 seconds and a quirk of the platform seems to be that when set it waits an entire cycle before updating. This is avoided when another application has already set the GPS listener and this is reflected in the high number of 2 second initial calls. The 20s -40s values suggest that the GPS has not been set and the application is waiting for the GPS to set itself. This is most likely due to a lack of access to the GPS satellite signal.

The non-initial case is shown below. It is much better behaved than the initial case because there is no possibility of the GPS being a null value.

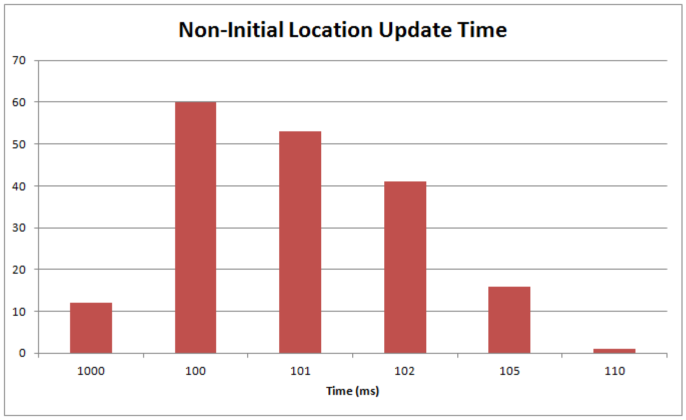


Figure 37: Non-Initial Update Time

The non-initial update time is mainly around 100mS. This means that for a car travelling at 100km/h the worst case offset between the pollution value and the location data as a distance is only about 2m. It is highly unlikely that pollution values will change over such a short distance. The graph also shows a significant reading for the 1second mark, with the GPS taking a second to update on 10 updates. This is because of the tunnel or locations where GPS was not available. The code inherently waits for a second for the GPS to update before it moves on.

As such the worst case scenario is either 1 second or 0.1 seconds depending on the scenario.

6.2.3 Accuracy of Location:

The final thing to be looked at is how accurate the location is. With GPS being enabled and with working satellites it is assumed that GPS is accurate to within 3-10 metres depending upon terrain and weather factors or even being inside buildings. As such this needs to be tested. An experiment was conducted on Friday the 31st of March 2013 to see what would be the worst case scenario of the pollution meter running inside a building. Although this does not establish a concrete worst case scenario it establishes an empirical worst case scenario. The result is shown on the pollution visualisation map below.

Figure 38: Location Experimental Error

In the above figure this is the worst case situation as any further obstacles would lead to a loss of signal. Hence experimentally the worst case error that we can calculate can be set at around 70 metres. This is an absolute worst case and in general operation on the roads the worst case is well within the 3-10 metre official range. To derive this worst case a Nexus 4 was used, however different phones due to different architectures may have different results.

The second case to look at accuracy is what happens when the tunnel algorithm kicks in when GPS is not available. In this case the worst case can be any value because all it does is draw a straight line between the start and endpoints. In a real life situation on a road the largest curved tunnel in Sydney is the M5 tunnel and the worst case error in that situation is 300 metres. However on tunnels which are straight this error is minimised significantly and the strategy is continually successful. The figure below demonstrates this issue.

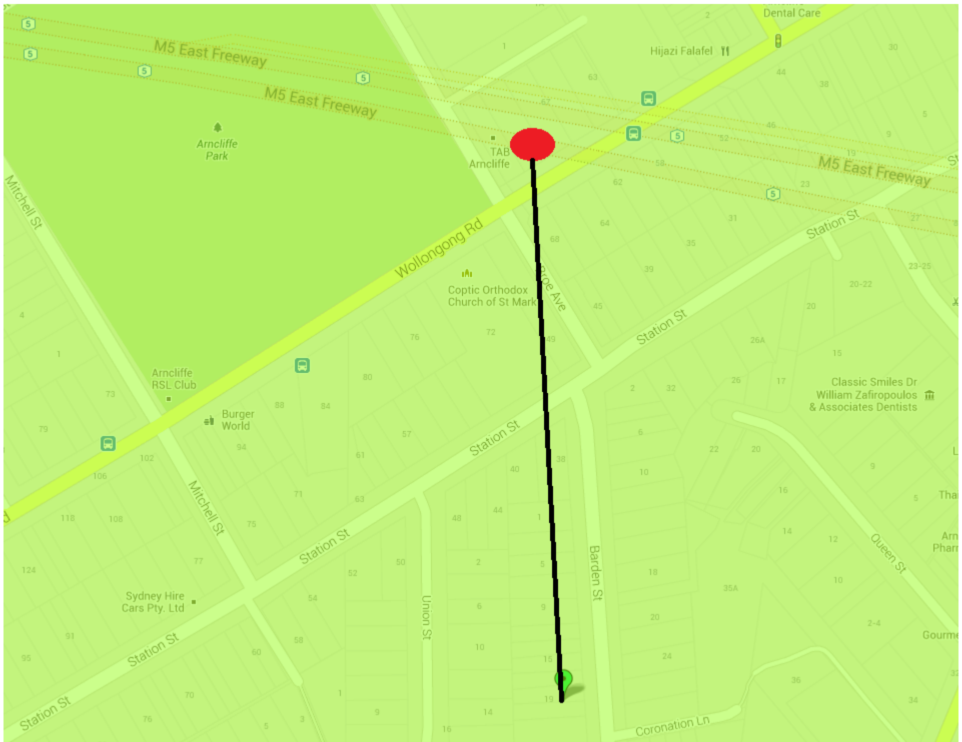


Figure 39: Location Experimental Tunnel Error

There is a solution to fix this issue. This is known as dead reckoning. The idea is to use the inbuilt accelerometer and gyroscope, and use the past best known position to estimate the current position. This strategy has been implemented by experimental autonomous vehicles. However the pay-off this strategy may not be high enough to warrant further investigation, as currently the worst case error is not approached unless in the M5 tunnel.

6.3 Data Quality Analysis:

The final area of analysis is the quality of the data, and this is important to know how much error there is in the data when its uploaded to the server. The important areas for analysis because the data can be affected by a range of factors. The major issues are as follows:

- Effects of Low Battery:
  As mentioned earlier the low battery life of the device will change the input and reference voltages of the sensor hence causing errors.
- Data Bounce:
  The data varies considerably below 5 ppm and continually bounces and this needs to be analysed.

6.3.1 Effects of Low Battery Life:

The device as mentioned earlier does not provide battery information to the mobile application, and the calculations are based on the assumption that the device will continually operate at 5V but as seen in the battery life (Appendix B) report this is not the case as the voltage regulator, low voltage output is not utilised in the design of the device. As such analysis had to be undertaken on what the worst case for the data would be. This is shown in the figure below.

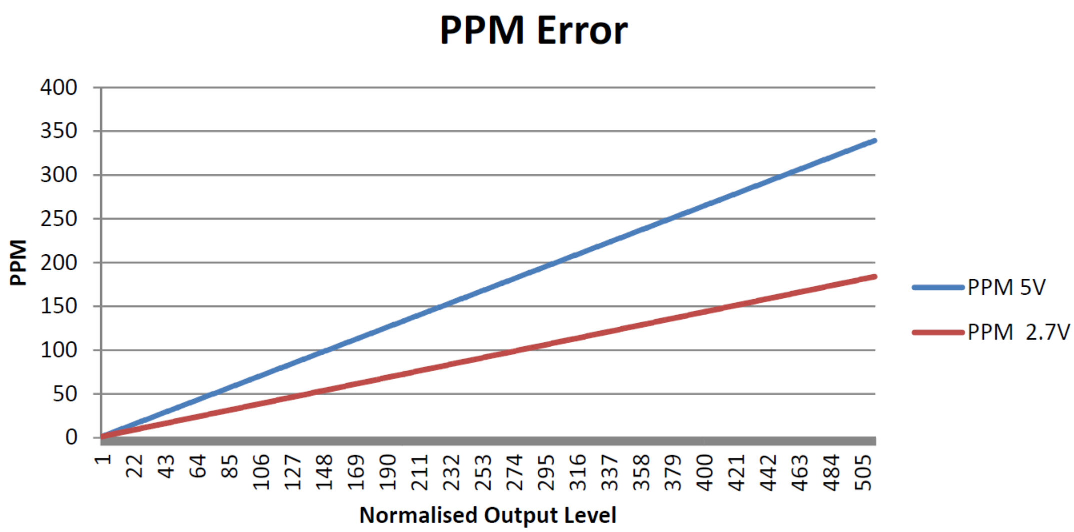


Figure 40: Error at different Voltage Levels

As seen in the graph there is a divergence between the real value found at 5V and the value found at 2.7V. This divergence is significant as it indicates the error in the system. Since the drop out voltage is 2.7V this means that this error is the worst case. In this case the best case error is 0% and the worst case is 46%. However the error increases at a logarithmic rate as shown in the graph below and further explained in the report in Appendix B.

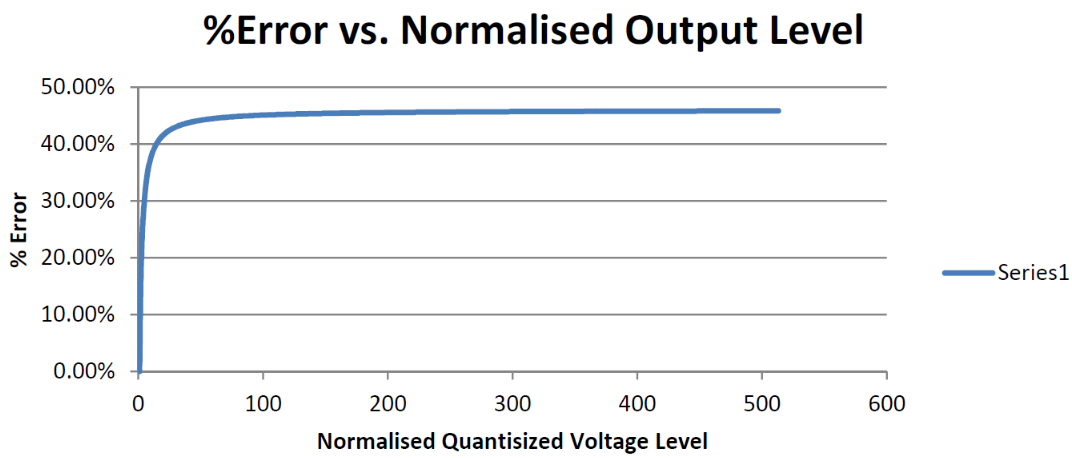


Figure 41: Change in Error

As seen in the above graph the worst case error can be assumed to be 46% throughout because of the steep climb at the start. This means that using the device after the 56 hour mark is inadvisable as accuracy is a major concern and this error rate is unacceptable for that purpose.

To resolve this issue I would suggest redesigning the board to send battery information to the mobile application, thereby allowing for on the fly changes in Reference Voltage data, leading to a more accurate data and a device with a maximum battery life of over a 126 hours.

6.3.2 Data Bounce:

This issue is not as significant as the above issue however it needs to be examined because it affects the accuracy of the data. This bounce in the data is evident at low values while it does not occur at high values above 5ppm. This is also unfortunate because the commercial meter can not detect concentrations below 5ppm, and hence it is not currently possible to know if this bounce is a reasonable statistic.

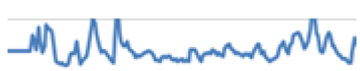


Figure 42: Data Bounce under 5ppm

It should be noted that the pollution meter values are averaged over a 10 second period or more specifically over a 100 samples to arrive at the values displayed on the above figure. This strategy should eliminate momentary spikes that are clearly errors although it may not entirely eliminate large spikes. As such a better calibration method is required to find out if this is constant variation is an issue or not.

7 Recommendations and Future Work:

As seen from the above analysis there are areas where improvements can be made. These improvements are part of the cyclical quality management system as it a step in the continual improvement in the system. The areas that can be improved in are all three sections of the Haze Watch project, these are the Device, and the Android application. In terms of future work an area that needs to be looked at is running more field trials and continued system analysis.

7.1 Recommendations for the Device:

The device needs to go through a revision process. This is so that it can resolve the bluetooth connectivity issues that plagued all of the field trials, and the issues related to battery life. This needs further analysis of the circuit to know why this is happening and the fixes need to be implemented, so that battery life can be improved significantly and by reducing bluetooth disconnections it will reduce the number one cause of user frustration with the device which is being forced to open the case and jiggle the batteries.

7.2 Recommendations for the Application:

The application also has areas that it could be improved upon, from the analysis above using a dead reckoning style strategy would be more accurate than the current strategy which is only adequate. Another area that would immensely improve the flexibility of the application is to remove the list of device id's and their corresponding mac addresses from the application itself and instead have it on the server. The application can download the list as necessary. This will allow future updates to the list to be made at one place and the updates will flow through to every application. This will improve the scalability of the application.

7.3 Recommended Field Trials:

The system is working, and has been demonstrated to be accurate. The next step is to see if there is a way to predict pollution data. From the earlier field trials there is a link between congestion and carbon monoxide pollution. Therefore this needs to be investigated by running more field trials in similar conditions. The suggestion is to compare peak hour traffic as it will have a similar form over 5 days of the week and it has a high level of congestion suggesting it will record values above the 5ppm limit set by the commercial meter and thereby avoiding the issue of data bouncing.

8 Conclusion:

This thesis began as an attempt to build a mobile application to seamlessly interface a pollution monitoring device and a server. However it grew to encompass testing not only the application but also the system and the trials that were run suggested that the system was working well and the analysis of the components of the system found issues that were hidden away by the overall system view.

The application can be declared a success, it has managed to meet the requirements of a good user interface and it provides the user with pollution data in a intuitive and non-intrusive way. It also managed to achieve the seamless interconnect between the monitor and server while providing timestamps and geotags even when no GPS signals were present and it managed to handle drop outs in the network.

The field trials established that the system functioned accurately and was as accurate as the commercial meter. It also established that pollution is localised both in the spatial and temporal dimensions. However it left open the tantalising possibility that pollution is a direct function of congestion and as such could be predicted based on that fact.

In the analysis section, shortcomings of the various sub-systems were found and these imposed hard limits for the current systems, but on the positive side they provided concrete areas in which improvements could be made thus being an integral contribution to the Haze Watch project.

Ultimately with reference to the aims of my thesis I have been successful in what I set out to achieve.

References

[1] Department of Health 2009, Summary of air pollution in NSW, NSW Department of Health, NSW Australia, viewed 5th September 2012, Available at: http://www.health.nsw.gov.au/publichealth/environment/air/air_pollution.asp

[2] Department of Environment and Heritage, 2012. Sydney Air Monitoring Sites, NSW Department of Environment and Heritage, NSW Australia, viewed 5th September 2012, Available at: http://www.environment.nsw.gov.au/AQMS/sitesyd.htm

[3] World Health Organisation, 2008, Public Health & Environment (PHE): Outdoor Air Pollution, World Health Organisation, United Nations, viewed 5th of September 2012, Available at: http://gamapserver.who.int/gho/interactive_charts/phe/oap_mbd/atlas.html

[4] EPA Victoria, 2013, What is Air Pollution, Environmental Protection Agency Victoria, viewed 2nd of March 2013, Available at: http://www.epa.vic.gov.au/air/aq4kids/pollution.asp

[5] Reuters, 2011, What is really going on with volcanoes?, WordlessTech, viewed 2nd of March 2013, Available at: http://wordlesstech.com/2011/01/24/what-is-really-going-on-with-volcanoes-whats-beneath-yellowstone/

[6] SES, 2010, Bushfire season to arrive early and be another busy one, Special Emergency Services, viewed 2nd of March 2013, Available at: http://www.ses-wa.asn.au/node/787

[7] Peter Essick, 2013, Air Pollution, National Geographic, viewed 2nd of March 2013, Available at: http://environment.nationalgeographic.com/environment/global-warming/pollution-overview/

[8] Sjaak Slanina, 2013, Air Pollution Emissions, The Encyclopedia of Earth, viewed 2nd of March 2013, Available at: http://www.eoearth.org/article/Air_pollution_emissions

[9] Marketing, 2012. Android dethrones Iphone as most owned smartphone platform in Australia, Marketing Magazine, viewed 11th September 2012, Available at: http://www.marketingmag.com.au/news/android-dethrones-iphone-as-most-owned-smartphone-platform-in-australia-17252/#.UHJcCk3MjoE.

[10] Nikolaus Youdale, 2010, Haze Watch: Database Server and Mobile Applications for Measuring and Evaluating Air Pollution Exposure, Thesis B Report, viewed 15th of April 2013.

[11] Dawei Lu, 2011, Android interface for pollution Android interface for pollution Android interface for pollution, Thesis B Report, viewed 15th April 2013.

[12] ISO, 2013, AS/NZS ISO9000, Quality Management Systems Fundamentals and Vocabulary, International Standards Organisation, viewed 4th of March 2013.

[13] Android, 2013, Developer's Design Principles, Android Developers, viewed 16th April 2013. Available at: http://developer.android.com/design/get-started/principles.html

[14] Judy Liu 2012, Carbon Monoxide and Nitrogen Dioxide Calibration Results, Thesis Report, viewed 10th November 2012.

[15] Matthew Kelly 2012. Haze Watch: Design of a wireless sensor board for pollution monitoring system, Thesis A report. Viewed 11th September 2012.

Appendix A: Email

Gmail by Google

Vishnu U <vishnu.unni@gmail.com>

Haze watch pollution

James Carrapetta <j.carrapetta@unsw.edu.au> Thu, May 9, 2013 at 10:10 AM
Reply-To: j.carrapetta@unsw.edu.au
To: Vishnu Unnikrishnan <vishnu.unni@gmail.com>, Vijay Sivaraman <vijay@unsw.edu.au>

Hi Vishnu,

I am not sure if you are familiar with the board design but I was having a closer look at the Bluetooth module and I notice that there don't appear to be as many connections to the chip as I used on my design. I have a suspicion that this would be the cause for the connection errors and any other Bluetooth problems you experience. I tried connecting to the device today and still can't get it to connect.

I know when I was designing the board there were a few connections I originally missed (from memory CTS or RTS pins). I was still able to connect to the device and get it working, but I was experiencing similar problems and it wouldn't be a very reliable process. After connecting those pins up, it significantly improved the performance of the Bluetooth. It might be worth investigating this. You may even be able to make a small modification with these boards (maybe just run a small wire between pins). But it would be worth checking my PCB layout and the one for this design.

Any tips you have for connecting the phone and the device as well?

Regards

JAMES CARRAPETTA | Project Administrator

Australian Centre for Space Engineering Research (ACSER)

University of New South Wales | SYDNEY | NSW | 2052

Mob: +61 431 868 303 | j.carrapetta@unsw.edu.au

Tel: +61 2 9385 4526 | www.acser.unsw.edu.au

On 8/05/2013 1:47 PM, Vishnu Unnikrishnan wrote:

> http://db.tt/O3EoZs5C

Appendix B: Battery Life Report

Pollution Monitoring Device Battery Life:

Abstract:

The battery life of the device using Varta AAA batteries is 56 hrs assuming a 5V cut off. Unfortunately this does not hold true, leading to the following issue that the device continues to work at values below 5V and the calibration equations cannot hold true. Hence the values reported at any time after 56 hrs is incorrect. Further this issue led to test the 3.3V linear regulator and it was found that this does not work as expected and outputs a voltage of 4.6V at 5V input which exceeds the maximum voltage supply of the bluetooth module at 3.7V. This issues need to be addressed.

Introduction:

As requested I undertook to measure the battery life of the device to ascertain how long the device would function before batteries would need to be replaced. As such this is a short report on the results of this test.

Hypothesis:

It was expected that the device would function until the voltage dropped below 5V, as per Matthew Kelly's thesis, "As soon as the battery voltage is less than the target voltage the device ceases to output the target voltage" (Kelly, 2008). Also from the thesis, it is conservatively estimated that battery life will be 2-3 hrs, however a note that suggests that battery life is well in excess of this estimate. From speaking to James V, it is estimated that the battery life will be approximately around 20hrs as the previous device had a battery life in that range.

Method:

The method used to set up the test is as follows. A device is set up with fresh batteries and a measure of the voltage across the power supply is taken at regular intervals. This process is kept up until the device stops transmitting from the block diagram this can be identified as when the green LED turns off.

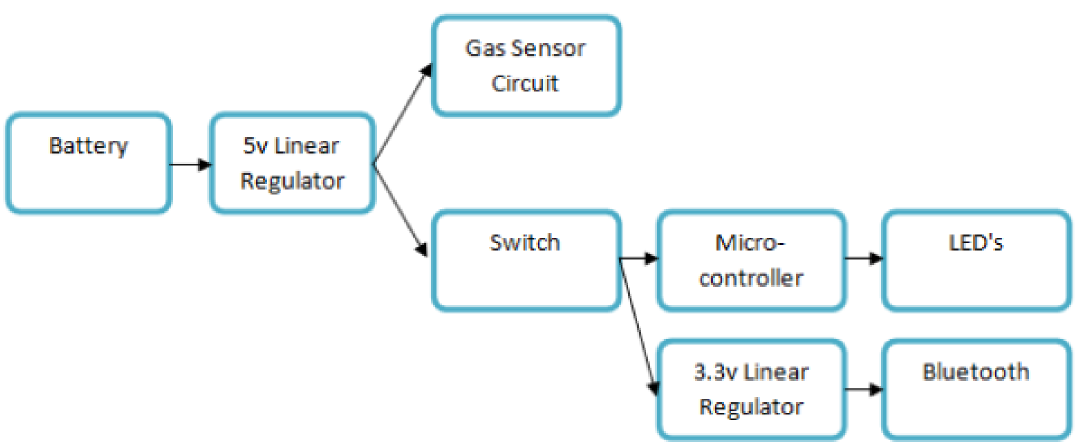


Figure 1: Power Diagram of a wireless Sensor Board (Kelly, 2012)

To check this assumption the android application can be used to connect to the device and see if it receives data.

Ideally this process would be repeated with multiple devices using multiple brands of AAA batteries.

Results:

The results for this experiment are displayed below in the table. This has also been plotted on a graph.

Time (hrs)	**Voltage (V)**	**Time (hrs)**	**Voltage (V)**
0	6.34	30.5	5.94
1	6.15	31	5.92
1.5	6.05	32.5	5.89
2	6	33	5.86
2.5	5.98	33.5	5.84
3	5.94	34.5	5.81
3.5	5.92	35.5	5.80
4	5.89	36.5	5.73
4.5	5.86	42.5	5.74
5	5.84	45.5	0.00
5.5	5.81	54	5.72
6	5.8	56	0.00
6.5	5.73	57	5.70
7	5.74	58	0.00
8	5.72	59.5	5.67
9	0	61	0.00
10	5.67	73	5.66
11	5.66	75.5	0.00
12	5.63	77	5.63
18.5	5.50	78.5	5.50

19.5	5.48	79.5	4.79
21.5	5.44	81	4.75
22.5	5.42	83.5	4.7
23.5	5.41	94.5	4.3
24.5	5.40	97	4.23
25.5	5.39	99	4.15
26.5	5.38	103	3.84
27.5	5.37	106	3.64
28.5	5.36	120.5	1.37
29.5	5.35		

Table 1:Measured Battery Voltage vs. Time

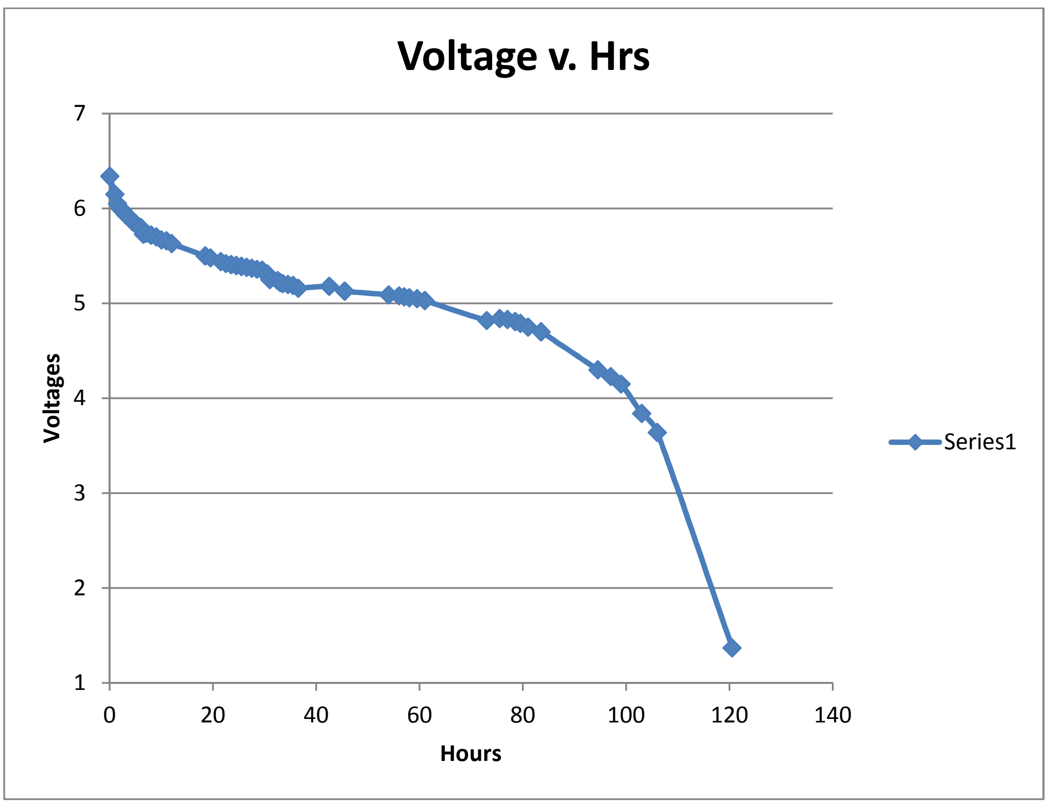


Figure 2: Plot of Voltage vs. Time

From the data above the predicted life according as per the details provided by Matthew Kelly is approximately 56 hrs far in excess of the 20 hrs predicted based upon a previous system.

The data gathering has 4 periods were no data was gathered for a period of more than 3 hours, this occurred during the night.

Analysis:

From the data above the predicted life according as per the details provided by Matthew Kelly is approximately 56 hrs far in excess of the 20 hrs predicted based upon a previous system.

Even though it was possible to calculate the battery life of the device there has been an unexpected development. The assumptions originally made that the green LED will turn off has not held. To test if the system was working, the android application was used and it was found the system was still working and was continually transmitting data.

The root of this problem can be traced back to the linear regulator which instead of blocking the lower voltages, continues to output it. This means that there will be an error with the Reference Voltage and hence the calibration equations cannot be applied, when the voltage drops below 5V.

It was during this stage that another issue was found the 3.3V linear regulator is not functioning properly in two of the devices that I tested. In both cases it output 4.6V, considering that it is of a similar type to the first linear regulator it can be assumed that the device will continue to function until it reaches 2.7V the minimum level required by the bluetooth module to transmit. It should also be noted that the 4.6V exceeds the maximum allowable supply voltage of the bluetooth module which increases the likelihood of burning out the module.

Taking this into consideration it seemed necessary to find the maximum error. As such the calibration equations were used with Vref being set at 2.5V and 1.35V the worst case where the module will have enough power to transmit, and the analogue circuit will continue to operate.

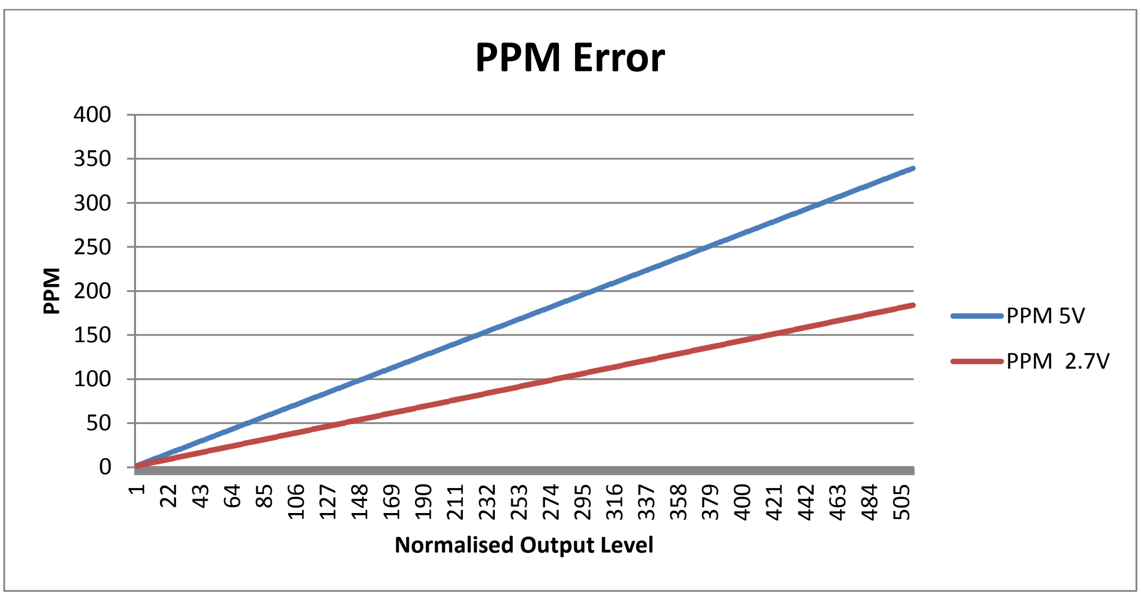


Figure 3: Difference in PPM readout due to supply voltage change

From the above diagram the error grows with each division. These divisions are equal to the levels of quantisation as outputed from the device to the android application. From the data analysis it can be seen that the minimum error is 0% while the maximum error is at 46%. However the error percentages increase in a logarithmic fashion as displayed by the graph. Hence at a low output voltage (low PPM) the error will change rapidly while at high PPMs the error will be a constant.

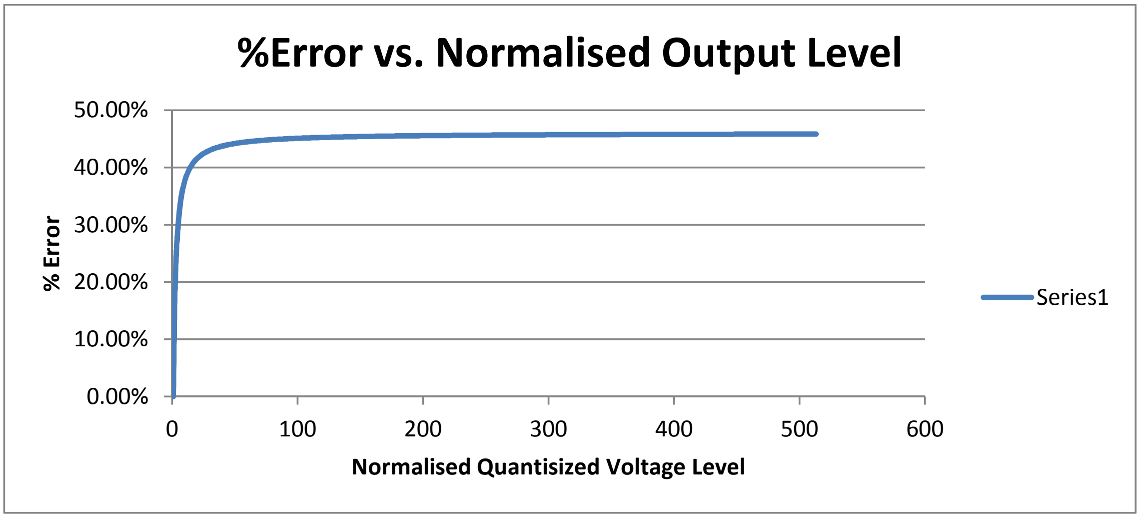


Figure 4: Percentage error vs. Voltage Quantisation

Conclusion:

From the information above it can be concluded the device has an effective battery life of about 56 hrs, using Varta AAA batteries. This may vary with different quality batteries. However it should strongly be noted that that the device has a significant flaw where it will continue to operate below the threshold of 5V. This can lead to incorrect readings being sent to the server and even at low voltages significantly wrong readings can be made. Hence this flaw needs to be addressed.