Snout

A mobile sensing platform

Social Tapestries

Technical Documentation

Proboscis inIVA and Birkbeck College London

Prepared by J.Taylor with contributions from:

Demetrios Airantzis, Karen Martin, Alice Angus, George Roussos, Giles Lane and Orlagh Woods

December 21, 2007

Abstract

This report starts by giving an overview this project and its main objectives. It then gives a brief background of sensor networks and their applications. Various constraints and limitations of wireless sensor networks are outlined together with current methods of tackling these issues. It then describes benefits that mobility can introduce to sensor networks and presents current state of the art in this field. It then proceed to evaluate *participatory sensing* as part of this project in which a purpose built portable pollution sensing platfrom was built and tested. It also demonstrate how this platform was easily extended to surpass its original purpose and was later used as an indoor tracking module using RFID technology.

Acknowledgements

Snout is a collaboration between Proboscis and Birkbeck College, commissioned and supported by inIVA. The project was conceived and developed by Giles Lane, Alice Angus and Orlagh Woods for Proboscis with Dr George Roussos of Birkbeck College. Proboscis' Consultant in Residence, Sarah Thelwall, also made significant contributions to the project's development and public event.

The Snout costumes were designed and made by Alice Angus and Orlagh Woods in consultation with electronics engineer, Demetrios Airantzis, who configured and built the gumstix wearable computer, power source, sensor array and LED display boards. Jenson Taylor (Birkbeck) worked with Demetrios, programmed the Snout sensing platform and prepared the technical documentation. The Snout web interface for mapping the sensor data to other online information about the local environment was created by Karen Martin.

The performers who wore the Snout costumes for the public 'carnival' event were Jordan McKenzie (as Mr Punch) and Bill Aitchison (as the Plague Doctor). Publicity photos were taken by Thierry Bal and video by Tom Dale. Special thanks to the team at inIVA: Gary Stewart, Helen Idle, Natasha Anderson and Josie Ballin; the staff at Cargo; and to Lucas Zamboulis and Alexandre Guitton.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

This project(*"Snout"*[1]) aims to explore and evaluate the use of participatory sensing[2]. As part of this project a mobile sensing platform was developed that was carried by users to collect data about their environment. The gathered data was later mappend onto the location in which it was collected using a geo-centric web service, allowing a visual presentation of pollution in the environment.

1.2 Snout

"Snout[1] is a collaboration between inIVA, Proboscis and researchers from Birkbeck College exploring relationships between the body, community and the environment. It builds on our previous collaboration on Feral Robots (with Natalie Jeremijenko) to investigate how data can be collected from environmental sensors as part of popular social and cultural activities. Scavenging free online mapping and sharing technologies as a form of 'guerilla public authoring', the project also explores how communities can gather and visualise evidence about local environmental conditions and how that information can be used to participate in or initiate local action. Snout will create two prototype sensor wearables based on traditional carnival costumes. Carnival is a time of suspension of the normal activities of everyday life – a time when the fool becomes king for a day, when social hierarchies are inverted, a time when everyone is equal. There is no audience at a carnival, only carnival-goers. Snout proposes 'participatory sensing' as a lively addition to the popular artform of carnival costume design, engaging the community in an investigation of its own environment, something usually done by local authorities and state agencies."

1.3 Project Goals

1.3.1 Snout Social Objectives

One of the required outcomes of Snout is to prepare means by which local communities and non-expert people can become capable of investigating their environments, in order to take necessary steps for improving the public and community health.

The concept of media scavenging is explored in this project; meaning finding free and easy to use resources on the internet to build what would normally require an experienced software developer to create. For example to build a customised website which would analyse or display gathered data requires a fair amount of work by a software developer however already built systems like this are offered on the internet which not only are free but also provide an easy to use interface or configuration panel for integration with other websites. Also a number of online services offer the creation of groups, forums and mailing lists together with WYSIWYG interfaces to build content pages. Such free and easy to use services means non-expert users can benefit from improved electronic means for communication at no extra cost and therefore are able to collaborate more efficiently on a project.

It is important to note, that the target users are not expected to have the necessary knowledge for developing equipment intended for environmental monitoring or creating the necessary website for analysis and visualization of the gathered data. Hence Snout aims to provide an easy to adapt framework for development and publications of such environmental observation projects. A guide to such a framework can act as a recipe for local communities to organize and develop environmental observation campaigns more accurately targeted at their concerns in the immediate locality.

1.3.2 Participatory Sensing

This project aims to explore and evaluate the use of participatory sensing[2]. Participatory sensing allows people carrying everyday mobile devices to act as sensor nodes and form a sensor network with other such devices. A large number of mobile phones, PDAs, laptops and cars equipped with sensors and GPS receivers exist which could be used in participatory sensing applications. The network of such devices can interact like any sensor network to collect and share local data or be part of a larger scale network, for example the sensor network could be queried through internet.

1.3.3 Geo-centric Data Visualization

As this sensing platform is aimed to be used by non-technical users, it is important to provide data in a form which is meaningful to this audience. One of objectives of this project is to evaluate various methods of representing data. Particular the use of geo-centric services such as Google map/earth is to be explored for associating sensor readings and their locations.

1.3.4 Real-time In-situ Feedback

It is important to notify users of the nature of the environment they are in, hence the provision of real time feedback to users is desirable. The aim is to provide a simple yet informative way of continously updating the user about his/her environment.

1.4 Intended Audience

This report uses terms that are regularly used in electronics and computer science. It is assumed the reader has some knowledge of electronic hardware and computer software.

1.5 Overview of Report

1.5.1 Introduction

This section provides an overview of the project and the main objectives and outcomes it aims to fulfill.

1.5.2 Background

This section provides an introduction to wireless sensor networks and outlines the advantages of mobility in wireless sensor networks. It also briefly describes participatory sensing and it advantages.

1.5.3 Hardware

This section summerizes various hardware components that was used in this project to create the mobile platform. It also gives some details about IO allocation of the device which will be essential for the development of the software.

1.5.4 Software Requirements Specification

This section characterizes the desired features of the software application to be run on the platform. Various input/output of the device are outlined, together with how they interact with each other.

1.5.5 Design and Implementation

This section briefly explains how the software was developed and the environment in which it was built.

1.6 Snout Costume Creation

This section describes the process of creating the Snout costumes.

1.6.1 Evaluation

This section reflects on the project demonstration and extensions to the platform. It also outlines future work on the mobile platform and indicates the possible direction of future research.

Chapter 2

Background

2.1 Introduction

2.1.1 Sensing

Observing various aspects of the environment is repeatedly done everyday. For instance humans are capable of sensing temperature, which may stimulate certain actions to accommodate their preferences such as switching on/off the air conditioning system. In a large number of industrial plants, monitoring of equipment is done by humans and their judgement determines when a component is to be replaced. Sensors are used to substitute humans for monitoring events and for sensing physical properties of an environment. Clearly one sensor on its own can not give precise information about the overall attributes of an environment or the state of an industrial equipment. Hence by increasing the number and the types of sensors used, not only is the coverage increased but also the gathered data more closely resembles properties of the actual physical environment. Using a large number of sensor has a number of valuable advantages;

- By using more sensors, radical events are detected easier. e.g. if a large number of sensors were used as opposed to one or two sensors to monitor a house for fire detection, it would allow detection of fire before it spreads whereas if only a few sensors were used the fire would have to spread until it reaches the location of a particular sensor, by which time it might be too late to take any preventive action.
- Gathered data is less biased. e.g. If an office building temperature is being monitored with many sensors and one particular sensor reading is very high due to its particular location, for example it could be placed next to a light. The odd readings of this sensor can be filtered out when compared with the rest of the sensor readings taken in the same room.

- Incorrect reading due to sensor failures can be detected. Sensors are prone to failures and tend to degrade over time. This results in incorrect readings being obtained. Having many sensors allows for early detection of sensor failures.

Management of large number of sensors on per sensor basis is not practical, which is why using a networked set of sensors is appealing. However networked sensors have some constraints. The major shortcomings of networked sensor networks are:

- Cost of wiring and deployment
- Difficulty of maintenance, mainly as a result of cabling issues and lengthy replacement procedure.
- Lack of flexibility in node deployment due to wired nature of sensors both to provide power and retrieve sensed phenomena.

Advances in technology has allowed for development of very small devices capable of processing, wireless communication and also sensing phenomena(physical attributes and properties being sensed are commonly referred to as phenomena in context of sensor networks). In comparison to networked sensors, the two most notable points about these devices are that i) they are very power efficient, which means they can run on batteries for a long time and ii) they can communicate wirelessly, which means there is no longer a need for cumbersome and expensive cabling. The setup of such networks have become popular in various observation applications(e.g. habitat monitoring) and spawn the field of wireless sensor network(WSN).

2.1.2 Wireless Sensor Networks

Overview

Sensor networks are used to closely monitor physical phenomena. Such networks are made up of nodes which have the ability to monitor one or more aspects of the physical world (e.g. light, heat, vibration). These nodes are also capable of communicating with one another and usually a central device which makes use of the gathered information. A number of applications for sensor networks are:

- Habitat monitoring to help protection of rare species[3, 4]
- Battlefield surveillance for detection of enemy intrusions[5]

- Traffic surveillance and control[6]
- Early detection of forest fires, volcano eruptions, flooding and earthquakes.
- Observing dispersion of hazardous chemicals in event of an emergency(e.g. Oil tanker leakage, biochemical attack detection)
- Monitoring conditions of industrial instruments and sub-components (e.g. Monitoring rail track vibration levels with real time feedback)

Due to the nature of applications for sensor networks, nodes are usually required to be very small in size, have some processing capabilities and be cheap since a large number of them are to be incorporated in a network. Also as these networks are typically required to be highly versatile, the ability of nodes to communicate wirelessly adds substantially to the practicality element of such systems. All these features and requirements enforces certain limitations on these networks.

Constraints

To accommodate for the essential features of wireless sensor networks, some drawbacks are inevitably introduced, e.g.

- **Limited energy supply;** nodes must be able to perform the functionalities required from them, however nodes are to be small in size and not to be connected via a physical medium hence it is necessary for nodes to have a reliable energy supply which they can operate from over an extended period of time. Accommodating such a power source within the required size restrictions is one major constraint on wireless sensor network devices.
- **Limited communication range and bandwidth;** as devices in today's sensor networks have a limited energy supply, nodes' communication range has been compromised to a certain extent in favour of energy efficiency . Today's nodes are capable of relaying messages to other nodes until the message reaches the base station(commonly referred to as multi-hop communication). This approach has a number of advantages:
  - Removes the need for sensors to have line of sight with the base station in order to communicate with it.
  - Using this multi-hop approach for relaying messages between short range radio hops is exponentially more power efficient than using larger hops to cover the same distance.

  – Less radio interference is caused among the nodes as each node's transmission range is shortened, which in turn means fewer retransmissions of lost or damaged packets and hence less energy consumption.

- **Limited storage and processing capabilities;** Even though this might not be critical for some applications as the base station is likely to have large storage and processing capabilities, it can affect particular applications where computation and permanent storage is required on per node basis. These constraints will also hinder the adoption of wireless sensor networks in new and emerging fields that need more complex features, because development and maintenance of large scale systems can prove too complicated and costly when adopting such applications to the limitations of today's sensor nodes.

Current research efforts are tackling each of these shortcomings. A large amount research is directed at extending the lifetime of wireless sensor networks for various applications. New methods for conserving energy both on the hardware, the software and the communication side are constantly being developed.

One major cause of rapid power depletion in wireless sensor networks is radio communication. Wireless radio communication consumes far more power than processing[7]. A number of techniques have leveraged on this fact to reduce power consumption of nodes. One such technique is data compression, in which the amount data transferred in reduced as a result of the compression process. Eventhough compression algorithms are processor demanding tasks, the fact that wireless communication uses more power and the data must travel between a number of hops before it reaches its destination makes this approach more feasible. Notably, with increase in network size, data packets will need to be transferred among more nodes which results in exponential power savings[8] when applying data compression.

Another technique used for reduction of wireless communication and hence power consumption is in-network processing and filtering of data. Data aggregation techniques have proved beneficial[9, 10], as less data travels through the network meaning less wireless communication and therefore less power consumption. However data aggregation introduces some delay in the network as data is held up at aggregator nodes while it awaits arrival of more data in order to perform the aggregation.

Other methods which tackle energy depletion from a different angle are energy aware routing[11] protocols. The aim of energy aware routing protocols are to maximise the network survivability(i.e. extending the time it takes before the network is partitioned into disconnected subnets). These protocols focuses on the overall network's connectivity up-time rather than individual power levels of nodes. So a packet might take a less optimised path to its destination in order to ensure the energy consumption of the network is more equally spread among all nodes. Eventhough more power is used to deliver that specific packet, the creation of disconnected subnets is delayed.

In spite of the mentioned improvements there are still a number of shortcomings in this field; For example in a wireless sensor network, gathered data (sensed physical phenomena) is moved from each node towards one or more central points known as sinks. A sink node either has virtually unlimited resources itself or is connected to a base station machine such as a desktop PC which virtually has unlimited power source, processing power and storage capacity. As all gathered data in the network travel towards the sink, the nodes closer to the sink have to relay messages far more often than other nodes. Considering the fact that wireless communication is a relatively power demanding task for a sensor node, it becomes obvious that nodes closer the sink will finish their power source before other nodes. As a result, network will be divided into islands of disconnected subnetworks. More importantly the base station gets disconnected from the rest of the network therefore data will not be delivered to the user. To deal with depletion of energy on nodes close to sinks due to high communication traffic, mobile sink nodes (known as data MULEs)[12] can be effective. Mules move through the network and collect gathered data from static nodes in the network. This technique distributes the network's power consumption more equally among the nodes in the network by reducing energy spent on relaying messages, and as a result extends the lifetime of a network. However mobility of Mule nodes, adds its own requirements(e.g. mobility power consumption). Nevertheless depending on the nature of an application this can be a feasible approach.

Approaches like the one mentioned above compensate for various shortcomings of static sensor networks by introducing mobility. Mobility in sensor networks exposes new possibilities and solutions in this field which are yet to mature.

2.2 Mobility in Sensor Networks

2.2.1 Overview

Realising the potential benefits of using mobile nodes in sensor networks has shifted a fair amount of interest in exploration of mobility in wireless sensor networks and also using robots in such networks. Currently research in the area of mobile sensor networks covers aspects such as clustering, routing, localisation, node relocation and various other application specific properties where use of a limited number of mobile nodes can significantly improve various characteristics of a sensor network. The main advantages to be gained from a mobile sensor network in comparison to its static counterpart, are:

i. prolonging the network lifetime,

ii. improving network coverage and

iii. the ability to repair network holes.

Introducing mobility to wireless sensor networks enables development of applications with a new set of requirements and functionalities that conventional static sensor networks would not be able to provide or efficiently support. Use of mobile nodes in sensor networks opens up a vast amount of new possibilities in this domain. However as solutions to current shortcoming are unravelled other solutions which were applicable in a static sensor network now render useless, hence new challenges are exposed. For example modifications to current communication protocols have been suggested which takes into account mobility of sensor nodes. [13, 14] Suggest adoptions of the LEACH(Low-Energy Adaptive Clustering Hierarchy)[15] protocol in order make it suitable for more dynamic and mobile environments. These adoptions are mainly centred around reinitialisation and frequent updates of local clusters upon nodes joining or leaving them. Likewise new methods for localisation[16, 17] have been suggested that improve upon previous localisation methods. Such modifications and evolution for various aspects of sensor networks are briefly covered in the following section.

2.2.2 Related work

Attempts to discover on how to best benefit from mobility in sensor networks and how to deal with the continuously changing environment in mobile sensor networks have blossomed a fair amount of research, of which some are briefly mentioned here.

A number of algorithms[18, 19, 20, 21] - mainly simulations - have been used to evaluate deployment of sensor networks using mobile sensor nodes in order to maximise network coverage. [18] models a sensor network where all nodes have locomotion capability and are equipped with sensors that can determine the range and bearing of both nearby nodes and obstacles. The nodes are modelled as particles that interpret their distance to other nodes and obstacles as a virtual force which repels them from one another. This allows for the creation of an evenly distributed network while increasing the network coverage as much as possible. To ensure the repelling is not done to the extent where nodes would fail to communicate with one another, a conceptual firm friction force is assumed to be acting on the nodes which avoids them from moving too far apart. [20] Considers use of static and mobile nodes in the same network where static sensor nodes detect coverage holes and compete for acquiring a mobile node to cover that area. The mobile nodes will analyse various requests for coverage and move to the areas which maximises the network coverage while taking into account movement and communication overheads.

[22] Utilises sensor nodes' mobility to relocate nodes in the network in order to replace failed sensors. This avoids creation of holes in the network which could potentially result in disconnected islands of sensor nodes. The relocation of nodes are done in a cascading manner which has two advantages; first the hole in the network gets covered quicker, second, the energy consumption of the relocation task gets spread out between the nodes participating in this task whereas if a single node was to perform this task it might have needed to travel a long distance and therefore consumed too much power by the time it reached the destination which would result in this relocated node failing(due to running out of power) soon after taking its new place.

[23] Looks at simulation of a sensor network with a small number of mobile robots relative to the static nodes. Mobile robots have the responsibility of replacing failed nodes and depending on the algorithm used for replacement of nodes the mobile robot might also be in charge of making decisions as to which mobile robot is to replace a failed sensor. Three algorithms are examined in these simulations; i) A central algorithm in which one robot does all the management task and assigns which robot should replace the failed node. ii) A distributed algorithm in which robots are spread out during the initialisation stage and the static sensor nodes choose one of the mobile robots that is closest to them as their maintenance robot which will be responsible if they fail. iii) Another distributed algorithm similar to the previous one, however in this one the robots broadcast their location as they are moving to replace a failed node. This allows static sensor nodes to select another robot that is closer to them to be their maintenance robot at any given moment in time. Each of the mentioned algorithms have their own advantages in terms of minimising communication and locomotion overhead as well as maximising scalability.

Some work has been done on using static sensor networks in conjunction with mobile robots in order to navigate a mobile robot to a desired location[24]. Other work on localisation and navigation includes one where RFID tags were used to more accurately localise a mobile robot[25]. In the field of robotics navigation of robots have been improved in indoor narrow spaces such as hallways[26] which in domain of sensor networks translates to a decrease in power consumption and faster relocation of mobile robots.

[27] Explores use of static nodes together with a single mobile robot for a home security system in which various emergency events are detected by the sensor network. The robot is informed which takes pictures of the location in which the event was detected and a home server sends these images along with the event details to the user's hand-held device.

[28] Presents the design of a mobile sensor node. This platform is approximately 0.000047 cube meters and it costs approximately $150. Even though this platform is relatively cheap and very small which makes it appealing for use in sensor networks, it can not be used in rough environments because its

mobility feature has been compromised to favour lower power consumption and its small size.

2.2.3 Participatory sensing

Another form of mobility in sensor networks is participatory sensing[2]. In participatory sensing, users make up the sensor nodes such that each user of the sensor network carries a mobile sensing device and gathers data which is shared among all users. Even though this approach is not applicable to all sectors, but it does have a number of advantages which can prove very useful for some applications. One such advantage is the mobility of nodes without any extra costs to the mobile node. Another advantage is that the mobile nodes can be recharged by their owners as part of another device. For example a PDA could run an application to sense noise pollution using its on-board microphone and when the user recharges the PDA, the sensor node is being recharged in fact. Hence the excessively strict and careful consideration of power consumption is eliminated.

Participatory sensing allows people carrying everyday mobile devices to act as sensor nodes and form a sensor network with other such devices. A large number of mobile phones, PDAs, laptops and cars are equipped with sensors and GPS receivers, which are potential candidate devices for participatory sensor nodes. The network of such devices can interact like a conventional sensor network to collect and share local data or be part of a larger scale network, for example the sensor network could be queried through internet.

Another example where participatory sensing could be useful would be the measurement of air pollution using sensing devices mounted on vehicles. Since all vehicles have power supplies and a majority also have a GPS unit, the sensing device can be simplified to a great extent. (Some cars even have sensors which sense air pollution.) Also considering vehicles cover a big area and a large number of vehicles are constantly on the move in different locations - particularly in city areas - a huge amount of up-to-date data can be gathered which closely represents the physical world.

Participatory sensor networks have the potential to allow users to request specific data from specific locations at specified date and times which will be provided by participants who meet those requirements. This type of requests does not have to be limited to human-mobile devices but can also include a large set of already deployed and managed static sensor networks which allow users to freely query them. So people can form their own participatory sensor networks or be part of a hybrid sensor network which is made up of human-mobile sensor nodes and static sensor nodes which allows one to gather a richer set of results. An optimistic vision is to have so many sensor nodes that the sensor network would become a real-time always-on sensor network which would represent the physical world very closely. This network could then be queried for various phenomena in one's locality and return rich sets of results with minimum delay.

Various aspects of participation in such networks are yet to be explored particularly in the area of privacy concerns. E.g. if a mobile device can be associated with a person, it can be queried for sensing particular physical phenomena and assuming the mobile device provides geo-coded data a person can easily be tracked and monitored.

Another aspect which has not been fully explored is how easily participatory sensing can be implemented without disturbing the flow of everyday life for the participants using their current mobile devices. For example if participants' mobile phone would send data on regular basis through its WiFi interface, and as a result draining the battery quickly, users would find it very inconvenient and a disruption to their life. Another example would be where the users were charged for packets of data that was sent using the GPRS feature of their mobile phone handsets. This is only an example attempting to highlight the point that users' priorities must be respected. So a solution to this arbitrary problem would be to use the mobile phone handset's wireless connectivity facility to send data only upon acquiring a free internet connection so no extra charge would incur and also only attempt this when the connection is obtained through the Bluetooth interface of the device, in which case the battery consumption would be much lower compared with a WiFi connection.

The vision for Participatory sensing applications is to allow users to go on about their everyday life while performing data collection tasks without any extra effort. Also with necessary means available to users, they should be able to deploy their own participatory sensing applications and encourage other people to join their network and create a community based sensor network. These networks could then be queried for data, perhaps through internet.

CHAPTER 3

Hardware

3.1 Device Overview

In order to create the mobile sensing platform a number of components were integrated. Each component is individually described is details in the following sections. Figure 3.1 and 3.2 show a brief overview of the final product and various components used in it.

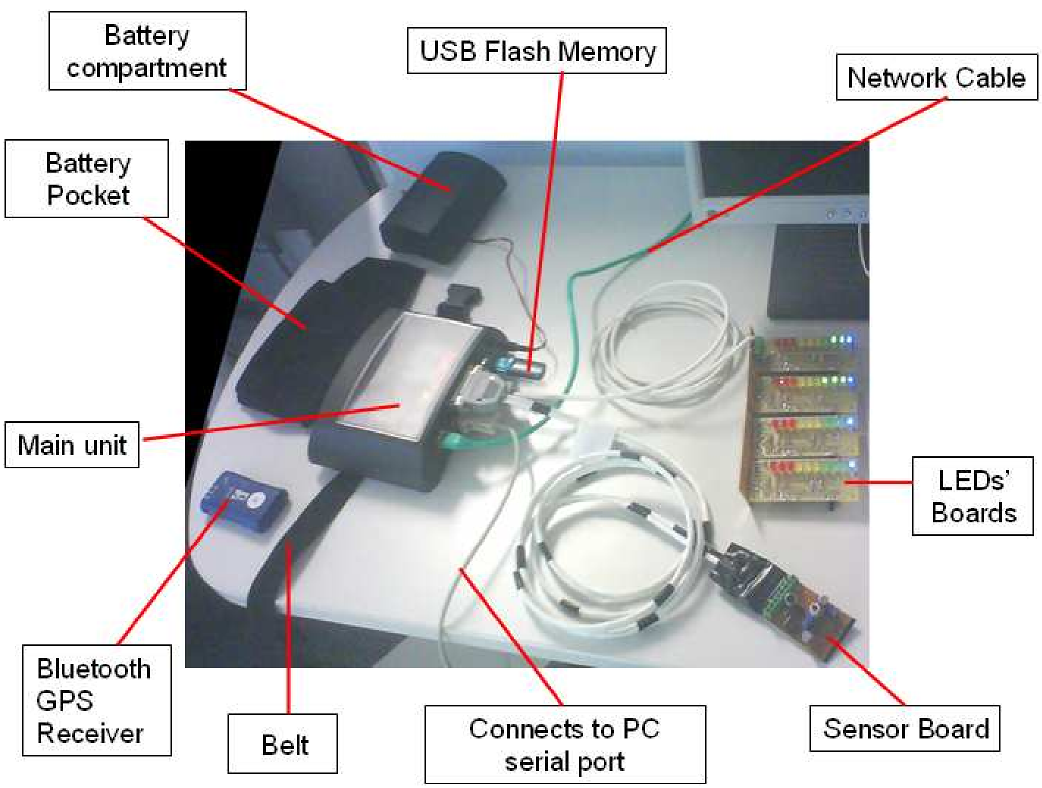

Figure 3.1: Snout platform hardware.

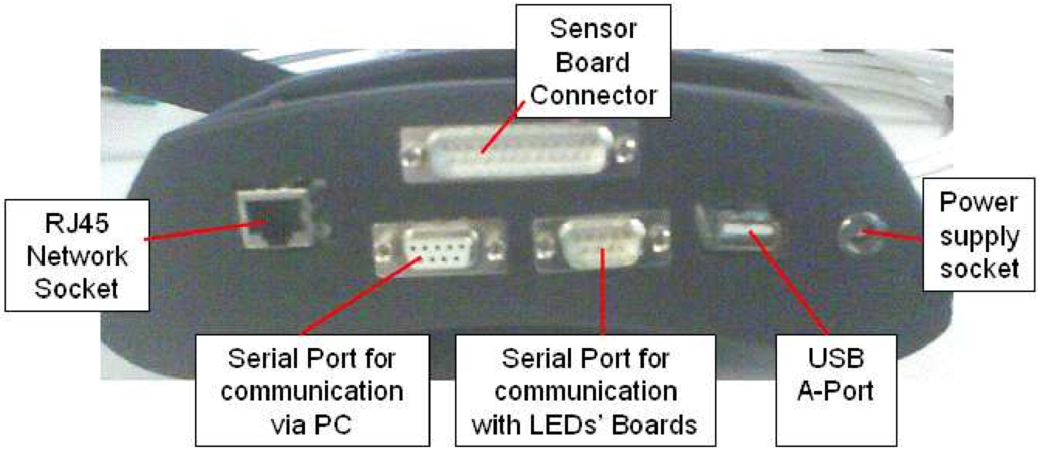

Figure 3.2: Sideview of the main module showing the ports.

3.2 Gumstix

The main processing unit and controlling part of the mobile device is the Gumstix[29]. The Gumstix is a single-board computer. The 'gumstix connex 400xm-bt' model was chosen for this device. It has a 400 MHz Intel XScale PXA255 processor which is a 32 bit RISC architecture based on the ARM design, and has kind power consumption requirements. Gumstix has 64 Mb of RAM, 16 Mb of flash memory and a Bluetooth interface. The dimensions of the gumstix board are 80x20x6mm which makes the gumstix a very attractive platform in terms of portability for a mobile device. Figure 3.3 shows the Gumstix board together with its Bluetooth module.

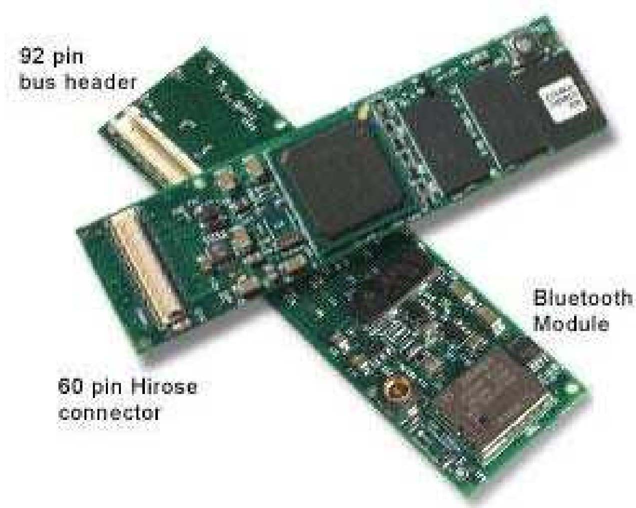

Figure 3.3: Gumstix board used in Snout platform.

3.3 Robostix

Robostix is an expansion board for the gumstix motherboard provided by the manufacturer of Gumstix. Robostix has an AVR ATmega128 microcontroller unit with analogue to digital conversion facility, capable of producing a 10-bit resolution digital output. It has 8 ADC (Analogue to Digital Converter) channels making it over qualified for our requirements, as we are only planning to connect four sensors. The Robostix board also has a large number of IO pins which can be used to control various external devices connected to it. For example LEDs can be connected to the IO pins in order to indicate various states of the device in real time or to control the On/Off state of sensors and other such uses. Robostix runs code as a standalone board however it can connect to the gumstix through one of its serial ports which enables easy cross-application communication which can give the gumstix control over its behaviour and of those sensors or other external devices connected to it. Hence the flexibility of the application is not limited in any form as result of the Robostix executing code in its own environment.

Figure 3.4 shows the Robostix expansion board.

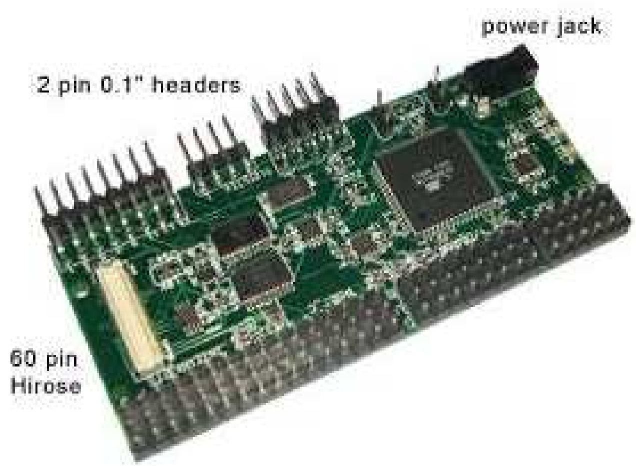

Figure 3.4: Robostix expansion board.

3.4 NetCF

NetCF is another expansion board for the gumstix platform. It has an Ethernet port (RJ-45) which allows for connections to 10-100baseT networks. This expansion board also has a Type II compact Flash adapter which can be used to connect a WiFi interface or extra storage or other devices with this interface. Figure 3.5 displays this expansion board on its own.

The use of this board allows easier and faster upload of executable files to the gumstix during development. Consequently it also frees up one of the gumstix' serial ports to be used for another purpose. It also makes communication with a network and the internet much easier compared with using the serial port of the gumstix board.

According to the requirements of this project the use of this board was not strictly necessary. However it did provide greater flexibility during the software development stage, by using it as a mean for transferring application programs to the Gumstix board, which is why it was integrated as part of the platform.

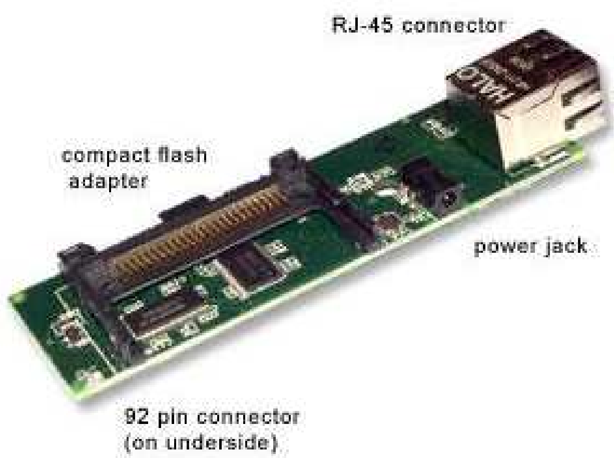

Figure 3.5: Network card expansion board.

3.5 Bluetooth Enabled GPS Receiver

The GPS module used was a 'Socket OEM Bluetooth GPS Receiver'[30]. This module has a Bluetooth interface which allows it to communicate with the Gumstix. Its dimensions are 51x85x21mm and it has its own rechargeable battery which is operational on single charge for eight hours. The noticeable disadvantage it has is the time it takes to obtain the first fix which is typically between 40-70 seconds.

Figure 3.6: GPS Receiver.

3.6 Sensors

3.6.1 Sensor Board

The sensor board hosts four sensors which connects the sensors to the Robostix ADC channels and IO pins for switching the sensors on/off. The four sensors are:

- Acoustic sensor also known as a microphone
- Carbon dioxide sensor
- Carbon monoxide sensor
- Organic solvent vapour sensor (detects benzene, alcohol')

Figure 3.7 shows the sensor board with its four sensors.

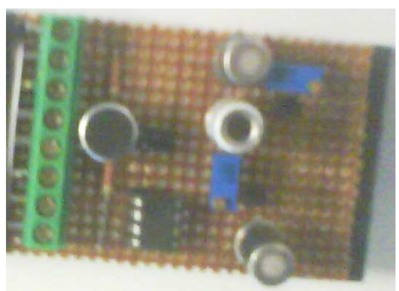

Figure 3.7: Sensor board constituting four sensor.

3.6.2 Acoustic Sensor

We used a capacitive microphone with an RC bias and filtering circuit, followed by an op-amp for amplification to an average voltage of 2.5V. There is nothing special about this circuit. All we need to do is to sample it.

3.6.3 Carbon Dioxide Sensor – TGS 4161

The *Figaro TGS 4161* sensor was used to monitor carbon dioxide levels. This sensor has a high sensitivity to carbon dioxide. It does now show sensitivity to other gases which makes it a good choice for carbon dioxide detection. For further information about this sensor please refer to its datasheet.

Figure 3.8 shows this sensor.

Figure 3.8: Carbon Dioxide sensor.

3.6.4 Carbon Monoxide Sensor – TGS 2442

A *TGS 2442 Figaro* carbon monoxide sensor was used. This sensor requires a heating procedure which is described below:

"The sensor requires application of a 1 second heating cycle which is used in connection with a circuit voltage cycle of 1 second. Each V_H cycle is comprised by 4.8V being applied to the heater for the first 14ms, followed by 0V pulse for the remaining 986ms. The Vc cycle consists of 0V applied for 995ms, followed by 5.0V for 5ms. For achieving optimal sensing characteristics, the sensor's signal should be measured after the midpoint of the 5ms Vc pulse of 5.0V." [31]

Figure 3.9 shows this sensor.

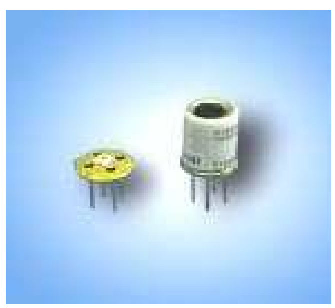

Figure 3.9: Carbon Monoxide sensor.

3.6.5 Solvent Vapour Sensor – TGS 2620

A *Figaro TGS 2620* sensor was used to detect organic solvent vapours. This sensor can detect alcohol vapour as well as solvents such as those used in cleaning products. This sensor is also sensitive to a number combustible gases including carbon monoxide, which makes it suitable for general purpose pollution sensor. The operation of this sensor does not require pre-heating or cooling period, which means it can be done with a simple electrical circuit.

Figure 3.10 shows this sensor.

Figure 3.10: Solvent Vapour sensor.

3.7 LEDs Boards

Four LED boards were constructed, each consisting of 10 Light Emitting Diodes in four different colours.(1 Blue, 3 green, 3 yellow, 3 red). The LEDs represent the sensor readings from low to high starting from blue.

The LED boards are connected to Robostix via its second serial port (UART1). A custom protocol was designed for the LED boards. The protocol defines the commands necessary to switch on/off the LEDs and how to address each LED board. The protocol is thoroughly described in later sections of this document.

Figure 3.11 shows the four LED boards.

Figure 3.11: LED boards.

This LED board uses the CRD89C51RD microprocessor offered by Cyrod Technologies [32]. It features a standard 8051 core alongside a 64K flash memory for storing the application code and 1k of RAM for application data. The program can be updated on the target board through the embedded ISP function of the processor. Up to 36 input/output pins are available alongside a rich set of peripherals like counters/timers, watchdog timer, PWM facility etc.

3.8 VDIP (USB module)

The VDIP module[33] uses the FTDI[34] module to provide a serial to USB interface. Hence gives the ease of program development for a serial interface while providing the convenience of USB support. This module is connected to the Gumstix via on of its serial interfaces (FFUART) which is shared among this and the console access through a PC. The VDIP module has its own instruction set which are quite primitive but nevertheless easy to use and more than sufficient for this application.

More details on how the sharing of the serial port functions is available in the later sections.

Figure 3.12 shows the VDIP module on its own.

Figure 3.12: VDIP module - provides a usb host for the Snout platform.

3.9 Circuit Board Design

Detailed diagrams and circuit board design files are available for download via the Snout website[1].

3.10 Port(IO/Serial) Allocation

3.10.1 Gumstix

Serial Ports

STUART The STUART serial port on the Gumstix connects it to the robostix. See section 3.10.2 for more details.

FFUART This serial port is what developers can connect to to obtain console access to the Gumstix. Another module which uses this shared serial port is the VDIP module which provides a serial to usb host functionality. See section 3.10.2 for more details on how to control the sharing of this serial port.

3.10.2 Robostix

Serial Ports

UART0 The UART0 serial port on the Robostix connects to the gumstix. So the TX line of UART0 connected to the RX line of STUART on the gumstix and the RX line of UART0 on the robostix connects to the TX line of STUART on the gumstix.

UART1 Robostix UART1 connects to the LED boards(four boards) which correspond to sensors' measurements. The appropriate commands to switch LEDs On/Off must be sent using UART1.

IO Pins(ports) / ADC Channels

C.0 - LED board power The C.0 pin(also refered to as port) controls the power line of the LED Board. Logic 1 switches the power On.

C.1 - Sensor board power The C.1 pin controls the power line of the Sensor Board. Logic 1 switches the power On.

C.2 - Carbon Monoxide Sensor Heater The C.2 pin controls the power line of the heater on the Carbon Monoxide sensor. Logic 1 switches the heater On.

C.3 - Carbon Monoxide Sensor Power The C.3 pin controls the power line of the Carbon Monoxide sensor. Logic 1 switches the sensor On.

C.4 This port is not used.

C.5 - VDIP Control The C.5 pin controls the status of the VDIP usb host module. **Logic 1 turns on power which switches the VDIP Off!** During power on and reset of the VDIP module, this pin should be in the desired state. (The C.5 pin is connected to the RTS line of the VDIP module.) The reason for controlling VDIP status is that the VDIP module uses a shared serial port(FFUART on Gumstix), so having the module switched Off ensures no interference on the shared serial port.

C.6 - GPS Fix Red LED The C.6 pin controls the state of a red LED on which indicates whether a GPS fix is available or not. Logic 0 switches the LED On, which means a GPS fix is available.

C.7 - Preheat Red LED The C.7 pin controls the power to a red LED which indicates that the sensors are in the preheat period, so as long as this LED is On, the sensors are not ready. This LED is not strictly necessary as all the sensors used have very short or no heating time, however if other sensors are used this LED will prove to be useful. Logic 0 switches the LED On.

F.0 - Acoustic Sensor ADC Channel F.0 is the ADC channel on which the measurement for the acoustic sensor is obtained.

F.1 - Carbon Dioxide Sensor ADC Channel F.1 is the ADC channel on which the measurement for the Carbon Dioxide sensor is obtained.

F.2 - Carbon Monoxide Sensor ADC Channel F.2 is the ADC channel on which the measurement for the Carbon Monoxide sensor is obtained.

F.3 - Organic Solvent Vapour Sensor ADC Channel F.3 is the ADC channel on which the measurement for the Organic Solvent Vapour sensor is obtained.

F.4 - Battery Voltage F.4 is the ADC channel on which the measurement of the battery voltage is read. Note, the measurement is one fourth(1/4) of the actual voltage.

G.0 - System Ready Green LED The G.0 pin indicates that the system is ready, i.e. the booting and all initialization processes are finished. Logic 0 switches the LED On.

G.1 - USB In Use Red LED The G.1 pin indicates that the usb host is transfering data to the the usb device(flash memory) so this LED should be On to indicate that the user should not unplug the usb device. Logic 0 switches the LED On.

G.2 - USB Not In Use Green LED The G.2 pin indicates that the usb device can be unplugged. So by turning this green LED On the user will know that the usb device is not in use. Logic 0 switches the LED On. Note, if no usb device is available then both LEDs(G.1 and G.2) should be Off.

Chapter 4

Software Requirements Specification

4.1 Platform Requirements Overview

After some considerations on how to satisfy the project goals, the main required functionalities of the platform were determined and agreed upon by all the people involved in prototyping this mobile sensing device.

The platform was required to sense four distinct physical phenomena. These were carbon monoxide, carbon dioxide, organic solvent vapour and sound.

The readings were to be displayed in real time to users of the device in order to provide some form of feedback to the user as to the environment they are in. This was decided to be done by using LEDs ranging from 1–10.

Another requirement was that the users could easily retrieve the collected data. This was to be done through the use of the usb flash memory. So the users would plug their usb flash memory stick into the device and the device would transfer the readings to the usb flash memory. This feature would give easy access of the collected data to the users however this data would not be user friendly to view and analyse. So to provide the results in a nice visual representation, the users should be able to upload the retrieved data to a geo-centric web service like Google Maps to view a plot of the sensor readings on the path they were collected.

4.2 Software Requirements Specification

4.2.1 Introduction

Purpose

This software requirements specification document aims to provide a comprehensive outline of what functionalities and outcomes are expected from the application program. It intends to give sufficient details to the software developer to be able to unambigously implement the proposed system.

Scope

The Snout software to be prepared must acquire sensor measurements, displayed measurements in real time on LEDs, acquire its GPS location and record all this data together with a timestamp for future use. The aim of this is to use the acquired measurements later by transfering them to geo-centric web service such as google map to give users a visual view of the sensor measurements. This software requirement specification document only addresses the software requirments for the application to be run on the sensing platform. This document does not detail the requirements for presentation of the data on geo-centric web services.

The software on the sensing device must start running upon switching on the device, as the device does not have other means for the novice user to interact with it.

Overview

The remainder of the software requirements specification document is organised as follows:

- Overall Description: covers an overview of the Snout application, general constraints and factors that affects this application and its requirement. The major functions that the application will perform are briefly mentioned in this section.
- Specific Requirements: outlines the technical information needed to design the software, such as external interface requirements, functional requirements, flow of data, diagrams, performance requirements and quality attributes.

4.2.2 Overall Description

Product perspective

Overview The system as whole aims to allow the users to monitor the air pollution in their environment. The device is a portable piece of equipment which shall be carried by users in outdoor environments where GPS readings are available, however the system must function in the lack of a GPS fix.

User Interfaces The device does not connect to a monitor, hence its user interface is limited to various LEDs that indicate certain scenarios.

- A set of light emitting diodes are used to indicate the measurement intensity of the sensors.
- A number of light emitting diodes are also used to indicate various states of the system, such as when the application has start up process has completed, when the sensor board is powered, when the LEDs boards are powered, when the usb flash memory is being written to and when it can be removed.

Hardware Interfaces The main processing unit communicates with a number of expansion boards which are part of the same sensing platform. These include:

- GPS receiver;
- Sensors' LED board; used to display the sensor readings in scale(1-10)
- Robostix; provides analoge to digital conversion, a number of serial ports and IO pins.
- Sensor board; contains a number of sensors and is accessed only through the robostix board.
- USB host(VDIP); is used to write data to a flash disk

Communication Interfaces

- Bluetooth communication between main module and the GPS receiver
- Serial communication between main module and ADC board(Robostix)
- Serial communication between Robostix and sensors' LED board

- Serial communication between main module and the USB host provider
- IO pin manipulation for sensors on/off status and also to set the status LEDs

Memory Constraints The main processing unit(Gumstix) - as defined int the hardware section - has 64 Mb of RAM, of which some will be used for the Linux operating system on which the program will.

The Robostix expansion board has very limited memory and processing capabilites. Refer to its datasheet for more information.[29]

Product Functions

The basic functionalities expected from the application running on the mobile platform are:

- Sense pollution(Carbon Monoxide, Carbon Dioxide, Organic solvent vapour and Sound)
- Display readings' intensity via light emitting diodes
- Obtain GPS location by communication with an external GPS receiver
- Store sensor readings together with GPS readings and a timestamp

Figure 4.1 shows the flow of actions necessary by the device and user.

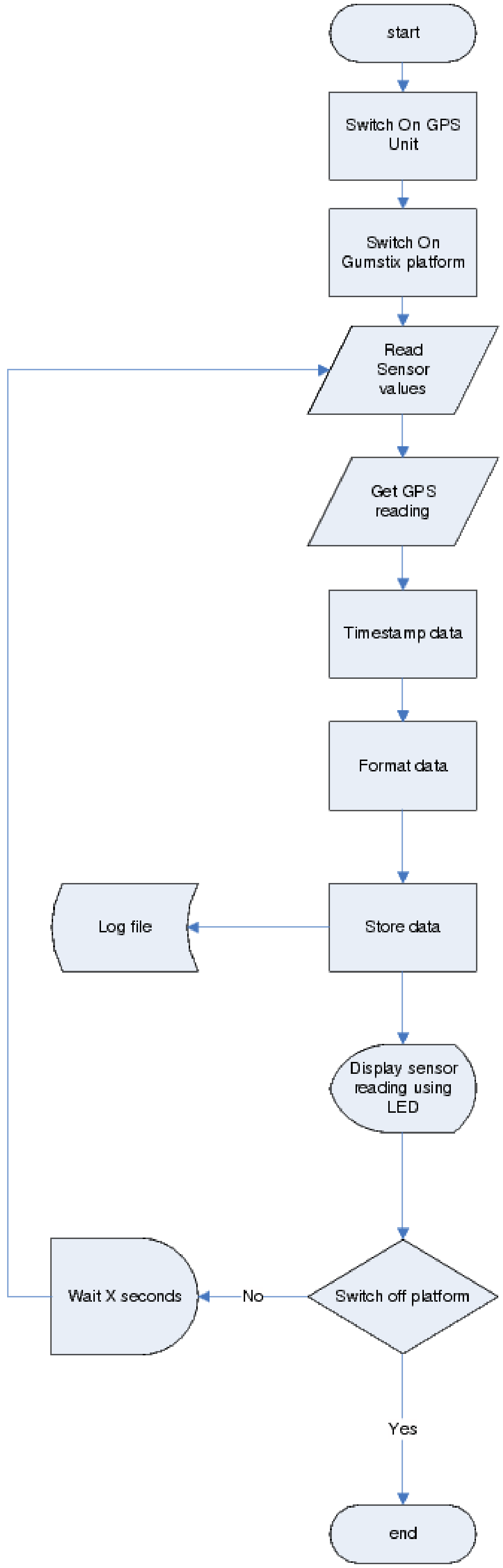

Figure 4.1: Flow of events

User characteristics

The users of the system need not have any technical expertise or previous experience of using such devices. The only action required from the users should be switching the device On/Off. The application must perfom all initialisation automatically upon start up. The users do not need to be presented with a Graphical User Interface (GUI) as no further interaction should be required from the users.

The users must be capable of recharging the batteries and switching the device on upon usage and switching it off once finished.

Constraints

The Robostix expansion board has limited resources both in terms of hardware and software. The programs must be written in C and then cross compiled on an external computer, transfered to the main processing unit(Gumstix) and then uploaded to Robostix through the serial port on which it is connected to the main processing module(Gumstix). The environment in which the code will execute does not have large sets of libraries due to its limited memory. Hence standard POSIX functions will not be available.

4.2.3 Specific Requirements

External Interfaces

Table 4.1: External Interface - Robostix

Name of item	Robostix
Description of purpose	This expansion board has eight ADC channels which are used to obtain sensor measurements. It also has a number of IO ports that are used to switch various parts of the device On/Off. These include LEDs, sensors and the VDIP(usb host) module. For more details on Robostix see section 3.10.1.
Source of input	Robostix receives sensor measurements on its ADC channels(F ports). For more details see section 3.10.2
Valid range, accuracy, and/or tolerance	Robostix is capable of producing a 10-bit resolution digital output from its ADC channels.
Relationships to other inputs/outputs	Robostix is connected to Gumstix(the main processing module) via a serial port(Robostix' UART0 to Gumstix' STUART). It is through this serial port that Gumstix and Robostix can communicate and exchange data. Robostix must wait for Gumstix' instruction to start polling the sensors, which must be done through the serial port and then sensor readings will be returned via the same serial port.

Table 4.2: External Interface - Sensor board

Name of item	Sensor board
Description of purpose	This board containes the four sensors which measure pollution. These include Carbon Dioxide, Carbon Monoxide, Organic Solvent Vapour and an acoustic sensor. This board provides the main source of input for this application. For more details on the board and its sensors refer to section 3.6
Source of input	This board is connected to the Robostix via Robostix IO pins. The exact pin assignments are explained in section 3.10.2.
Valid range, accuracy, and/or tolerance	The values received from the sensors must be scaled between one and ten depending on the intensity of the measurement.
Units of measure	The values received from the sensors must be graded between one and ten depending on the intensity of the measurement for displaying on the LED boards and for a geo-tagged view. However the full values received from the ADC must be logged and available for inspection.
Timing	The sensors should be polled at three(preferably less) seconds intervals. All sensors apart from the Carbon Monoxide sensor do not have preheating periods, i.e. they provide a continoues stream of measurement. However for the CO sensor there is particular heating cycle that must be followed for optimum results. This cycle is explained in section 3.6.3.
Relationships to other inputs/outputs	The measurement retrieved from each sensor on the sensor board must be appropriately scaled, packaged and sent to the LED board. The LED boards are connected to Robostix via its UART1 serial port.

Table 4.3: External Interface - VDIP(usb host)

Name of item	Viculum VDIP(USB Host)
Description of purpose	This serial to usb converter module provides the platform with a a usb host which will be used to store the collected data onto a usb flash memory. So that users can easily upload the data to an online geo-centric service.s
Timing	Once a usb flash memory is detected, logging of data must be done to a file on the flash memory as well as to the Gumstix' permanent storage. The steps involved including those from the user's perspective are shown in figure 4.2.
Relationships to other inputs/outputs	This device shares the serial port(FFUART) of the Gumstix which is also used for connecting to Gumstix and opening a terminal. For more details on using the shared serial port see section 3.10.2.
Data formats	The formatted data must be the same as that of the logs.
Command formats	For a complete set of commands to be used with this module view its documentation[33].

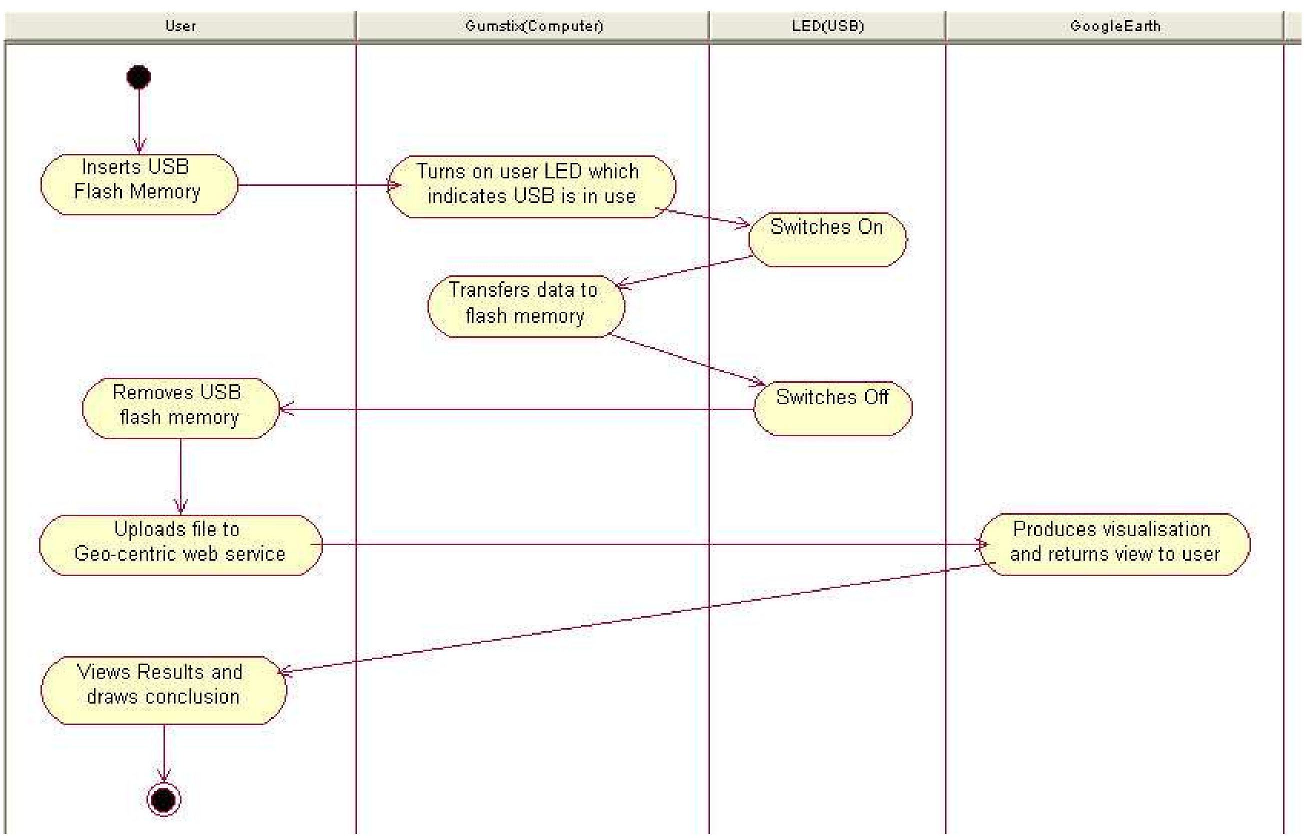

Figure 4.2: USB Logging flow of events.

Table 4.4: External Interface - LED Boards

Name of item	LED Boards
Description of purpose	There are four LED boards that host ten LEDs each. Every board corresponds to a single sensor and it contains ten LEDs. The purpose of these LEDs are to display the sensor reading in a scaled version of the measurements from one to ten. More details on this board is available in section 3.6.5.
Source of input	The input received from the sensor boards by Robostix must be scaled and packaged according to the command format described in table 4.5 and sent via Robostix UART1 serial port.
Valid range, accuracy, and/or tolerance	Each set of LEDs contains ten LEDs which are categorized as the following: • One blue LED which indicates the sensor is up and running. (i.e. the sensor is functioning and has not been damaged.) • Three Green LEDs to indicate a low reading. • Three Yellow LEDs to indicate an intermediate level of intensity measured from the sensor. • Three Red LEDs to indicate a high intensity measurment from the sensor.
Units of measure	The sensor measurements should be scaled from one to ten to accomodate the LEDs available on this board.
Timing	The frequency of updating the state of these LEDs should be equivalant to that of the sensor polling iteration.
Relationships to other inputs/outputs	Each set of LEDs is uniquely identified with an address. This address is assigned by physically changing the location of jumpers on each board. The addressing scheme is as follows: • Address 0: jumper 0 missing, jumper 1 missing • Address 1: jumper 0 present, jumper 1 missing • Address 2: jumper 0 missing, jumper 1 present • Address 3: jumper 0 present, jumper 1 present
Command formats	The necessary commands for switching the LEDs On/Off is outlines in table 4.5.

Table 4.5: LED Board Protocol Example Transmission

Hex	Dec	ASCII	Description	Transmission Order
0x02	2	STX	Start of message character (Start of TeXt)	1
0x08	8	...	Number of bytes in message core (Message size) It is the binary number of bytes contained between the "start of core message" and the "end of core message". In this case, 8	2
0xB1	177	...	Checksum; Sum of decimal byte values Modules 256. • set sum byte variable to zero. • add the ascii value of each character • place here the result of this byte addition. • the sum in this case is 0xB1 (177)	3
—— START of CORE message ——				
0x30 0x31	48 49	0 1	Board address, represented in two bytes. Most Significant Nibble first, Least Significant Nibble second.	4 5
0x35	53	5	Command 5 to be executed	6
0x31 0x31 0x31 0x31	49 49 49 49	1 1 1 1	LEDs to be turned on. This has now become the binary description of the outputs. A bit cleared to zero turns the led on, a bit set to one turns it off. For example, if all leds are to be on, a value of 0000 (48 48 48 48) must be sent. If only the first two leds will light, then the value 03FC (48 51 70 67) must be sent. The blue led is the LS bit, the third red led is the MS bit.	7 8 9 10
0x57	87	W	A "*W*" (87) will indicate a write operation. (Capital letter W)	11
—— END of CORE message ——				
0x03	3	ETX	End of message character (End of TeXt)	12

Functional Requirements

Table 4.6 gives an overview of the requirments together with their priority level.

Table 4.6: Functional Requirements

#	Priority	Name	Description
1	Essential	Sensing	Obtaining sensor measurements from the environments through the available sensors.
2	Essentail	Realtime feedback	In-situ displaying of sensor readings via the available LEDs for the users to have an indication of the environment around them.
3	Essential	Data logging	Store sensor readings together with GPS readings and a timestamp to the platform's permenant storage.
4	Desirable	GPS Location retrieval	Obtain GPS coordinates to include in the log.
5	Desirable	USB Flash memory logging	Store collected data to the usb flash memory once it is inserted to ease the task of retrieving the logged data from the platform.

Software System Attributes

The application must be developed on an external computer and cross compiled and then uploaded to the Gumstix platform. Gumstix runs a stripped down version of Linux which is provided by the Gumstix manufacturer. Thorough details on how to develop programs for this platform is available via the Gumstix website[29]. It is possbile to write programs in variety of languages including C,C++ and Java for this platform.

CHAPTER 5

Design and Implementation

5.1 Software Environment

5.1.1 Gumstix Linux

The gumstix hardware platform is supported by customized GNU/Linux distribution based on the Buildroot system – "a set of Makefiles and patches that makes it easy to generate a cross-compilation toolchain and root filesystem for a target Linux system using the uClibc C library" – and the U-Boot boot-loader.

To develop the application for the gumstix platform a Linux-based(Fedora) host system with a fully-featured GNU software development environment properly set up (autoconf, make, gcc, etc), was used.

5.1.2 Programming Language

The programming language used to develop this application was C. The main advantage that C has over other programming languages is the efficiency it has when running. Although this applisation does not require huge efficiency but having such an advantage gives it longer battery life which is essential for any mobile device. The main part of the application which runs on Gumstix uses POSIX standard libraries, hence it can be run on any Linux platform with minor modifications.

5.1.3 Robostix Environment

There is no analog to digital conversion (ADC) capability directly available on the main gumstix processor board. Thus, the robostix add-on board was used in this prototype, due to the ADC functionality present in its AVR ATmega128 microcontroller. As described earlier, the gumstix and robostix boards are interconnected via their serial ports – STUART (/dev/ttyS2) and UART0, respectively – making data exchange between software running on both processors possible. This feature is exploited to give the application (running on gumstix) access to sensors attached to any ADC channels of the robostix.

The communication protocol on the robostix side consists of waiting for incoming ASCII characters on UART0, that represent ADC channel numbers (from 0 to 7), and responding with the current voltage value on that ADC channel, in hexadecimal format encoded as an ASCII string (e.g. 0x03f9). On the robostix' ATmega128 this function is implemented with an endless loop in a C program.

5.2 Program Flow

The application software drives the platform by instructing the Robostix to sample the sensors' readings at predefined time intervals. Each reading is then time-stamped and together with a GPS reading logged onto the storage unit. The program also controls the LED boards to light up the necessary number of LEDs reflecting the intensity of the measurements. Each sensor has a dedicated LED board which reflects the changes in the sensor reading in real time. Figure 5.1 gives a brief overview of the program flow.

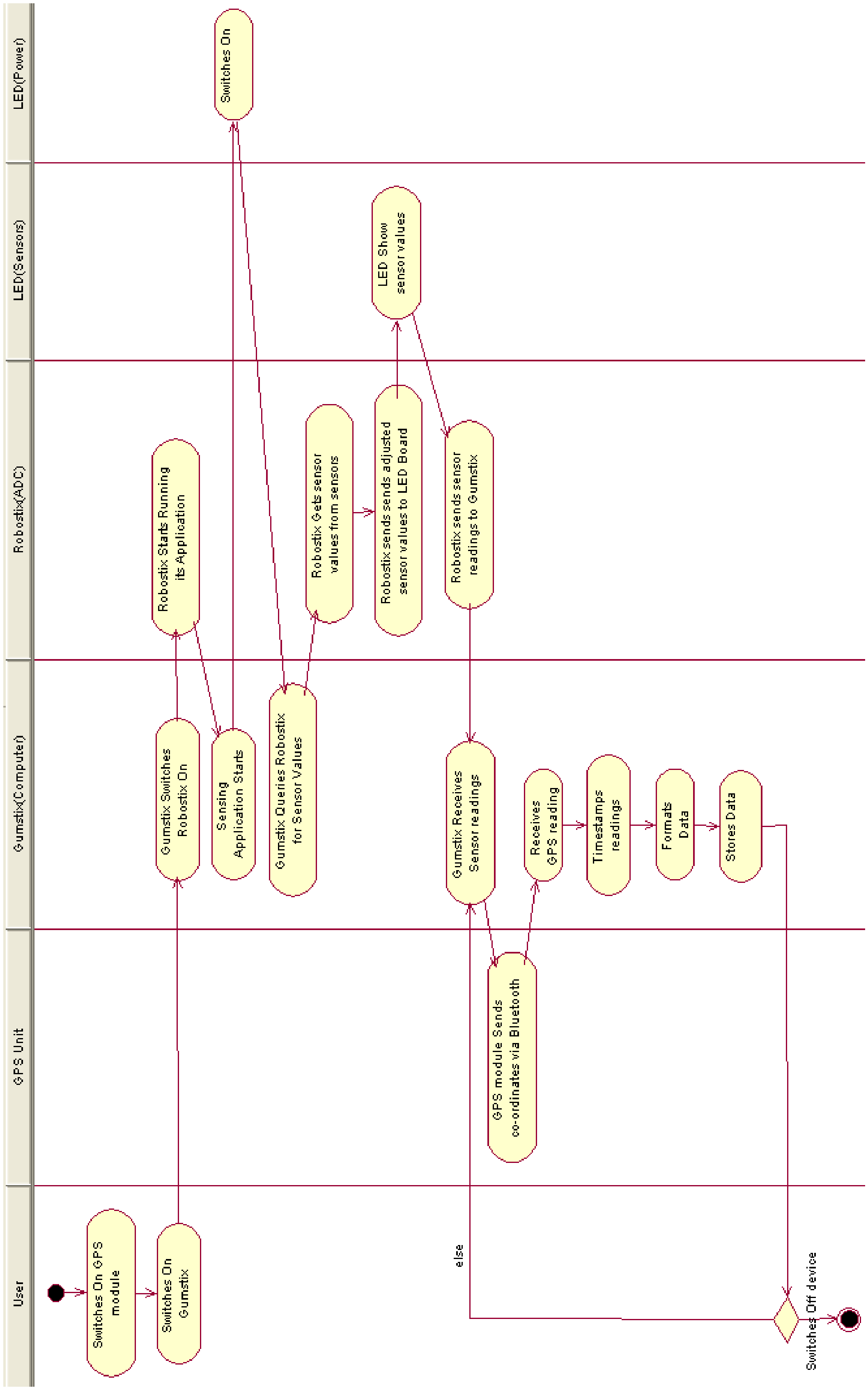

Figure 5.1: Program flow.

5.2.1 Real-time In-situ Feedback

As data is collected using participatory sensing, it is important to provide real time feedback to users as to what the sensor readings are. This is to notify the users of the nature of the environment they are in. So for mobile devices carried in a petrochemical processing plant, if users are in a highly polluted or toxic environment they are notified of the conditions and therefore can take necessary actions based on the information sensed and delivered to them in real time. E.g. close a tap or leave an environment or perhaps both.

Aside from the threatening aspect of highly polluted environments, this feature provides a better user-interaction means for the participants and allows them to monitor their immediate environment in real time. Hence it gives the users a better experience, and naturally a real time analysis of the situation is done by the user. For example if a visibly polluting car is nearby or there is fire in the vicinity, the user will realise that the high reading of carbon dioxide is strongly correlated to the visible smoke. Whereas if the gathered data is later viewed and analysed, the user might assume that a particular location is typically a more polluted area.

Currently the real time feedback is done using a set of ten light emitting diodes representing each sensor. The sensor measurements received from the sensors by Robostix are scaled between one and ten, and this number is sent to the appropriate LED board to show the intensity of the reading.

5.2.2 Geo-centric Data Visualisation

In order for any gathered data to have meaning in some context, certain circumstances in which data is being collected, needs to be known. In environmental monitoring applications the absolute geographical location of the collected data is fundamental. So all gathered data can be geo-coded (acquired data is also time tagged). This will enable the analysis of data to be correlated with affecting variables in that vicinity. Thus necessary actions can be taken to tackle any problems.

The amount of collected data in environmental monitoring application tends to be large, which means the amount of data to be viewed and analysed is large, hence it is necessary to summarise data in an effective way so that users can easily extract the useful information for further scrutiny. One of desirable outcomes of the Snout project was to evaluate various methods of representing data. Particularly the use of geo-centric services such as *Google map* and *Google earth* were explored for associating sensor readings and their locations. This approach allows the user to follow the path of the mobile device and observe changes in the environment visually. One advantage of this approach is that it gives users a more tangible way to experiment with results and to assess the whole monitored environment in a glance.

Producing useful presentation of gathered data was an aspect of the project outcomes. After all it is the gathered data which users are interested in. If the gathered data can be presented in a meaningful way, valuable information can easily be extracted from the presentation and therefore necessary conclusions can be drawn. Figure 5.2 shows colour coded representation of Carbon monoxide readings in an environment.

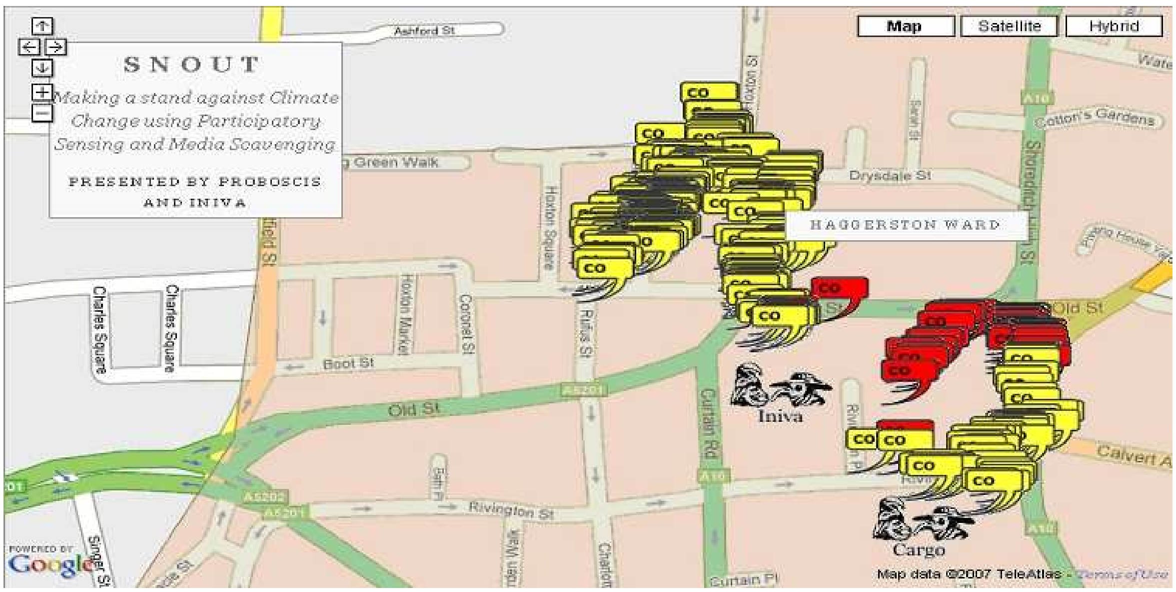

Figure 5.2: Carbon monoxide readings.

5.2.3 Sample Log

Below is a snippet of the log file acquired during a trial. The columns - from left - represent timestamp, sensor name, ADC value, scaled value, availability of a GPS fix, GPS latitude and GPS longitude.

```
3241 CO_2 0x0360 06.00 yes 51.527809   -0.081058
3242 CO__ 0x003c 00.00 yes 51.527809   -0.081058
3243 CSV_ 0x01b2 08.00 yes 51.527817   -0.081053
3243 BATT 0x026f 12.17 yes 51.527817   -0.081053
3245 MICR 0x035d 09.00 yes 51.527817   -0.081053
```

5.3 Scavenged Web Interface

5.3.1 Snout London: The Website

The design brief for SNOUT was to create a 'scavenged' website, on the theme of environmental sensing, by using and combining existing web tools, services and online content into a new format. The SNOUT site aimed to encourage action in two ways: firstly, the site aimed to inform its audience about local issues of environmental pollution and potential action for change that people might take in response to these and secondly, the site acted as an example of the process of scavenging, illustrating how existing web-based tools and information can be combined to create a new product.

Figure 5.3 shows the website.

5.3.2 Ning

The base tool for the construction of the SNOUT site is Ning, a free online service for creating, customising and sharing social networks. The Ning platform provides REST API'S, PHP API'S and JavaScript API's giving access to the core Ning services and database as well as to external web services such as Flickr and RSS feeds. The Ning documentation offers snippets of code allowing for easy integration of these services. For SNOUT, it was essential to include an instance of Google Maps on the site in order to map the environmental sensor information detected by the carnival characters during the SNOUT event. As there is no option to add Google Maps to a site using Ning's site templates, the SNOUT site was built from scratch using Ning's 'developer' option. This allows complete control over the code while maintaining access to Ning services such as site back-ups, access to social networking lists etc.

Figure 5.3 shows the website.

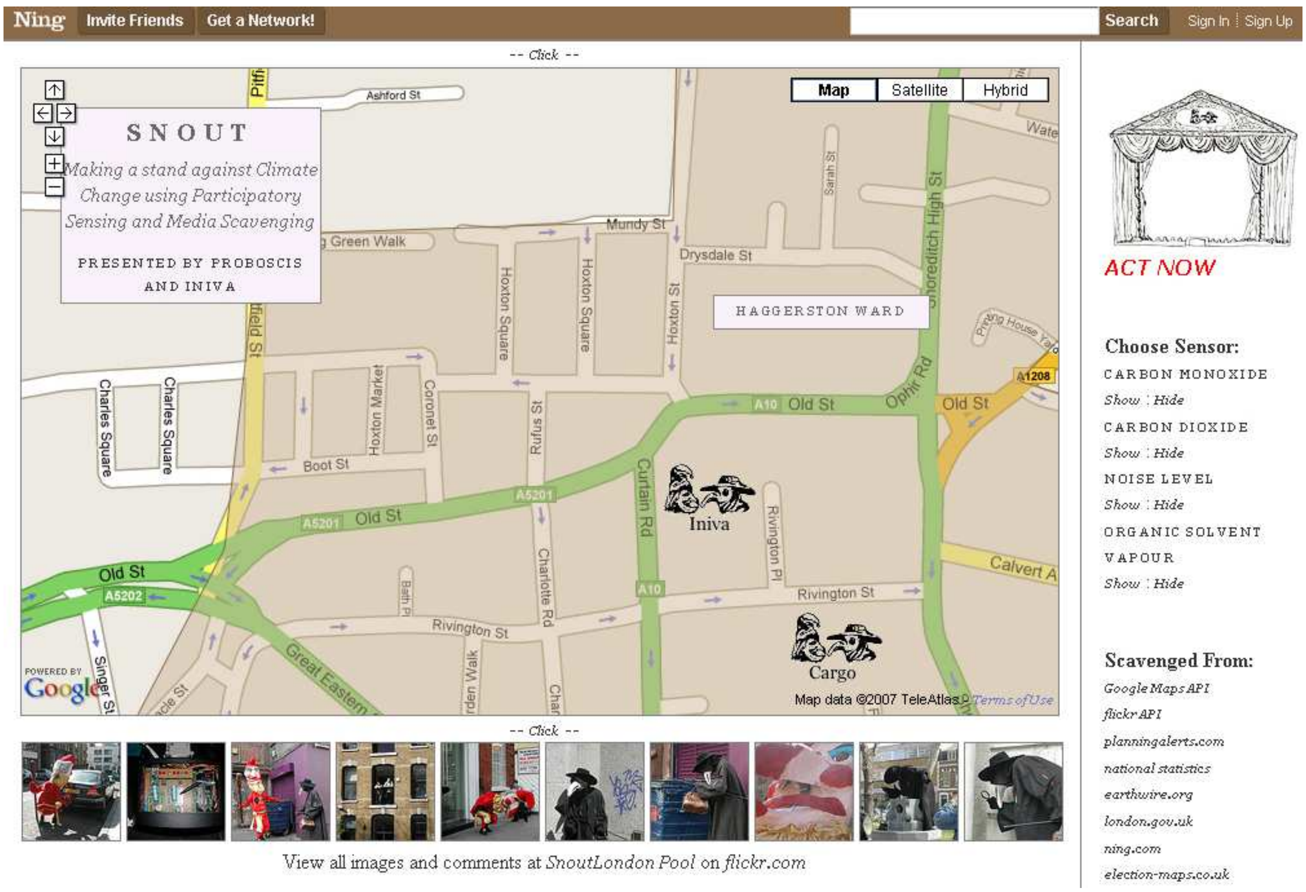

Figure 5.3: Snoutlondon website

5.3.3 Flickr

Flickr is an online photo management and sharing application that aims to help people make their photos available to the people who matter to them, and to enable new ways of organizing and searching photos. For the SNOUT website we created a Flickr group called SnoutLondon where images of the construction of the carnival costumes as well as pictures taken during the walk could be posted. Ning's PHP API means just a few lines of code needed to be added to the SNOUT website to create a connection to Flickr. On loading, the SNOUT site searches Flickr for images tagged with the word 'snoutLondon' and displays the ten most recent photos with this tag. Customisation of layout and search terms (username, tag and so on) is done by modification of the PHP code.

Figure 5.4 shows the images from *flickr*.

5.3.4 RSS

RSS (Really Simple Syndication) is a format used to publish frequently updated digital content. An RSS feed provides links to content on external websites. As with the Flickr web service, Ning PHP API provides an easy way to pull external RSS feeds into your website through the addition of just a few lines of code. Modifying this code alters the way the feeds are displayed on the page. RSS feeds on the SNOUT site are provided by two websites, earthwire.org and london.gov.uk. Using RSS feeds allowed us to offer two different perspectives on environmental issues (one local, one global) in a news-like format.

Figure 5.4 shows the RSS feed.

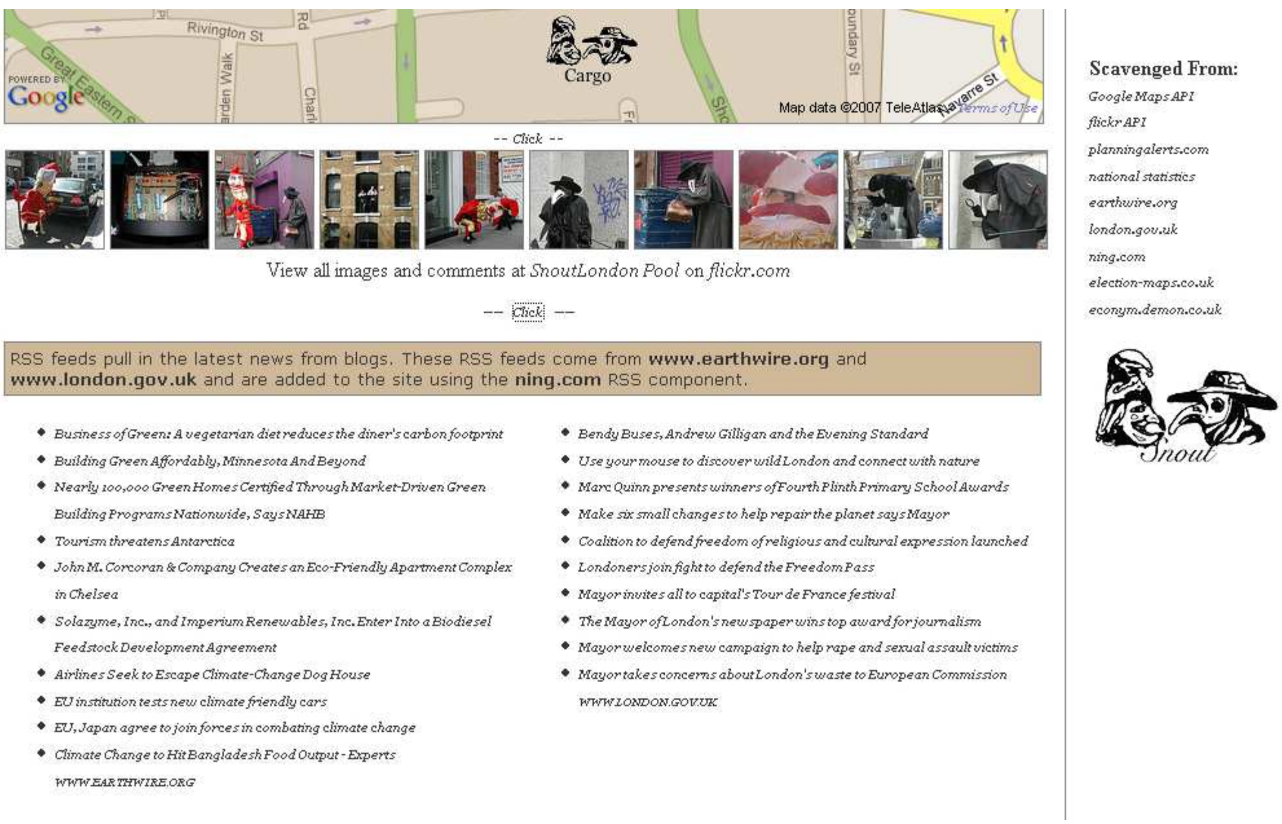

Figure 5.4: Flickr photos and RSS feed.

5.3.5 Google Maps API

An essential element of the SNOUT website is the display of pollution information collected by the environmental sensors during the SNOUT walk. This information is encoded as XML with categories for sensor type (carbon monoxide, carbon dioxide, noise or organic solvent vapour), location (lat/long coordinates) and sensor value (scaled to 1 – 10). SNOUT uses the Google Maps API GDownloadUrl method to access the XML file and JavaScript to extract the sensor information and display each sensor reading as a separate Google Maps icon located on the map. Information on the boundaries for Haggerston ward where the SNOUT event was to take place were found on the website electionmaps.co.uk. A semi-opaque polygon covering these boundaries was drawn onto the map using Google Maps GPolygon method. Further information relating to environmental issues was found on the National Statistics website. An extension to the GPolygon method by Mike Williams at econym.demon.co.uk provided a way to detect the location of a mouse click in relation to a polygon (whether the click was inside or outside the polygon). This meant the information from National Statistics could be accessed by clicking inside the Haggerston Ward polygonal area.

Figure 5.5 shows the Haggerston Ward polygonal area.

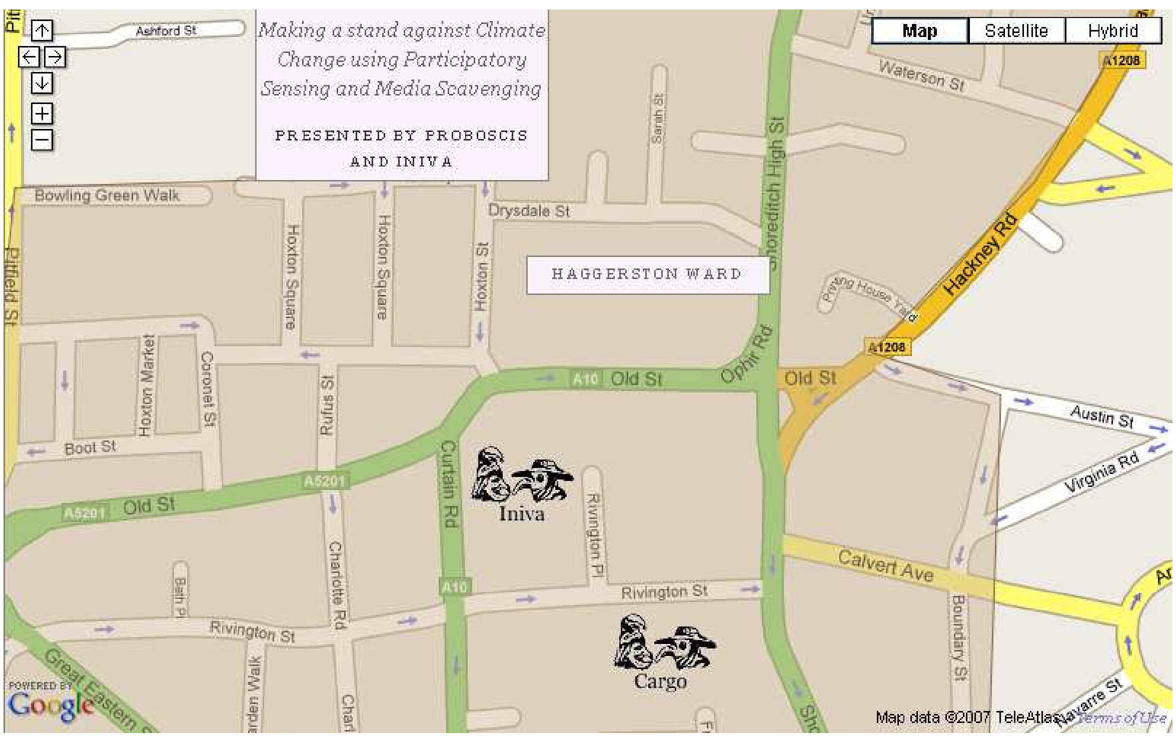

Figure 5.5: Google map.

5.3.6 Call to Action

The call to action section at the bottom of the page is a collection of links to various external community and governmental web sites with an interest in the issue of environmental pollution. Again, spanning local and national agencies the links help people use the web to make a stand on air quality in their locality by, for example, signing up to, or initiating governmental petitions, finding local organic food co-ops, or accessing discussion forums.

To support the second aim of the SNOUT site – to illustrate the process of creating a scavenged site – information about the construction of the SNOUT website can be found by clicking on the links above each section of the page. On clicking, a DHTML div reveals a brief description of the relevant service.

Figure 5.6 illustrates this page.

Figure 5.6: Snoutlondon call for action.

CHAPTER 6

Snout Costume Creation

6.1 Requirements

The Snout costumes needed to look arresting, be wearable and at the same time incorporate all the electronics. The masks had to be light and strong. The clothes needed to be strong enough to have the sensors and cables sewn in without the costumes sagging. Most of the costumes for sale we researched are made in lightweight nylon so we decided to make our own costumes instead based on easily available patterns. We chose simple patterns that could be adapted and made to fit a range of sizes as well as incorporate the electronics. The mask and head needed to have enough room for the sensors to be concealed but have a free air flow around them, and the sensors had to be sealed off from the performer's breath.

6.2 Plague Doctor

6.2.1 How it fits together

The performer wears a belt that holds the computer, battery pack and GPS unit. A cable attaches that to the costume. This connection was made easy to pull apart to remove the costume.

6.2.2 Plague Doctor Mask

The mask was purchased from a supplier of Venetian masks and adapted: the sensors were installed in the end of the beak with a cable running back up to the side of the face and down underneath the cloak to the computer: the sensor unit is attached to the mask with velcro: a plastic seal was fitted into the beak to close off the end of the beak from the performer's breath (it was made from a plastic bag and some adhesive velcro): new airholes were drilled above the seal: a stronger elastic headstrap was attached to the mask.

Figure 6.1: The Snout

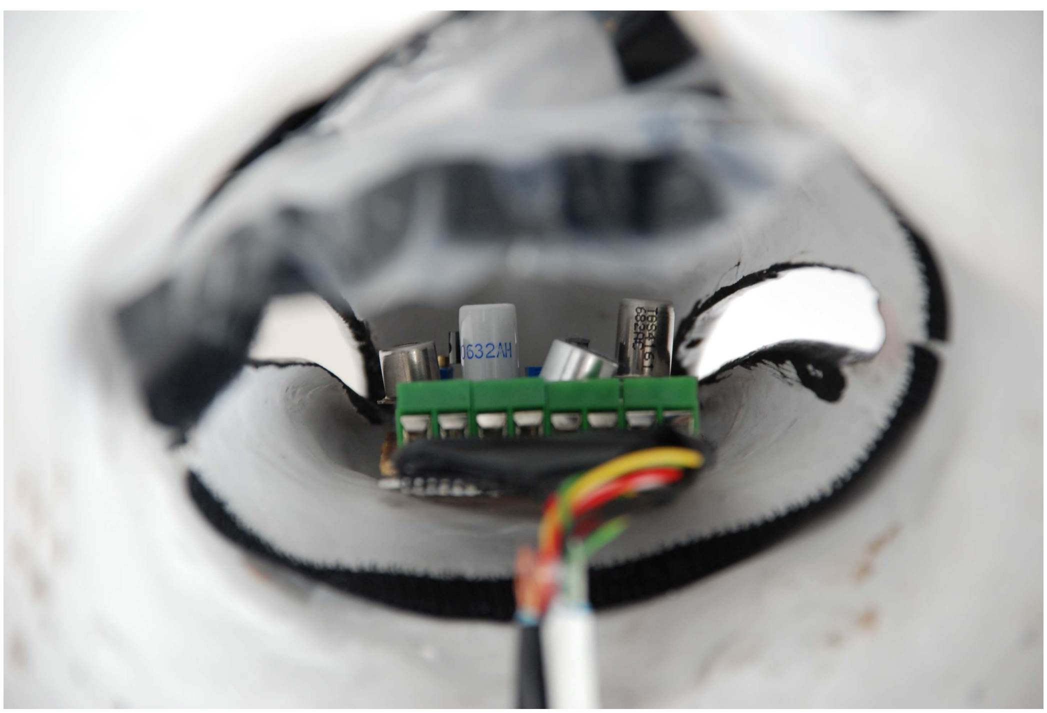

Figure 6.2: Mask Interior

6.2.3 Plague Doctor Cloak

Simplicity pattern no. 5840 (Misses', Mens and Teens robe and tunic) in black cotton twill (an inexpensive canvas would also work). We made cloak pattern A, omitting the capelet, taking the point off the end of the hood and adding velcro fasteners.

Cabling was held in place with hand stitched wide bias binding. Starting from the back (where the cable connects from the computer, worn on a belt, to the costume) the cable runs up the back then to each of the four LEDs. To support the LEDs, millinery mesh was glued onto the inside of the costume with PVA and a slot was cut through the cloth. The LEDs are pushed through this and a backing of mesh was placed over the back of the unit, using velcro, to keep it in place.

The Plague Doctor's props include a fan, a Gladstone bag (old fashioned doctor's case) and a water spray.

6.3 Mr Punch

How it fits together:
The performer wears a backpack that supports the large head and holds the computer, battery pack and GPS unit. A cable attaches that to the jacket. This connection was made easy to pull apart for the performer to put on and put off the costume.

6.3.1 Mr Punch Head

The head was worked out form a plasticine model as shown in Figure 6.7,

then the large head was made of paper mache on a form of fine chicken wire, as shown in Figure 6.8.

Space was made for the sensors to be attached under the nose, using velcro so they could be removed. PVA was mixed to the glue to harden it and a layer of muslin soaked in paste laid over the surface and pushed to shape wrinkles and skin texture. It was painted using acrylic paints and the edges of the wire were padded with interfacing. The head was supported above the wearer, who looks out of the neck ruff, on a pole attached to an adapted backpack. We followed instructions on making a backpack support and attaching the head to it from the The Puppeteers' Cooperative website and there are other resources for making giant puppets online.

Mr Punch's head also has a ponytail made from knitting wool cut into lengths laid like a curtain onto a strip on masking tape then sewn to a strip of interfacing and glued to the head.

Figure 6.3: Cloak

Figure 6.4: Cloak Inside

6.3.2 Mr Punch Clothes

Simplicity pattern no. 9800 (Adults' clown costume and hat) in red velvet and yellow nylon satin (otherwise red canvas or another material heavy enough to sew the cabling into the inside). We made pattern 1 with the following adaptions: shortened the trousers to fit below the knee; omitted the ruff; omitted the pom poms; left the front open and put press studs to convert it into a jacket; opened up the centre back seam to insert a tailored hump and padding; gathered the base of the jacket, added a frill around the bottom and a yellow satin belt; added various bits of braid and embroidery; glued black diamonds to the chest and sleeves in the places the LEDs would be visible.

6.3.3 Neck Ruff

Made from one metre of inexpensive nylon satin and and organza (so the performer can see out) the ruff is a gathered strip of cloth with a frill at each end. The frills are stiffened with wire passed through a narrow hem. The ruff is held in place on the bottom of the head and the neck of the jacket with velcro.

Cabling is held in place with hand stitched wide bias binding. Starting from the back (where the cable

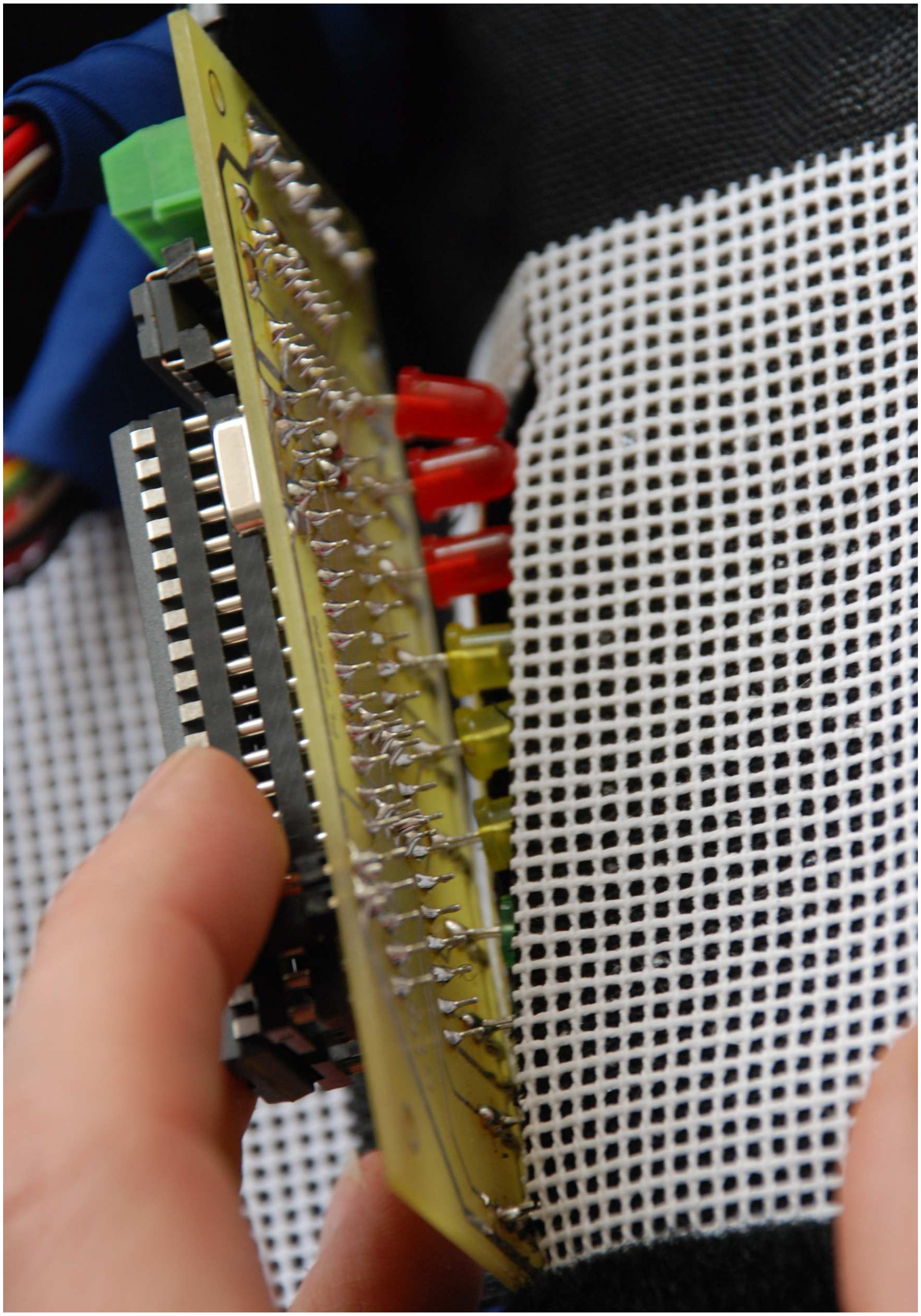

Figure 6.5: Cloak LED

connects from the computer, worn on a belt, to the costume) the cable runs up the back then to each of the four LEDs. To support the LEDs, millinery mesh was glued onto the inside of the costume (as

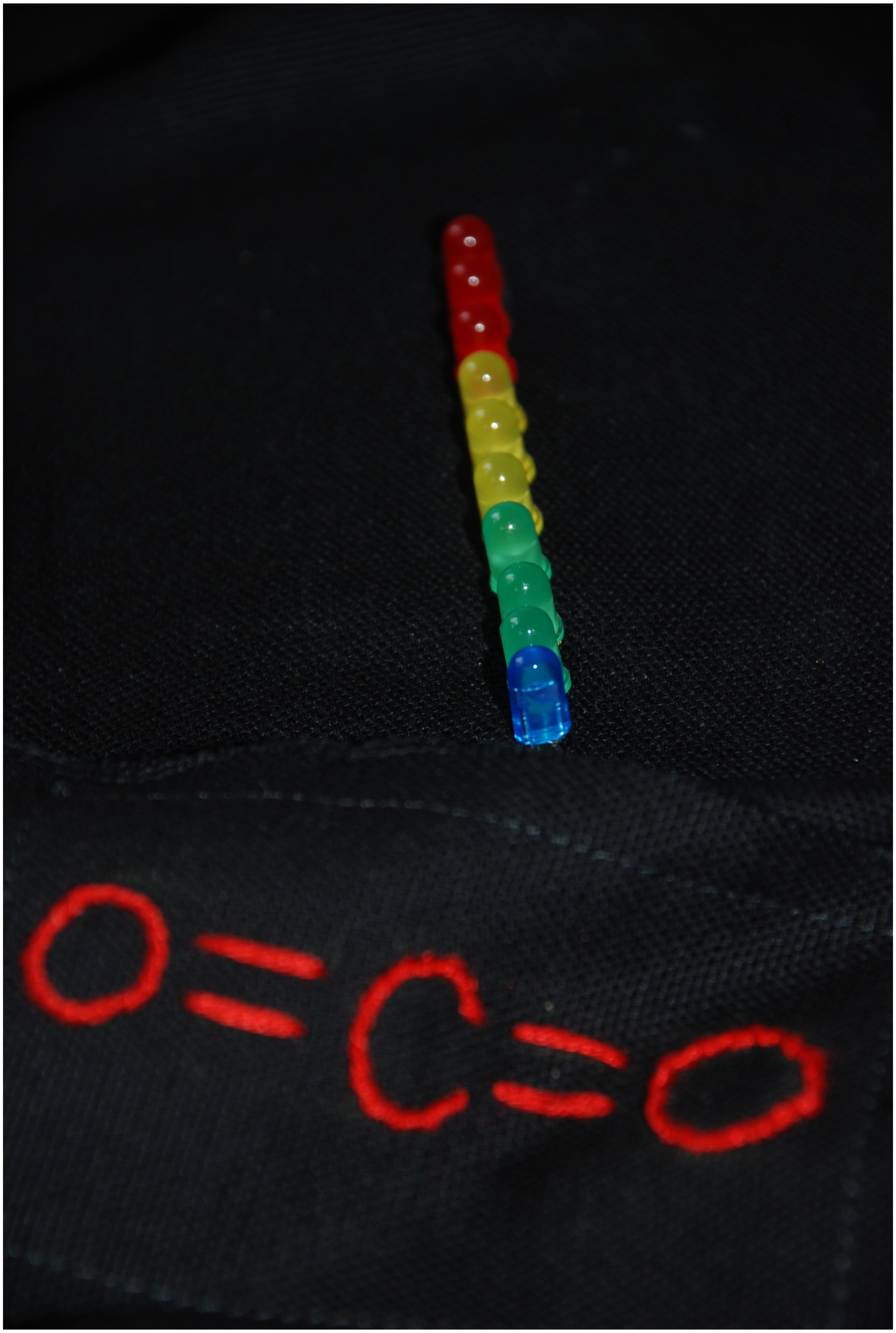

Figure 6.6: Cloak Embroidery

Figure 6.7: Mr. Punch Model

above).

The symbols for each sensor are hand embroidered with a raised stitch and stitched onto the costume under each LED.

Mr Punch's props include sausages (tights stuffed with bubble wrap) and a slapstick (cardboard tube, bubble wrap, masking tape and paint).

Figure 6.9: Mr. Punch's painted face

Figure 6.10: Mr. Punch Jacket

Figure 6.11: Mr. Punch's Neck Ruff

Figure 6.12: Mr. Punch's Jacket inside view

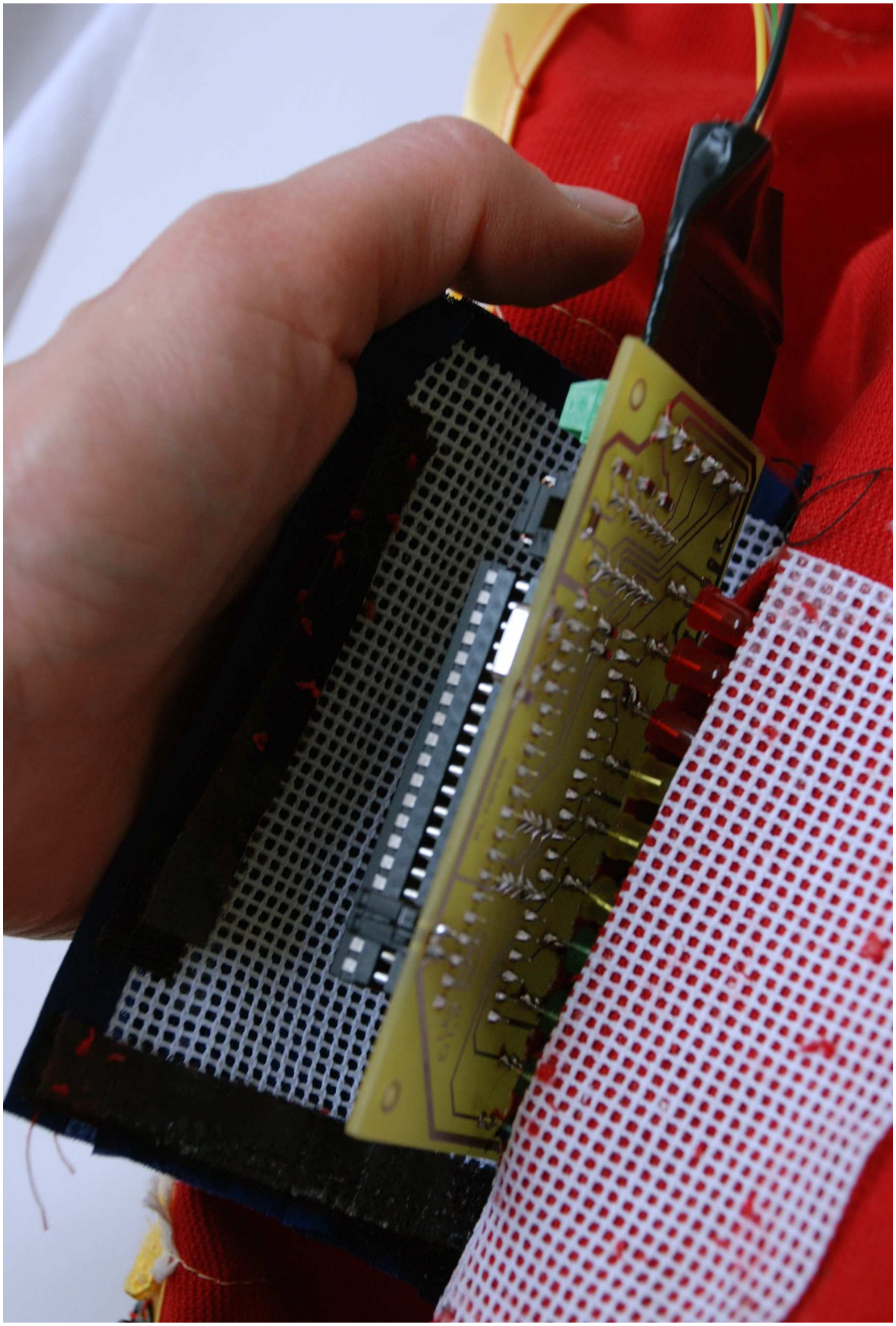

Figure 6.13: Mr. Punch's LEDs

Figure 6.14: Mr. Punch's Embroidery

Chapter 7

Evaluation

7.1 Schedule

Due to the late delivery of the hardware components the majority milestones of the project were pushed back. Hence the initial schedule was revised to reflect these changes upon the delivery of the hardware. The following diagrams show the initial and later modified timetable.

7.1.1 Initial Timetable

ID	Task Name	Duration	Finish
1	Research	51 days	Thu 01/02/07
2	Proposed Requirements	12 days	Fri 08/12/06
3	Finalized Requirements	63 days	Fri 23/02/07
4	Hardware construction	23 days	Wed 31/01/07
5	Hardware Testing	10 days	Fri 02/02/07
6	Software Design	35 days	Fri 02/03/07
7	Software Implementation	37 days	Fri 23/03/07
8	Prototype Testing	1 day	Tue 20/02/07
9	Final Testing	5 days	Fri 23/03/07
10	Evaluation	1 day	Mon 26/03/07
11	Project Report	45 days	Fri 01/06/07

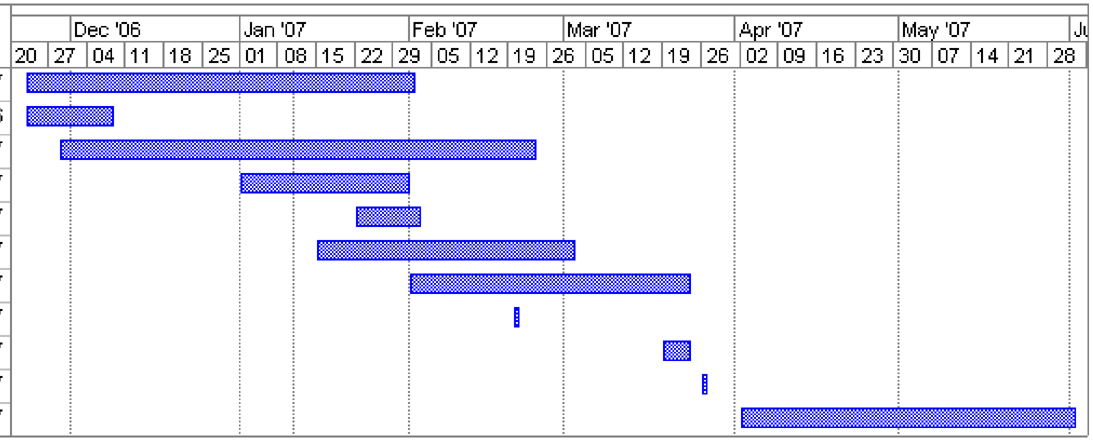

Figure 7.1: Initial schedule.

7.1.2 Updated timetable

ID		Task Name	Duration	Start	Finish
1		Research	51 days	Thu 23/11/06	Thu 01/02/07
2		Proposed Requirements	12 days	Thu 23/11/06	Fri 08/12/06
3		Finalized Requirements	63 days	Thu 21/12/06	Mon 19/03/07
4		Hardware construction	23 days	Mon 26/02/07	Wed 28/03/07
5		Hardware Testing	10 days	Mon 12/03/07	Fri 23/03/07
6		Software Design	20 days	Mon 12/03/07	Fri 06/04/07
7		Software Implementation	25 days	Mon 12/03/07	Fri 13/04/07
8		Prototype Testing	1 day	Wed 04/04/07	Wed 04/04/07
9		Final Testing / Demo	1 day	Tue 10/04/07	Tue 10/04/07
10		Evaluation	1 day	Wed 11/04/07	Wed 11/04/07
11		Project Report	45 days	Mon 16/04/07	Fri 15/06/07

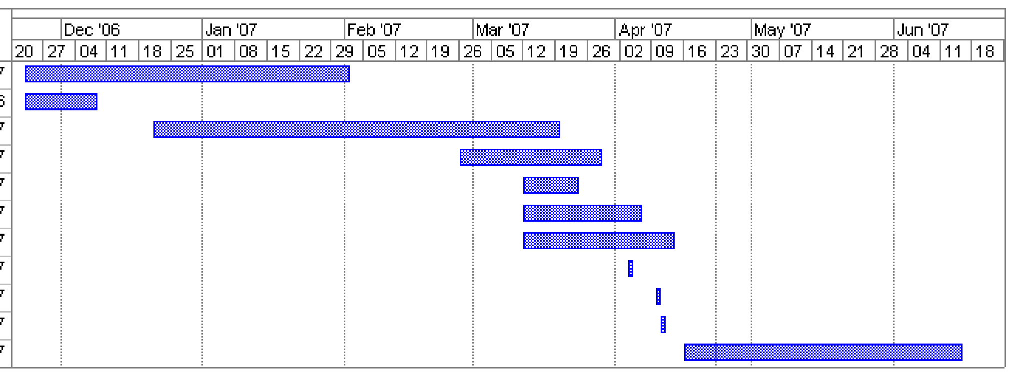

Figure 7.2: Updated schedule.

7.2 Demonstration

On $10^{th} April 2007$ the Snout platform was tested as part of an event organised by Proboscis[35]. Two performers dressed up in carnival costumes and carried the Snout platform on the path outlined in figure 5.2 to collect data. The low power consumption of the platform was impressive in comparison to *Feral Robots*[36]. The gathered data was later mapped for the visualisation of sensed pollution. Figure 5.2 shows the carbon monoxide readings from this trial.

The performance was followed by presentations outlining the wider context of the Snout project as well as displaying the results of the trial to members of the public.

7.3 Platform Extension

The Snout platform provides ease of extendability via its numerous IO capabilities. The Snout platform also includes a USB host module. Primarily the aim of this module was to allow data logs to be saved onto a USB flash memory as well as to the permanent storage of Gumstix. This would allow the users to plug in a USB flash memory and log sensor readings while on the move and later import this file into a mapping application to create the desired visualisation. The USB host allows for ease of integration of other external devices with this platform. So it was decided to use the Snout platform together with an RFID reader for monitoring and tracking of cleaning practises in an environment(aimed at hospital environments). The RFID reader was attached to the USB host of the Snout platform and successfully functioned when reading RFID tags. The application would detect tag ids and send these id strings via Bluetooth to a desktop computer. The received ids would then be used to generate a visual track of the travelled route.

To demonstrate this application, a set of RFID tags were taped to the lab floor and the Snout platform was mounted onto a vacuum cleaner with its RFID reader placed at the bottom of the vacuum cleaner. Once everything was set and running, a biased hoovering process began which resulted in the sequential reading of tags placed in the lab. Figure 7.3 shows the results of this trial.

The extended Snout platform was later mounted on top of an *iRobot Roomba[37]*(a robot vacuum cleaner) and put to the test. The tags Roomba read did not truly represent the path it had taken. This was mainly due to the small range of the RFID reader. Ideally the RFID reader should detect all tags that are withing the suction area of the vacuum cleaner. At the time of writing this we have obtained a suitable size antenna for the RFID reader and plan to rerun the tests soon.

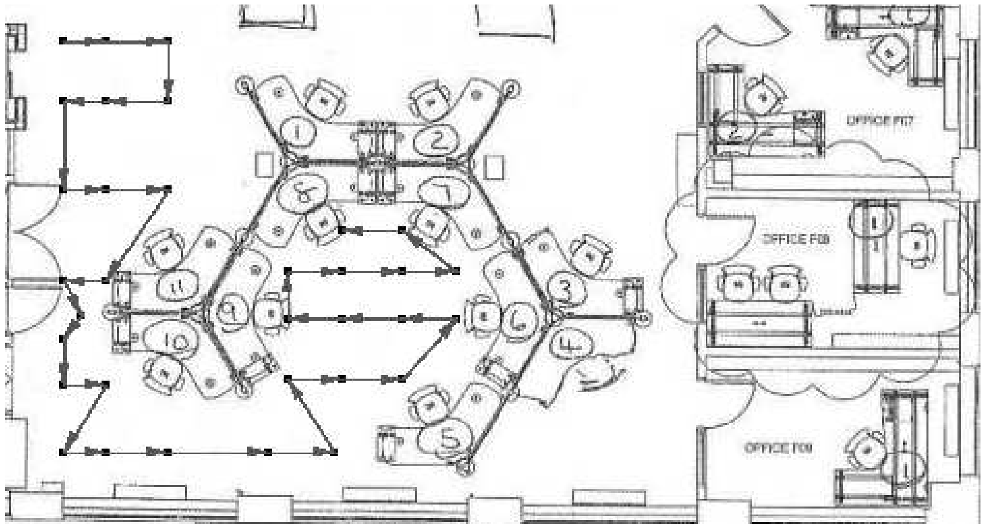

Figure 7.3: Sequential reading of RFID tags taped to the lab floor.

7.4 Future Work

7.4.1 Snout Platform

The Snout application platform can be improved in a number of ways. A number of of these future improvements are mentioned here;

- A useful addition to the current approach is to transmit collected sensor readings via Bluetooth to a hand-held device such as a PDA. This would allow multiple users to get a fine grain view of the physical environment in real time.
- Conversion of sensor readings into standard values as opposed to the current method of scaling them between one and ten. This would allow the acquired data values to be compared against any other data set.
- Faster sampling rate of the sensors in particular for the acoustic sensor in order to provide a more real time feedback to the user, which enhances the user interaction aspect of the application.
- Reducing power consumption of the device by switching off the sensor board in between sampling cycles.

7.4.2 Future Research Directions

Fure work consists of enabling the mobile device to interact with mobile phones. So users can ask for a particular sensor reading via the bluetooth on their mobile phone and receive a reading promptly.

Also enabling the device to communicate with other sensing devices can richen its sensor readings and give a comparison measure in real time. For example if a static sensor which is known for having accurate readings is placed somewhere in the environment, the mobile device to communicate with such a sensor to re-calibrate its own sensors. That aside, users would be able to see two measurements by two different devices in the same environement which would give users more confidence in the sensor readings presented to them.

References

[1] Snout project, March 2007. URL http://socialtapestries.net/snout.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. 2006.

[3] Habitat monitoring on great duck island. URL http://www.greatduckisland.net. http://www.greatduckisland.net.

[4] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-589-0. doi: http://doi.acm.org/10.1145/570738.570751.

[5] Tatiana Bokareva, Wen Hu, Salil Kanhere, Branko Ristic, Neil Gordon, Travis Bessell, Mark Rutten, and Sanjay Jha. Wireless sensor networks for battlefield surveillance. October 2006. URL http://www.cse.unsw.edu.au/~tbokareva/papers/lwc.html.

[6] Sinem Coleri Ergen, Sing Yiu Cheung, Pravin Varaiya, Robert Kavaler, and Amine Haoui. Wireless sensor networks for traffic monitoring. April 2005. The Fourth International Conference on Information Processing in Sensor Networks (IPSN 2005 April 25-27).

[7] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/332833.332838.

[8] Christopher M. Sadler and Margaret Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 265–278, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-343-3. doi: http://doi.acm.org/10.1145/1182807.1182834.

[9] John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. In

SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles, pages 146–159, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-389-8. doi: http://doi.acm.org/10.1145/502034.502049.

[10] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575–578, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1588-6.

[11] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks, 2002. URL `citeseer.ist.psu.edu/shah02energy.html`.

[12] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. January 2003. URL `citeseer.ist.psu.edu/shah03data.html`. Proc. IEEE Wksp. Sensor Network Protocols and Apps., January 2003.

[13] Jae-Young Choi, Jun-Hui Lee, and Yeong-Jee Chung. Minimal hop count path routing algorithm for mobile sensor networks. In *IMSCCS '06: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences - Volume 2 (IMSCCS'06)*, pages 616–621, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2581-4. doi: http://dx.doi.org/10.1109/IMSCCS.2006.241.

[14] Do-Seong Kim and Yeong-Jee Chung. Self-organization routing protocol supporting mobile nodes for wireless sensor network. In *IMSCCS '06: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences - Volume 2 (IMSCCS'06)*, volume 2, pages 622–626, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2581-4. doi: http://dx.doi.org/10.1109/IMSCCS.2006.265.

[15] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0493-0.

[16] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-868-7. doi: http://doi.acm.org/10.1145/1023720.1023726.

[17] Pubudu N. Pathirana. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing*, 4(3):285–296, 2005. ISSN 1536-1233. doi: http://dx.doi.org/10.1109/TMC.2005.43. Member-Nirupama Bulusu and Senior Member-Andrey V. Savkin and Member-Sanjay Jha.

[18] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem. In *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, June 2002. URL `citeseer.ist.psu.edu/howard02mobile.html`.

[19] Andrew Howard, Maja J. Matari'c, and Gaurav S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Auton. Robots*, 13(2):113–126, September 2002. ISSN 0929-5593. doi: http://dx.doi.org/10.1023/A:1019625207705.

[20] Guiling Grace Wang, Guohong Cao, and Thomas F. La Porta. A bidding protocol for deploying mobile sensors. In *ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols*, pages 315–324, Washington, DC, USA, November 2003. IEEE Computer Society. ISBN 0-7695-2024-3.

[21] Guiling Grace Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. Technical report, 2003. URL `citeseer.ist.psu.edu/663411.html`.

[22] Guiling Grace Wang, Guohong Cao, Thomas F. La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks, March 2005.

[23] Yongguo Mei, Changjiu Xian, Saumitra Das, Y. Charlie Hu, and Yung-Hsiang Lu. Replacing failed sensor nodes by mobile robots. In *ICDCSW '06: Proceedings of the 26th IEEE International Conference Workshops on Distributed Computing Systems*, page 87, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2541-5. doi: http://dx.doi.org/10.1109/ICDCSW.2006.90.

[24] Peter Corke, Ron Peterson, and Daniela Rus. Communication-assisted localization and navigation for networked robots, April 2004. URL `citeseer.ist.psu.edu/corke04communicationassisted.html`.

[25] Matthai Philipose, Ken Fishkin, Dieter Fox, Wolfram Burgard, and Dirk Hahnel. Mapping and localization with rfid technology. Technical report, Intel Research, December 2003. URL `citeseer.ist.psu.edu/philipose03mapping.html`.

[26] C. Chang. Improving hallway navigation in mobile robots with sensor habituation. In *IJCNN'00: In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-*

Volume 5, volume 5, page 5143, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0619-4.

[27] Yoon-Gu Kim, Han-Kil Kim, Suk-Gyu Lee, and Ki-Dong Lee. Ubiquitous home security robot based on sensor network. In *IAT '06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 700–704, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2748-5. doi: http://dx.doi.org/10.1109/IAT.2006.128.

[28] Gabriel T. Sibley, Mohammad H. Rahimi, and Gaurav S. Sukhatme. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. volume 2, pages 1143–1148, Washington, DC, USA, May 2002. IEEE Computer Society. URL `citeseer.ist.psu.edu/article/sibley02robomote.html`.

[29] Gumstix, inc. URL `http://www.gumstix.com`. Gumstix, inc. is a leading manufacturer of "Full Function Miniature Computers" (FFMC) boards.

[30] Expansys - socket oem bluetooth gps receiver. URL `http://www.expansys.com/p.aspx?i=119717`.

[31] Figaro an iso9001 and 14001 company. URL `http://www.figarosensor.com/`.

[32] Cyrod technologies inc. URL `http://www.cyrod.com`.

[33] Vinculum vdip module. URL `http://www.vinculum.com/prd_vdip1.html`.

[34] Ftdi - future technology devices international ltd. URL `http://www.ftdichip.com`.

[35] Proboscis. URL `http://proboscis.org.uk`.

[36] Dima Diall and Demetrios Airantzis. *Urban Tapestries - Feral Robot Client Architecture*. Proboscis and Birkbeck College, London, United Kingdom, 1.1 edition, March 2006. URL `socialtapestries.net/feralrobots/`. Technical report of a Feral Robot prototype, 2nd generation.

[37] irobot roomba vacuum cleaner. URL `http://www.irobot.com`.