Examensarbete

HÖGSKOLAN I SKÖVDE

ENERGY-EFFICIENT SENSOR MANAGEMENT

How dynamic sensor management affects energy consumption in battery-powered mobile sensor devices.

ENERGIEFFEKTIV SENSORHANTERING

Hur dynamisk sensorhantering påverkar energikonsumtionen i batteridrivna mobila sensorenheter.

Examensarbete inom huvudområdet Datavetenskap
Grundnivå 30 högskolepoäng
Vårtermin 2012

Marcus Johansson

Handledare: Gunnar Mathiason
Examinator: Anders Dahlbom

Abstract

This thesis has investigated how the energy consumption can be reduced in a mobile sensor unit by using a dynamic measurement scheme. This was done by developing a scheme based on inspiration from existing works in related areas and on techniques found in literature. The developed scheme was then implemented on a mobile sensor unit and tests were conducted where the energy consumed by the scheme was measured. This was compared to a static baseline approach in order to evaluate the efficiency of the scheme. The results showed that on the platform used in this thesis the developed scheme can reduce the energy consumption in a typical scenario by 4.7% or 6.7% depending on which sensors are used. A conclusion drawn is that the platform has a major impact on how effective the scheme can be.

Keywords: energy efficiency, sensor management, context-aware

Table of Contents

1 Introduction

The development of cheaper and more powerful mobile devices has brought mobile computing to a more widespread use. This development has also resulted in cheaper and more competent mobile sensors. There is a wide variety of areas where mobile sensor could be applied. The Swedish Defense Materiel Administration (FMV), among other organizations, is interested in investigating the possible usage of mobile sensors.

When non-trivial sensors are used a sensor management system is needed. Sensor management controls the sensors, collects the data reported by the sensors and handles the data. The purpose of a sensor manager is to ensure that all parts of a sensor system work together to accomplish their purpose.

To keep a device mobile batteries are often used as the power source. The lifetime of the device is therefore determined by how long the battery lasts. To prolong the device's lifetime, several approaches have been developed to minimize the energy consumption. Different techniques have emerged on a hardware level to reduce the power consumption. To fully utilize these, software techniques such as Dynamic Power Management (DPM) have emerged.

With the increased use of mobile devices equipped with sensors, such as smartphones, context-aware systems have gained in interest. A context-aware system uses available sensors to determine the current context of the system in order for the system to adapt its behavior accordingly.

Mobile sensor systems typically use battery-powered mobile computer-units that are equipped with sensors and some means of communication. For the system to perform its task in a satisfactory way it is important to utilize the energy available in a way that prolongs the lifetime of the system without causing loss of important sensor-data. A sensor manager is needed to reduce the energy spent on sensors and communication when the context of the object being measured indicates that the sensor-data is not important.

This thesis investigates how a dynamic sensor management scheme affects energy consumption in a mobile sensor device.

To investigate this, a dynamic measurement scheme was developed based on inspiration from existing works in related areas and on techniques found in literature. Depending on the current context, the scheme determines how to control sensors and communication. The scheme was then implemented in a mobile sensor-unit. A number of tests were performed that show the energy consumption of the scheme’s various modes. From these tests the effect of the scheme was calculated.

The tests demonstrated that the scheme has a minor impact on the overall energy consumption. On the platform used in this thesis, the scheme reduces the energy consumption with 4.7% or 6.7% depending on which sensors are used.

2 Background

In today's society everyone has mobile computers. With smartphones and tablets mobile computing has reached new heights. The development towards smaller and more powerful computer units has made all this possible. This development has also brought about smaller and more competent mobile sensors. These sensors are often placed on mobile computer units whose computing capacity is increasing at the same time as the price is dropping. These mobile devices have opened up new possibilities in a wide array of areas.

2.1 Sensor Management

Today sensors can be found in many different situations. They are used in everything from monitoring and controlling military equipment to tracking zebras (Liu et al., 2004) or monitoring trained canines (Britt et al., 2011). For all these sensor systems to function as they are supposed to, sensor management is needed.

Sensor management is an area where much research has been done over the years. Sensor management is needed when sensors are non-trivial. When the sensors that are used have different modes of operation or if they are connected in ways where they are dependent on each other, a management system is needed to get the sensor to perform as intended.

Sensor management incorporates a lot of different aspects. Ng and Ng (2000) give a generic description when they state that sensor management means "to manage, co-ordinate and integrate the sensor usage to accomplish specific and often dynamic mission objectives". With this description Ng and Ng explain that sensor management is about controlling sensors, efficient use of sensors and combining multiple sensors to work together.

Sensor management can have several different roles and functions. Ng and Ng (2000) divide the functionality into three levels.

Level 1 – Lowest level of sensor management. It is about the individual control of each sensor. Frequency and direction are two examples of what can be controlled.

Level 2 – Medium level. Focus is on modes and tasks of the sensor. Sensor management on this level is about prioritizing tasks and determining which mode should be active.

Level 3 – Highest level. It covers how sensors work together to provide the best coverage or efficiently using the sensor at hand.

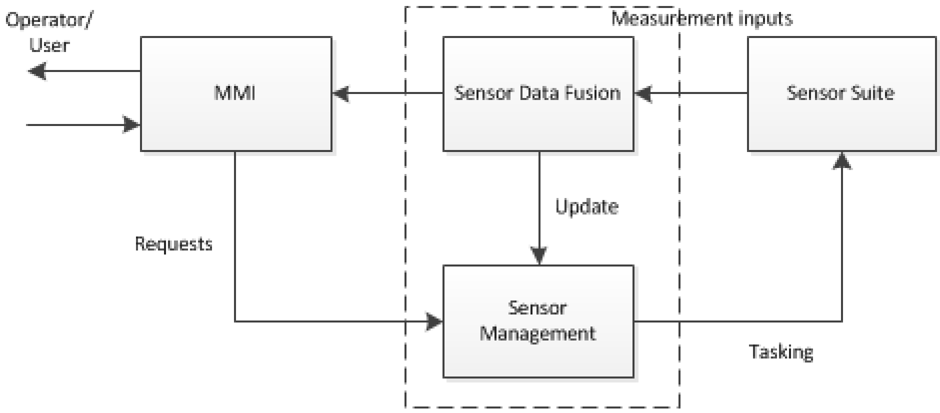

Figure 1 Basic model of sensor management. (Ng and Ng, 2000)

Ng and Ng (2000) present a basic model of a sensor management system. The model can be seen in figure 1.

The data from the sensors is collected by the data fusion unit where it is combined, processed and assessed. The data fusion unit usually has several layers to it. These layers correspond, to some extent, to level 1, 2 and 3 in the Joint Directors of Laboratories (JDL) model. A more in-depth description of the JDL model is presented in chapter 2.1.2. The data fusion unit provides data to the Man Machine Interface (MMI) and to the sensor management.

The sensor management receives requests through the MMI from the user. The sensor management then controls the sensors based on the updates from the data fusion and the request received from the MMI. The MMI handles all communication between the operator and the sensor management system.

2.1.1 Platform and architecture

When choosing a platform there are two options: either multiple sensors on a single platform or multiple sensors on multiple platforms. Both alternatives have their advantages and disadvantages (Buede, 1990).

Buede (1990) claims that the major reason for a single platform to handle several sensors is that the sensors provide different information. The information from the sensors is then combined to get a clearer view that includes all necessary aspects. Besides controlling the individual sensors, management is also needed to determine which sensors to use, how and for what purpose. An important aspect here is which role the operator has; is he or she controlling the sensors in real-time or providing priorities on a required basis?

Multiple sensors on multiple platforms have become very common in the form of sensor networks. Beude (1990) recognizes communication as the major problem for these systems. Communication is needed for everything from control of the sensor node to collecting sensor data. In sensor networks there is no central node that everyone is connected to; instead communication is achieved by forwarding data through the nodes that are in range. The control of the sensors and the fusion of the sensor data are important points for the sensor management of these systems.

When it comes to the architecture of sensor management there are three main design types (Xiong & Svensson, 2001).

- Centralized
  The centralized design has a central node or processor that performs all decisions, gives instructions and collects the data.
- Decentralized
  The decentralized design does not have one point where all decisions are made but lets the individual nodes and platforms make their own decisions. Communication is a major part of such a design as all co-ordination is achieved by sharing information.
- Hierarchical
  The hierarchical design is a mix between centralized and decentralized. Several levels exist in a hierarchy where the top level collects and fuses the data from all sensors.

Which design to use depends a lot on how the system's hardware is constructed. If we have a single platform a centralized design is the natural choice even if the hierarchical could be used as well. On multiple platforms it is more natural to choose a decentralized or a hierarchical design, although a centralized design could also be made to work well. This is illustrated in figure 2.

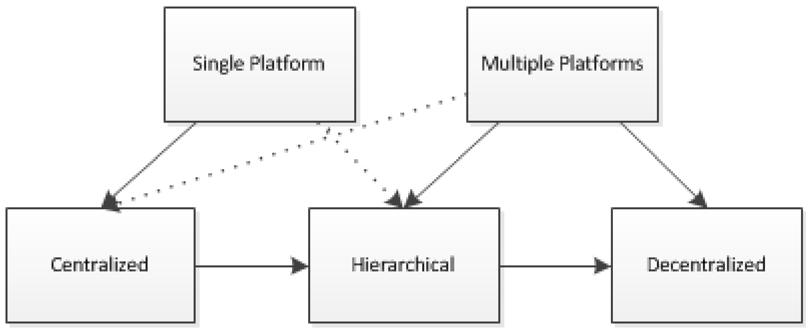

Figure 2 Architecture related to platform.

2.1.2 Sensor Data Fusion

An important part of sensor management in multi sensor systems is sensor data fusion. Sensor data fusion is the combination of data given by the sensors to provide a clearer and better view when communicating the results. Sensor fusion is a subset of information fusion. Several definitions of information fusion exist in literature. Boström et al. (2007) review a variety of these and propose a definition based on previous works.

> "**Information fusion** is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making."

A model commonly used to describe the different levels of information fusion is the so-called JDL model. The JDL model was developed as a conceptual model in order to unify terminology and improve communication among military researchers. Although the model originated in the military it can still be applied in other areas. The JDL model separates the different functions that can be found in information fusion. It defines five levels. (White, 1998; Xiong & Svensson, 2002)

> Level 0: Low-level data association and estimation. This level deals with raw input data and tries to detect and estimate how to read the data to acquire the most valuable information.

Level 1: Object refinement and assessment. This level uses the information gained in previous level and aims to detect individual entities or objects and determine their attributes.

Level 2: Situation refinement and assessment. Level 2 uses the information from earlier levels and assesses the relations between entities in order to provide a view of the current situation.

Level 3: Threat refinement or impact assessment. This level takes the information from earlier levels one step further and makes a prediction on the impact the current situation might have in the future. An assessment is made of the situation will affect the purpose of the system.

Level 4: Process refinement. The last level does not make any further estimations or assessments. The last level uses the information gained in previous levels to control the available resources in order to improve the whole fusion process.

To provide a view of the role sensor management fills in information fusion Xiong & Svensson (2002) relate the JDL model to concepts found in sensor management and provide examples of sensor management activities performed at the different levels. Level 0 is described as the refinement of the physical signals acquired from the sensors. Levels 1, 2 and 3 can sometimes be difficult to separate and do not need to be performed in any particular order. Levels 1, 2 and 3 deal with the processing of the sensor data to detect objects or entities, to place them in a situational context and to make an assessment of the significance the sensed data has on the system. Level 4 can be directly mapped to the sensor manager that uses the output of levels 0-3 to make decisions on how to control the system and the sensors in order to improve the process of information fusion and in the best way fulfill the purpose of the system.

2.2 Energy Consumption

A major problem with mobile devices is that they usually have batteries as their power source. This makes power consumption an important aspect as the battery capacity is what determines the lifetime of the device. There are several ways to deal with this problem; one of the most common solutions is to try to minimize the power consumption through dynamic power management.

2.2.1 Dynamic Power Management

Dynamic Power Management (DPM) is a set of techniques where the components in the system change performance based on the computational need at the moment. DPM is not a single technique but rather a collection of techniques to control the performance of the system, and thereby the power consumption, in various ways (Benini et al., 2000).

We can look at DPM either as a component centric approach with different techniques to control the specific component or we can look at it from a system-level. When looking at specific components it is usually the Central Processing Unit (CPU) that is in focus with a variety of common techniques for reducing the power consumption. At a system-level we can look at DPM as a technique that shuts down components or devices that are not being used at the moment and when they again are needed it wakes them up (Benini et al., 2000).

If we consider the CPU in terms of DPM we have a multitude of different techniques that can be used to reduce power consumption. Here two of these are summarized.

Dynamic Voltage and Frequency Scaling (DVFS) is a technique under DPM where the voltage and speed of the CPU can be changed dynamically. When the CPU load is low the speed and voltage can be reduced. This results in longer time for the computation to be completed but as the voltage is lower, less power is consumed. As the relationship between the voltage and power consumption is non-linear, less power is used when spreading out work compared to doing work in high speed bursts.

Most if not all modern CPU's have some kind of power management in the form of switching between different power states. The Texas Instrument OMAP 35xx series used on the Gumstix Overo computers that will be used in this thesis has a functionality called Dynamic Power Switching (DPS) (Musah & Dykstra, 2008). DPS makes the system switch between high-power mode and low-power mode during the system's active time. With DPS the system uses its resources to complete its task as fast as possible followed by automatically switching back to a low-power mode. DPS aims to maximize the time spent in idle without losing performance. This makes it suitable for use in a real-time system. In this thesis different levels of idle-modes of the CPU will be utilized to minimize the energy consumption.

2.2.2 Software Affect Energy Consumption

All these techniques are good and work mostly on a hardware and OS level. However, when designing software for these systems, care has to be taken to realize the full benefit of the techniques. Lorch and Smith (1998) writes about different levels of energy management and states that although some aspects of energy management are unsuitable on an application level and mostly are controlled by the OS there are strategies on the application level that will affect energy consumption. Tennies (2003) list two approaches to fully utilize the idle mode of an embedded system.

The first is intelligent waiting which is about entering idle mode as often as possible. Instead of using loops to poll a register to see if a flag is set an external interrupt could be tied to the flag and wake the CPU from idle-mode. If an external interrupt cannot be used a system timer can be used to periodically wake the CPU to see if the flag has been set. If no change is detected the CPU can return to idle mode.

The second is event reduction which is about keeping the CPU in idle as long as possible. By analyzing the events and actions that would wake the CPU from idle and avoiding unnecessary interrupts significant energy waste can be avoided. When data is received by a serial port, interrupts are usually generated with intervals of a few bytes. For example: we could have an interrupt interval of 8 bytes. If a data chunk of a few kilobytes were to be received, with a rate of about a hundred kilobits per second, a huge number of interrupts would be generated in a short time. Each interrupt would wake the CPU only to have it put back to idle. As these transitions have a small cost to them in the form of the time and computation it takes to make the transition, and as the OS wants to perform its tasks when the CPU is woken, energy is wasted.

2.3 Context-Aware

A context-aware system is able to sense the environment and adapt its operation according to the current context without needing explicit user interaction. Mobile devices have introduced new possibilities for context-awareness. Mobile device can be expected to change context more often as they are able to be brought to new locations and new contexts. With mobile devices, location is often a key when deciding context. Different locations might want to provide different functionality. What determines the context can vary depending on the purpose of the system. The required context information can be retrieved by things such as different kinds of sensors, network information, device status or current user activities (Baldauf & Dustdar, 2004).

A context model is used to define and store context in a form a machine can process. A context model consists of several context atoms (Baldauf & Dustdar, 2004). A context atom is described with a set of attributes. The two most important are context type and context value. Context type states what category the context belongs to. It could be something like temperature, position or speed. Context value refers to the raw sensor data. Other attributes that could be included if needed by the system are description, time stamp and source.

For example with temperature we could have three contexts; hot, normal and cold. They all have the same type but as the value of the temperature sensor changes, the context changes between hot, normal and cold. The context at a given moment could then be similar to what is shown in table 1.

Table 1 Examples of Context

Context type	Context Value	Context
Temperature	20 °C	Normal

Context can be derived from either single or multiple sensors. Single sensor systems usually focus on location as the context, but with multiple diverse sensors a more complete context can be obtained. Gellersen, Smith and Beigl (2002) designed a series of diverse systems incorporating multiple heterogeneous sensors to provide a clearer view of the context. They showed that using cheap and simple sensors and combining the information gained from these could provide a context that could not be gained from single sensor systems.

2.4 TENA

As this project is implemented in a military environment there are some circumstances that have a significant impact on the project. For this work the circumstance that carries the most weight is the means by which communication should be performed. FMV is trying out a new system for communication between computers and programs. This is an American system by the name of Test and Training Enabling Architecture (TENA). All computer communication in this project will be realized through TENA.

TENA is an advanced middleware developed by the United States military. TENA was developed to provide a common infrastructure for military test and training ranges and other facilities. The goals with this common infrastructure are to enable interoperability between facilities in a quick and cost-efficient manner, as well as to promote the reuse of different applications and assets.

TENA is used for testing and training in a distributed way that combines the use of real sensors with data from virtual and constructive systems. A system like this is called a Live, Virtual and Constructive (LVC) system. LVC refers to the combination of three types of distributed simulations and applications (Noseworthy, 2008).

- Live – Applications connected to real physical assets.
- Virtual – Simulations of physical assets such as airplanes and tanks to enable testing and training in a virtual environment.
- Constructive – Simulations of synthetic environments to enable testing and training with virtual models of physical assets.

To make all these different kinds of applications work together in an efficient manner it is important for them to be able to communicate with each other efficiently in real-time. As TENA is the center of these applications, TENA needs to fulfill significant requirements in regards to these aspects.

TENA supports several variants of communication (Noseworthy, 2008). The simplest of these is message passing where data packets are sent with an address that leads to the recipient application. Another form of communication is publish-subscribe where an application can publish its data while indicating what kind of data it is. Another application can then subscribe to that specific kind of data and will then receive the data of that kind that gets published. Distributed object-oriented computing is a more advanced way of communication. It tries to make method invocation on remote objects to seem like method invocation on local objects.

TENA is cross-platform and available for all major operating systems including Windows, Linux, OSX, IOS and Android. TENA is owned by the US government and available for registered users at their webpage: http://www.tena-sda.org.

3 Problem definition

A typical problem for organizations that take measurements on different kinds of equipment is for the measurement-device to be non-intrusive. Non-intrusive means that the device should be deployed and perform its measurements with a minimal influence on the environment and the equipment being measured. One aspect of this is that the device should be small enough that its presence does not affect the test results. A smaller size could also enable a quicker deployment. One consequence of the size limitations of the device is that the capacity of the batteries will be limited and by that the amount of energy we can store in them. It is therefore a necessity for the device to conserve the available energy so that the lifetime of the device is sufficient for the objective to be completed.

With mobile sensors power-saving idle-modes are important features as the purpose is to monitor the sensors' environment. This often involves measuring changes over relatively long periods of time. An idle mode where the unit can sleep is therefore an important part in prolonging the lifetime of the sensor-units.

The duty-cycle of a sensor consists of sleep, measurement and communication. A duty cycle is measured as the ratio of active time to the total time. A sensor typically has a low duty cycle which means that the share of time the sensor is active is low and most of the time is spent in an inactive sleep state. As the unit's energy consumption is minimized during sleep, what remains to work with is measurement and communication. Both actions are major consumers of energy in the sensor unit. As such we want to minimize the usage of these while still keeping the sensor-unit's data relevant and accurate.

One method to do this is to implement a dynamic measurement scheme. The aim of such a scheme is to evaluate the sensor-unit's state from the sensor feedback and act differently depending on the context of the measured object. If the context of the object is of a kind where the measurements would not be of interest, there is little need to waste energy on rapidly conducting measurements and communicating these to the listeners. If instead we let the sensor-unit idle for longer periods when the object's context is not interesting, or likely to provide important sensor data, we will save energy for later. A dynamic measurement scheme would keep a longer interval between updates and communications when the circumstances indicate that the data is not likely to change. When the sensor data indicates that something interesting is happening it changes mode and performs more measurements and communicates more often to keep the accuracy of the data at a sufficient level.

This thesis proposes to use a dynamic measurement scheme to reduce battery usage and thereby extend the lifetime of the measurement system. The following question will be answered in this thesis:

> *How much can the energy consumption be reduced in a mobile sensor unit by using a mode-sensitive and dynamic measurement scheme?*

The measurement system will be applied and evaluated in a system for monitoring conditions in military vehicles. The system will be implemented on a predetermined platform that has been chosen with respect to the availability of TENA and the possibility to integrate with the military vehicles. The use of the platform will generate an answer to the question that is somewhat restricted the particular platform and not completely general.

An answer to the question will be reached by completing the following objectives:

1. Investigate existing schemes and approaches available in literature.
2. Based on the discovered schemes, develop and implement an improved scheme for mode-sensitive sensor management in order to reduce energy consumption.
3. Evaluate the scheme and compare it with a static baseline approach in terms of consumed energy and measurement accuracy.
4. Describe the results in terms of battery and system lifetime during a typical execution plan in the setting.

By following these objectives the thesis will have a base in existing works from related areas. A scheme will be generated that is designed for the platform and that utilizes available techniques in the best way. This scheme will be evaluated and by comparing it with a baseline approach the effect the scheme has on the energy consumption can be expressed. This will lead to an answer to the question.

3.1 Motivation

At the Swedish Defense Materiel Administration (FMV) Test Range in Karlsborg a huge variety of tests are performed to evaluate the properties of materiel aimed for military use. The tests are made with measurement equipment and sensors that are large and wired. This leads to a lot of work moving equipment around and setting it up for use. FMV is interested in investigating the possibilities with new mobile technology. By adopting mobile technology, FMV wants to move away from some of the older systems. Among other things, FMV hopes to gain mobility, ease of use and the ability to follow up the tests in real-time. As a step towards adopting these new systems FMV wants to investigate some of the hurdles and possible problems that need to be dealt with when moving to more mobile systems.

3.2 Method

To complete the objectives the following set of methods will be used.

A limited literature study will be performed to investigate dynamic schemes and related approaches. This will be done to get a firm basis in current research and to ensure that the techniques used are up to date. The literature study will be limited to focus on dynamic and adaptive sensor management in context-aware and mode-sensitive systems with the purpose of reducing energy consumption. The literature study will be completed when enough existing works in related areas have been processed to be able to develop a suitable scheme. The scheme should be based on the existing approaches that have been discovered in related areas.

After this, a scheme will be developed and implemented. The scheme will be devised for a single platform with multiple sensors and will thus have a centralized architecture as discussed in chapter 2.1.1. The scheme will be implemented using C++ and TENA on a platform running Linux.

Implementation was chosen as a method in order to get a result based on real tests in a real world situation. Alternative approaches would generate more general answers but they would be theoretical and not applied on a real platform. As implementation of a scheme was chosen as a method the best way to evaluate it is by a quantitative experimental method. With a quantitative method, numerical results will be gained that conclusions can be drawn from. An experimental evaluation method will be suitable for this as the tests performed will generate numerical values. The scheme will therefore be evaluated by performing a series of tests. These tests will be performed in a lab under controlled circumstances where the consumed energy will be measured and recorded. Sensor data will be gathered and communicated in accordance with the different states of the scheme. The measured energy consumption for each state will be compared and contrasted with each other. The values of the different states will then be mapped to a typical execution pattern. This will be compared to a baseline approach of a static measurement scheme in order to evaluate the reduction of the energy consumption.

An alternative to developing and implementing a scheme in order to answer the question might be to perform an extensive literature survey where existing techniques and approaches would be thoroughly investigated, compared and contrasted with each other. That would perhaps lead to a more general answer. However, this would be a theoretical approach that would lack the assurance an applied approach might give.

An alternative to implementing a developed scheme could be to evaluate the scheme by simulating how it would behave when implemented. The advantage of this would be that full control over the platform could be had as it would be a simulation and it could therefore be manipulated to behave in a desired way. The disadvantage here is that it is a theoretical approach without applied results.

The approach chosen in this thesis, to develop and implement a scheme on a real physical platform and evaluate it with an experimental method, will generate an answer that will give a clear indication of how much the energy consumption can be reduced. The chosen approach will, to some extent, be tied to the platform and might not generate identical answers on different platforms. However, it will give an indication of the possible gain that can be achieved by using a dynamical measurement scheme.

3.2.1 The platform

The platform that will be used in this thesis is The Gumstix Overo FE COM mounted on the Gallop43 Expansion board. The Overo FE is run by a Texas Instruments OMAP 3530 which has an ARM CPU running at 600MHz. 512MB of RAM is present as well as both Wi-Fi and Bluetooth. The Gallop43 expansion board provides a Global Positioning System (GPS) receiver, 3-axis accelerometer and a set of signal inputs where additional sensors and components can be connected. The Overo computers run a version of embedded Linux called the Ångström Distribution. The version of the Ångström distribution used in this thesis is based on a custom Linux kernel of version 2.6.39 with features for power management made available.

The energy consumption of the Overo is 1000 milliwatt without Wi-Fi and Bluetooth. Bluetooth has an energy consumption of 165 milliwatt at high performance and 66 milliwatt at standby. Wi-Fi has an energy consumption of 760 milliwatt at high performance and 132 milliwatt at standby.

3.2.2 Sensors

The sensors that will be used in the implementation are the built-in GPS and accelerometer together with temperature and pressure sensors.

Temperature and pressure sensors will be connected via the inputs on the Gallop43 board. The energy consumed by these sensors is not provided by the platform used and will not be included in this project. The temperature and pressure sensors will be used to follow change in temperature and pressure.

The accelerometer present on the Gallop43 is a 3-axis accelerometer from ST Micro with the name LIS33DE. The accelerometer has an energy consumption of less than one milliwatt. The accelerometer will be used to register change in movement.

The GPS available on the Gallop43 is the NEO-5G from U-blox. It has an energy consumption of between 77 and 88 milliwatt. The interaction with the GPS will be realized through the gpsd software provided by the Linux operating system (OS). The version of gpsd used in the project is 2.95. The GPS will be used to track position and speed.

3.2.3 Application

The TENA middleware will be used for the implementation. A version of TENA specially tailored for the Overo is available. An existing TENA-application for the communication of sensor data from vehicles will be used and modified. The application will be extended with a dynamic measurement scheme that will control how the data is gathered from the sensors and how and when the application will communicate with the connected listener applications.

4 Related research

In energy-efficient sensor management there are two major areas where significant research has been performed. These are wireless sensor networks and context-aware systems.

4.1 Wireless sensor networks

A wireless sensor network is a mesh of sensor nodes that communicate directly with the sensor nodes that are in range. By laying out a grid of sensor nodes, a large area can be covered. Sensor data is propagated to specific nodes often called sink nodes or base stations. A sensor node is generally a fairly simple device which consists of three parts; one part for sensing and acquirement of data from the environment, one part to perform calculations and processing of data and one part for communication and data transmission. A battery is usually what powers the nodes in a sensor network.

There have been vast amounts of research done in the area of wireless sensor networks. To provide a comprehensible overview of the area, Anastasi et al. (2009) present a survey on energy conservation in wireless sensor networks. Anastasi et al. claim that the communication is usually the most energy-costly component of a wireless sensor node. Therefore we want to minimize the use of communication and let the communication subsystem sleep as much as possible. In some wireless sensor network the nodes are equipped with sensors that consume a considerable amount of energy. Here some kind of sensor management scheme is needed to reduce the use of the sensors while still keeping the accuracy needed for the task.

Anastasi et al. (2009) conclude that three main techniques to reduce energy consumption in wireless sensor network exist. These three are duty-cycling, data-driven approach and mobility. Anastasi et al. describe duty-cycling as the behavior of alternating between active and sleep states depending on the present demands. Duty-cycling in wireless sensor networks usually controls the communication device and puts it to sleep when it is not needed. Many approaches to do this in efficient ways have been suggested, most of which take into consideration the whole network and look at a larger scale than one single sensor node.

The data-driven approach aims to reduce the amount of data that is transmitted to the sink node. It has two subdivisions; the first aims at reducing the communication by not transmitting redundant and unneeded data while the second aims at managing the sensors in an energy-efficient way. One approach here is adaptive sampling where the number of samples taken is reduced in accordance to an adaptive sampling scheme. One other approach is to use low power sensors with low accuracy to trigger a more advanced sensor to perform measurements with higher accuracy.

Mobility is the third technique and can be used when individual nodes are moving within the network. Communication can be reduced if data is transmitted close to a sink node as compared to transmitting data far from it.

4.2 Context-aware systems

Context-aware systems are usually implemented in wearable devices where they are used to determine which context the wearer is in. The devices are either smartphones or special devices constructed specially to determine the context of its wearer.

CLAD (Murao et al., 2008) is an example of a special device designed for context recognition. CLAD uses several accelerometers that are placed on different parts of the wearer's body. The data from the accelerometers are then used to determine the context of the wearer. CLAD also has a simple control scheme to reduce energy consumption. When CLAD detects the battery is running low it powers off some of the connected sensor according to a scheme set by the operator.

Many recent research projects in context-aware systems use smartphones as their platform. Smartphones are typically equipped with a range of different types of sensors including accelerometer, GPS and microphone. CenceMe (Miluzzo et al., 2008) is an example of an early context-aware system for smartphones. CenceMe is also a social network based on sharing context and situational information of the phone owner. To prolong battery lifetime and reduce energy consumption CenceMe uses duty cycling to control both communication and sensing components.

The goal of the energy saving approaches found in context-aware systems is often to find the most efficient way to accurately determine the context. SeeMon(Kang et al., 2008) is a context-aware system built for smartphones and other sensor-equipped mobile devices which conserves its energy by deciding the smallest set of sensor necessary to correctly determine the context. The sensor set is dynamically calculated depending on the current context and the sensors not currently needed are left to rest.

Wang et al. (2009) present a framework for an energy efficient mobile sensing system (EEMSS). The core of EEMSS is a sensor management scheme that describes each context-state by certain sensor criteria. Transitions between states will take place when the criteria are met. The sensors used in each state are clearly defined and the sensors that are not necessary are powered off. The system will stay in a state until the sensor readings fulfill the criteria for transitioning to another state. The states and the transition criteria are set by parsing an xml-file and can therefore be easily adapted to various circumstances. Each individual sensor is then tuned with respects to sampling periods and duty cycles to achieve a good balance between measurement accuracy and energy efficiency.

5 Implementation

Although two related areas of research have been looked through and presented in the previous chapter, no solutions have been found that are directly transferable to the circumstances of this thesis.

In wireless sensor networks the individual sensor nodes are simple and seldom have more than one kind of sensor. The approaches to energy saving in individual sensor nodes are therefore often concentrated on the communication module or the node as a whole. In this thesis a more varied set of sensors will be used that the scheme has to take into consideration. Although many interesting techniques can be found in wireless sensor networks no complete scheme or approach was found that can be directly transferred to the system in this thesis.

Although context-aware systems often use a set of sensors that is similar to the system in this thesis their purpose is specific. Their purpose is to determine the current context of the device or the wearer of the device. This shapes the algorithms used to minimize energy consumption. A common approach is to use an algorithm to find the lowest amount of sensor needed to determine each context. These systems use the sensor data in order to determine context while the system to be built in this thesis is interested in the actual sensed data. In the works of context-aware systems where similar sensors were used they were used in such a way that prevented a direct use of them in this thesis.

As no existing scheme or approach could be taken directly and be built upon, it was decided to construct a scheme from scratch. The scheme combines experience from energy saving in wireless sensor network and context-aware systems to reduce the energy consumption in a mobile sensor device.

From the studies in wireless sensor networks it is seen that changed duty-cycling is a good way to go in order to reduce the energy consumption. It is also seen that avoiding sending redundant and unnecessary data is an effective way to reduce the use of the communication module. Extra calculations are preferable, compared to more communication.

From this the conclusion can be drawn that reducing the communication in the measurement scheme is one thing that should be done. This is done by reducing the amount and increasing the quality of the data being transmitted to the listeners. One goal is also to perform duty-cycling on the communication module and, to some extent, the device as a whole.

From the studies in context-aware systems it is seen that the primary methods to reduce the energy consumed by sensors are to power off the sensors we do not need and to control the sampling frequencies of the sensors. By setting up a context-model the sensors needed to determine that particular context can be used while the rest can be powered off until they are needed.

From this the conclusion is drawn that some kind of context model can be used. The context model will be used both to give a clearer picture to the listeners of what the sensor readings mean and to determine which sensors are needed. The context model will also be used as a foundation to dynamically decide sampling frequencies and communication intervals.

5.1 The Scheme

As the core in the measurement scheme there is a context model from which a series of states are defined that the scheme can operate in. Transitions between states are controlled by a set of criteria based on sensor readings. The transitions possible from one state to another will be clearly defined and based on sensor readings. The states also serve as a foundation to dynamically decide sampling frequencies and communication intervals. This is somewhat similar to what was done as part of EEMSS (Wang et al., 2009).

There are two groups of sensors. The first records movement and consists of the GPS and the accelerometer. The second monitors the vehicle environment and consists of temperature and pressure sensors. The first group is present on the expansion board of the Gumstix device and can be controlled to a certain degree. The second group is connected through Analog-to-Digital Converter (ADC) input channels and cannot be controlled in any form other than the frequency of which we read the input channel.

Based on the two groups of sensors the context model is divided in two parts; movement-related and environment-related. The moment-related context group have two contexts; Still and Moving. The accelerometer is used to determine whether the vehicle is moving or not.

The context is Still when the vehicle is not moving and the engine is off. The GPS is off but samples are taken of temperature and pressure. Communication can be done with long intervals that allow the scheme to sleep for relatively long periods. As long as the scheme is in the Still context temperature and pressure are not used to determine the context.

The context is Moving if the vehicle has the motor running. The accelerometer is used to measure the vibrations generated by the running engine and the GPS is turned on to measure position, speed and direction. Temperature and pressure sensors are sampled and the communication interval is based on their context. Figure 3 shows the movement context model and the possible transitions.

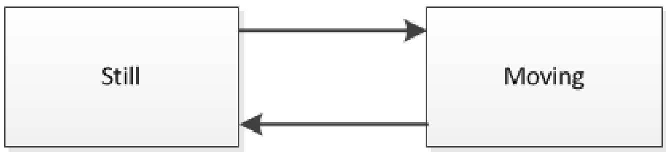

Figure 3 Movement contexts

The environment-related context group have three contexts; Normal, Medium and High. These contexts are named after how interesting they are. The environment-related contexts are only active when the movement context is set to Moving. They take both temperature and pressure into consideration when determining the current context.

The Normal context is considered to be where the system will spend the majority of the time when the vehicle is running. This is where the scheme will be when nothing stands out of the ordinary. The level of interest is therefore relatively low and the context is named Normal. The sampling and update frequency is relatively low to keep updates on a level where energy is saved and enough data is still transmitted to keep the accuracy of the measurements valid.

Medium context is active when temperature is lower than the expected value and the pressure value is not in the High context. In the Medium context there is a slightly higher sampling and update frequency compared to the Normal context. The Medium context is expected in a startup phase. Here the temperature value is expected to steadily increase until it reaches a normal level where a transition to the Normal context occurs. The level of interest in the Medium context is slightly higher than the Normal context as it is expected in a startup phase. The startup phase is a phase where it is interesting to follow the sensed values until they have reached a level where they are expected to stay. If the Medium context is reached while not in a startup phase something is outside the ordinary which motivates more frequent updates to provide a clearer view.

The High context becomes active when temperature is higher or pressure is either lower or higher than expected. Here the sampling and update frequency is relatively high. If values are acquired that place the scheme in the High context it is suspected that something is wrong and more data and more frequent updates is wanted. The level of interest is high as something causes either the temperature to be too high or the pressure to be lower or higher than what is healthy for the vehicle. The correlation between temperature and pressure and the change over time is what is interesting. Any possible correlation between temperature and pressure and the data the GPS provides is also interesting and might provide a clearer picture on how, why and when the vehicle starts to show faulty behavior. The more frequent updates provide a clearer picture and give more data that result in an overall clearer picture of the state of the vehicle. In figure 4 the environment context model and the possible transitions can be seen.

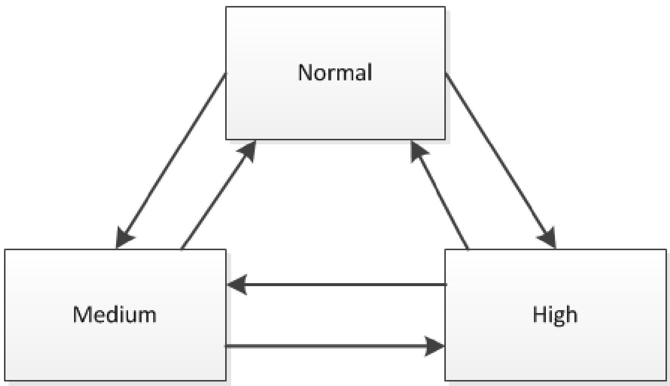

Figure 4 Environment contexts

5.1.1 State transitions

When combining the two parts of the context model, four states are acquired that the scheme can operate in. This is because the environment contexts do not come into effect when the Still context is active. The Moving context could be said to contain the three environment contexts. A more complete model over the states and the transitions between them is shown in figure 5.

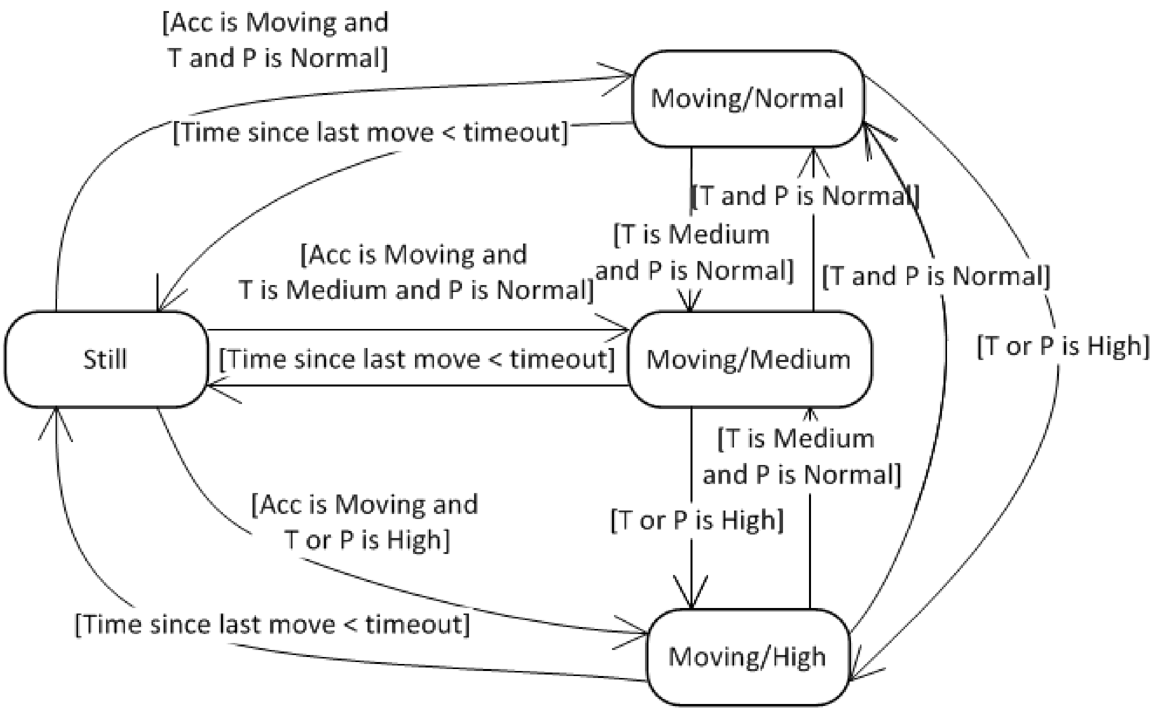

Figure 5 State Transition Diagram

Figure 5 shows all states and all possible transitions between them. T stands for Temperature, P for Pressure and Acc for accelerometer.

The temperature and the pressure have individual bounds set that decide which context they are in. The combination of the temperature context and the pressure context then generates the environment context which in turn is used when the scheme decides which state the system should be in.

To decide which context the temperature is in two variables control the bounds. The first variable sets the border value between Normal and Medium. A lower value will give a Medium context and a higher value might give a Normal context for the temperature. The second variable sets the border between Normal and High. A lower value might give the Normal context and a higher value will result in the High context for the temperature. A temperature value between the bounds generates a Normal context for the temperature.

The situation for pressure is similar. Two variables control the bounds. The first sets the lower border of what is Normal. A lower value will give a High context and a higher might give the Normal context for the pressure. The second variable controls the upper border of what is Normal. A higher value will give a High context and a lower might give a Normal context for the pressure. A pressure value between the bounds generates a Normal context for the pressure.

If the Still state is active and movement is being registered by the accelerometer a transition is made to one of the moving states depending on the values of temperature and pressure. If one of the moving states is active a transition back to the Still state can only occur if no movement is registered during a specified period of time. This period of time is controlled by a variable that holds the timeout value.

The High state is active if either temperature or pressure is within its bounds. To leave High for Medium or Normal both temperature and pressure have to move outside its bounds.

The Medium state is active if temperature is within the bounds set for Medium and pressure is not within the High bounds.

The Normal state is active when both temperature and pressure is within the Normal bounds.

5.1.2 Parameters

The condition variables that regulate the transitions between the different states are general and can be set by passing along parameters when the application is started. This means that the behavior of the application can be adjusted to a certain degree. The variables that declare the border values which cause the scheme to switch state can be set by providing the sensor values that should act as bounds. The timeout variable that determines when the scheme should go to the Still state can be set by providing the number of seconds to wait.

The accelerometer can be controlled by setting how sensitive it should be when detecting movement. It can also be deactivated, which will cause the scheme to always consider itself to be in a moving state.

The GPS can also be disabled by using a simple startup argument. The GPS will then be completely left out of the execution and will not be used in any regards.

An important group of parameters that can be controlled at startup is the length of the periods. The length of the periods decides with which frequency the sensors should be sampled and the sensed data communicated. This can be controlled by setting the amount of time that should pass between updates and sensor sampling. It can be set for each state by providing the time as a number of milliseconds at startup. If these parameters are not set at startup, default values will be used. The default values are set to one second for High, five seconds for Medium, ten seconds for Normal and thirty seconds for Still.

5.1.3 The algorithm

To give an understanding of how and when the scheme determines context and collects data the main algorithm is shown as a model in figure 6. A more in-depth description of the algorithm in pseudo code is provided in appendix A.

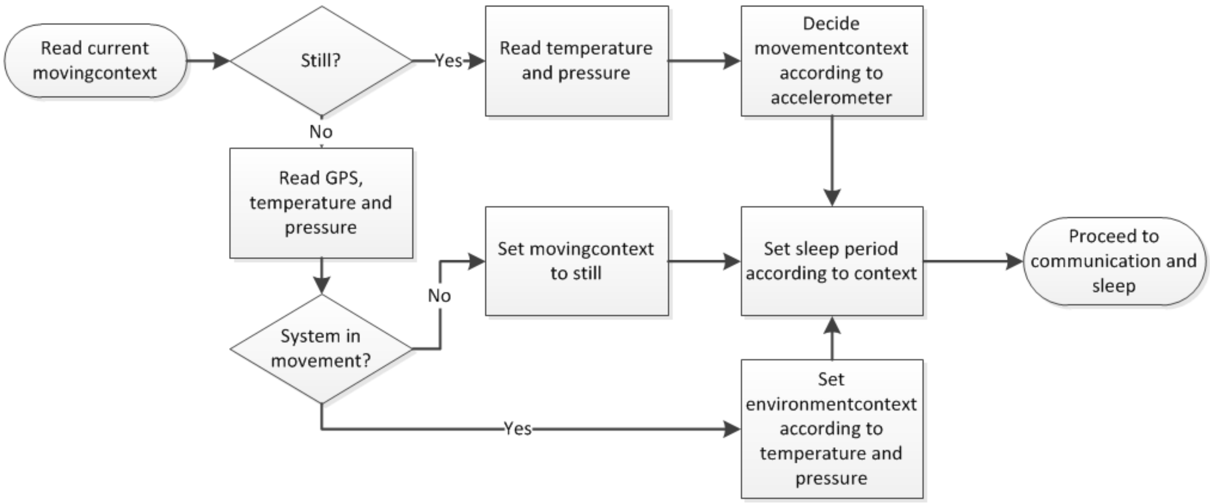

Figure 6 Simplified model of the algorithm

The algorithm starts out with checking what the current moving context is set to. If it is in the Still context state a route is taken where the temperature and pressure values are first read from the sensors. Then the accelerometer is checked and the right moving context is decided. If the accelerometer is moving the moving context is changed. After this the time to sleep is set depending on which context state is active. It then proceeds to communication and sleep for the set time period.

When it is time to wake up the algorithm starts over at the beginning of the model where the moving context is read. If the moving context is set to moving the other route is taken. Here values from the GPS are gathered in addition to temperature and pressure. After this the accelerometer is checked for movement. If there is movement the moving context is left as moving and the environment context set in accordance with the values gathered from temperature and pressure. The time to sleep is then set according to context. Communication and sleep follows.

If the moving context is set to moving and the system is no longer in movement when checked, the moving context is set to Still and the time to sleep is then set. The values of temperature and pressure are not used to decide context when the moving context is set to Still.

5.2 The System

To realize this measurement scheme the model in figure 7 has been constructed and used. The model is based partly on the model for sensor management described in the background chapter and partly on some of the models used in context-aware systems.

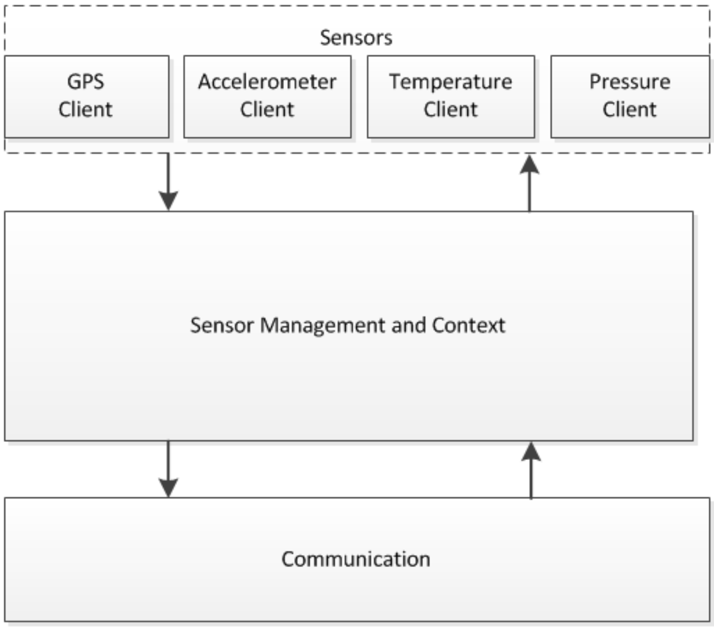

Figure 7 System model

The system model consists of three major parts. The sensors part consists of separate client modules that provide an interface to the specific sensors. Each sensor client receives instructions from the sensor management and context module. This would typically be actions such as initialization and shutdown. The sensor management and context module also gathers the data from the sensors and processes this to determine the current context.

The context is then determined and set. The gathered data is then passed on to the communication module. Sensor management and context can also take instructions from the operator through the communication module. These instructions could be such as suspending the device for a given time. The communication module takes data from the data fusion module and transmits this to the listeners. It can also receive instructions that it passes on to the sensor management and context module. It is in the communication module that TENA comes into play. All external communication is done through the TENA middleware.

5.2.1 Temperature and Pressure

The temperature and the pressure sensors are connected to the system through Analog-to-Digital Converter (ADC) channels. These sensors are the most important to the application as it is their values that indicate whether something is wrong with the vehicle our system will be deployed in. The temperature is measured in degrees Celsius and is expected to range from regular air temperatures to a few hundred degrees Celsius. Pressure is measured in kilopascal and is expected to range from around zero to around one hundred negative kilopascal.

Both the temperature and the pressure sensors are external sensors with power supply outside the reach of the system. The only way to interact with them is through the ADC channels. The only thing the ADC provides is the value of the amount of voltage that is applied to the specific channel. The values gained from the ADC are mapped against real values of temperature and pressure. In terms of power consumption the ADCs effect on the system is constant whether or not the values are read.

The temperature sensor and the pressure sensor are handled by the Temperature Client and the Pressure Client. These modules are fairly simple as all they do is read the values provided by the sensors and translate these into real values in the appropriate units.

5.2.2 Accelerometer

The accelerometer is an important component in deciding which context the system is in. It is the accelerometer that decides if the context is still or moving.

The accelerometer that is present on the used platform is a three dimensional accelerometer with a very low power consumption of less than one milliwatt. The values of the three dimensions are read and combined to get a momentary value of the combined force exerted on the device. To be able to detect movement the change in the exerted force is measured over a short period of time.

In this implementation the accelerometer is handled by the Accelerometer Client. The scheme calls the accelerometer client to ask whether it has detected any movement. To detect movement a sample is read from the accelerometer and after a short delay another sample is taken. These samples are compared against each other and if they differ the accelerometer is deemed to be in movement. This means that each call to the accelerometer client has a small delay where it waits to see if the accelerometer’s values changes. But as this call is only made once per cycle it does not have a major impact on the application.

5.2.3 GPS

An important sensor component of the system is the GPS. The GPS provides data about position, velocity and time. For this system the positioning data is valuable as the ability to track the unit and see if there might be some correlation between movement, temperature and pressure. The GPS available on the platform has an energy consumption of between 77 and 88 milliwatt.

For easier access to the GPS the OS provides the gpsd software. Gpsd is a service daemon that provides a simplified interface to the GPS. It is commonly used within the Linux community and handles the direct communication with the GPS device. This includes providing the latest positioning data as well as instructing the GPS when to power down and sleep. A GPS usually provides data on a regular interval of one second. This means that you cannot request data from the GPS and expect it to be readily accessible whenever you want it. Before you can get anything from the GPS it need to get a valid fix by searching for available satellites. This can take a variable amount of time. When a fix is acquired new positioning data will be continuously provided. Gpsd works by collecting this data and making it easily accessible to the client application.

In this implementation the connection with the GPS is handled by the GPS Client. The GPS Client opens up a connection to gpsd and tells gpsd that it is interested in data from the GPS. Gpsd then starts communicating with the GPS and tries to acquire a fix with valid positioning data. The GPS Client starts an internal thread that polls gpsd for updates from the GPS with a regular interval. This thread keeps on running until it is told that no more data is needed. When the GPS Client is told to end its connection to the GPS it first stops polling for new updates and then tells gpsd that it is no longer interested in GPS data. After this, the connection is closed and cannot provide any valid data until it is told to acquire a new connection to gpsd. The call to the GPS Client to retrieve data takes a value as a parameter. This value specifies the timeout of the call. As it can take a while to get a fixed position there is no guarantees that a fix will be acquired within reasonable time. If the timeout value is reached, the call will end the search for a fix and return zero-values. This is done to allow the algorithm to continue even if no GPS data is accessible.

5.2.4 Communication

For the communication of the system, two options are provided by the platform. There is Bluetooth and Wi-Fi. The choice between these does not alter the shape of the algorithm or the application to any degree. What the choice does affect is the impact the scheme has on the overall performance and energy consumption of the system.

There are a few major differences between Wi-Fi and Bluetooth. The energy consumption of Bluetooth is considerably lower while Wi-Fi has higher transmission rates. As the amount of data the system need to communicate is well within the bounds of what the transmission rate of Bluetooth can handle, Bluetooth is chosen as the means of communication.

The Bluetooth module has different modes of operation where the energy drain is considerably lower when it is not transmitting or receiving. This means that the amount of data transmitted or received should have an impact on the overall energy consumption.

5.2.5 Power management

A very important aspect of the measurement scheme is to make use of the power management features in the hardware. The main feature that was believed to have an impact on the result was a deeper sleep mode. The manufacturer of the platform used in this project claims the core of the platform has an energy consumption of 1000 milliwatt. If the system is allowed to sleep when idling, it is claimed that the energy consumption of the core can be lowered by around 350 milliwatt. This is a reduction of 35% and could be of significant importance.

By activating sleep-while-idle the system will be allowed to reach the sleep state while no processes are active and the processor is idle and waiting for new work. However, to allow the system to go to sleep the serial Universal Asynchronous Receiver/Transmitter (UART) ports of the system have to be sleeping as well. If the serial ports are active they will prevent the system from going to sleep even if the system is idle. The sleep of a serial port is achieved when a timeout is reach without communication occurring over the serial port.

In this system two serial UART ports are used. These are the means by which the GPS and the Bluetooth module talk to the system. As both GPS and Bluetooth are used extensively this will have an impact on the application. To be able to make us of sleep-while-idle the serial ports have to go to sleep, which they can only do if there is no communication over them. As soon as the communication starts they will wake up and prevent further sleep until the timeout is reached. What the scheme needs to do is to try to minimize the use of these serial ports and try to keep the use of them concentrated to a short period of time. This is done by trying to be as smart and restrictive as possible in the use of Bluetooth and the GPS.

In this system the control over Bluetooth is restricted as TENA handles the external communication of the application. All that can be done is to regulate how often to transmit information.

The GPS can be controlled to some extent. But as all communication with it is done through the gpsd software the control is a bit limited. The scheme try to minimize the GPS usage and the communication with the GPS by telling gpsd that it is interested in GPS data right before it actually needs it. Immediately after receiving the desired data, gpsd is told to stop the communication with the GPS. More than this cannot be done, much relies on gpsd and that it will have time to tell the GPS to stop sending updates in order for the serial port will get down to sleep fast enough for there to be any valuable gain.

5.2.6 Overview of energy consumption

To get an overview of the energy consumption of the system this chapter summarizes the consumption of the major components and what can be done to reduce the energy consumption of each component.

The Overo computer unit that is the core of the platform has a reported energy consumption of one watt. The Overo has a power input of five volts. When running on five volts the Overo consumes 200 milliampere (mA). The sleep mode of the Overo is claimed to reduce the energy consumption with 350 milliwatt (mW). With our supply of five volts the consumption of the Overo in sleep would be a reduction of 70 mA to a total energy consumption of 130 mA.

The Wi-Fi module has a claimed energy consumption of 760 mW at high performance when transmitting and 132 mW at standby. At five volts this would mean a consumption of between 152 mA and 26.4 mA. Wi-Fi is not used as part in this project but as there were problems with faulty drivers, it could not be turned off without causing the whole system to become unstable. During the whole project the Wi-Fi was therefore left powered on but unused. Wi-Fi consumed approximately 100 mA in the state it was left in.

Bluetooth has an energy consumption of 165 milliwatt when transmitting in high performance and 66 milliwatt at standby. At five volts this is 33 mA in high performance and 13.2 mA when standing by.

The GPS has a claimed energy consumption of 77 mW to 88 mW. Recalculated to milliampere this becomes 15.4 mA to 16.5 mA at five volts.

These numbers show that the method to save most energy is to let the Overo sleep. A smart use of Bluetooth will also be significant. To not let the GPS work more than necessary will help in keeping the consumption at a lower level.

6 Results and Analysis

To properly evaluate the scheme, measurements of the energy consumed during a typical execution in its proper environment should be done and compared with the baseline approach. The proper environment for the application is for it to be deployed in a military vehicle where it will be running for many hours without pause. The resources or the time to perform valid tests in that setting did not exist within the bounds of this project. To get as good and valid test results as possible with the resources available tests were performed on the different states the scheme can operate in. By knowing how much energy was consumed in each state these values could be mapped to the execution pattern the scheme would have if it was deployed in its proper environment. By doing this the amount of energy that would be consumed by the scheme could be calculated and this could then be compared to the static baseline approach. The static baseline approach was considered to be the same as staying in the High context state for the whole duration of the execution.

As described in chapter 5, the scheme has four states in which it can operate. Tests were therefore performed on each of these states. By simulating values for the temperature and pressure sensors and by setting the parameters of the scheme the execution can be controlled so that the scheme can be properly tested in its various states. The simulated values will not have an impact on the results as there is no significant differences in the way the scheme acquires the simulated values compared to how it acquires real values from the temperature and pressure sensors. A lot of care was taken to insure that all parts of the platform were running stably before any tests were executed. Several trial runs were made before the actual tests were performed. All tests were run after each other without any change to the system to make sure that the same circumstances applied to all executions. For every state, one execution was made were the scheme was measured for five minutes. As the length of the periods is between one and thirty seconds, multiple iterations of even the longest period fit well within five minutes. The effect the scheme has on the energy consumption should therefore be clearly visible during the executions. The measurements of five minutes were therefore deemed to give data with enough validity. These measurements were recorded and later transferred to a computer for analysis.

After the tests were performed on the scheme in its standard version it was clear that the GPS caused a lot of energy consumption by preventing the system from reaching sleep. As the scheme functions properly without data from the GPS, the decision was made to tests the scheme with the GPS disabled. The GPS could be disabled by passing a simple parameter as an argument at the start of the application. The scheme would then only gather data from the temperature and pressure sensors. These results will be interesting in a situation where prolonging the systems lifetime is more important than the data acquired by the GPS. A possible situation where this might be the case could be if the system was deployed in a military vehicle that is out on a longer mission in which it is important to keep the conditions of the vehicle under control. The information provided by the GPS would in this case be less important than getting indications when the conditions of the vehicle start to deteriorate.

All tests were performed in a lab where the Overo was connected to a power supply that generated a steady voltage of five volts. The equipment used to perform the measurement was an advanced digital multimeter (DMM) with the capability to record measurements and plot graphs. The DMM could take samples as often as once per second. Each sample contained a minimum, a maximum and an average value of the current during the time the sample was taken.

6.1 Test results

The collected test results are displayed in this subchapter. Table 2 lists the values gained from the tests on the standard version. Table 3 lists the values gained from the tests made on the version without GPS. The tables list the values as they are recorded by the DMM. No calculations were made to acquire these values. They are just as they were measured and reported by the DMM. The values show average, maximum and minimum of the current over the whole execution made in each state. Each state was executed and measured for five minutes. The current is presented in milliamperes (mA).

Table 2 Test results, Standard version

State	Average current	Maximum current	Minimum current
High	351 mA	377 mA	331 mA
Medium	349 mA	375 mA	263 mA
Normal	344 mA	376 mA	257 mA
Still	314 mA	368 mA	257 mA

Table 3 Test Results, No-GPS version

State	Average current	Maximum current	Minimum current
High	343 mA	376 mA	327 mA
Medium	329 mA	368 mA	257 mA
Normal	322 mA	373 mA	257 mA
Still	309 mA	373 mA	254 mA

6.2 Analysis of standard version

A summary of the results from the tests on the standard version can be seen in table 4. The average value is what is interesting when evaluating the scheme and it is listed in the left column. The right column displays the reduction of the average current for each state compared to the High state. This shows the gain in energy savings to be had for each state lower than the High state which is also the baseline approach. These are the values later used when putting together the typical execution pattern of the scheme in its proper environment.

Table 4 Average current of the scheme's states, standard version

State	Average current	Reduction compared to High
High	351 mA	-
Medium	349 mA	0.6%
Normal	344 mA	2.0%
Still	314 mA	10.5%

In order to get a better understanding of these numbers, subsets of the executions are presented as graphs in figures 8-11. These graphs show subsets with the length of two minutes taken from the longer executions that were made. These subsets are representative of the whole executions and were selected to show patterns in the energy consumption caused by the scheme. The length of two minutes was selected as the generated graphs were easy to read and two minutes were deemed to be enough in order to show the effect the scheme has as each state can go through several periods to possibly show a pattern. The graphs show the data collected by the DMM used to measure the current during the tests. There is one sample for every second and every sample contains a maximum, a minimum and an average of the current during that second. In the graphs there is time on the horizontal line and current on the vertical line. Each sample is shown as a bar indicating the range between the minimum and maximum values. A marker is used to show the average value on every sample.

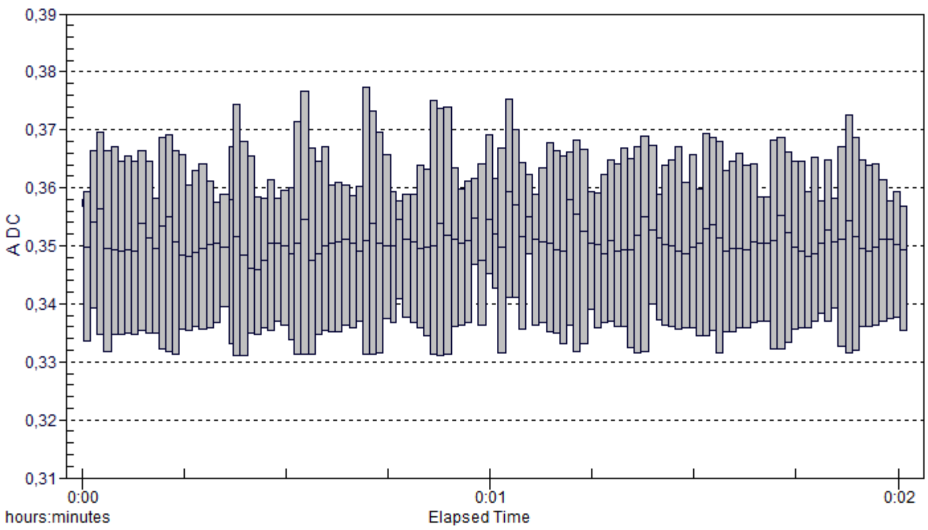

Figure 8 Graph of High state, standard version

In figure 8 a subset of the test on the High state is depicted. In the High state there is a period of one second which means that the scheme does the same thing every second. As the measurement equipment take one sample per second, the scheme should have the same effect on every sample taken. The patterns that can be seen in figure 8 are therefore believed to be caused by elements outside of the scheme's control. It could be related to processes belonging to the OS. Parts of the application that the scheme does not have full control over such as TENA or the GPS could also have some effect on the results. One thing that does not occur in the High state but that hopefully will occur in the other states is for the system to reach sleep. In figure 8 there is a fairly regular current without any major dips which would happen if the system would reach sleep. The short period of the high state means that the Bluetooth and GPS prevent the system from reaching sleep.

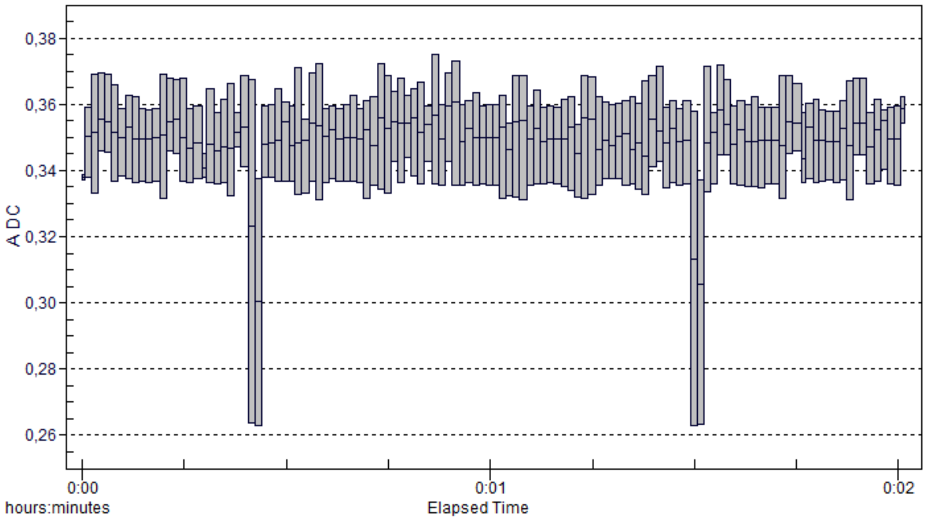

Figure 9 Graph of Medium state, standard version

In figure 9 a graph of the Medium state is shown. Medium has a period of five seconds which mean that the scheme sleep for almost five seconds after it has collected the sensor data and sent this via Bluetooth. If everything functions as desirable the internal communication with the GPS and Bluetooth will cease the moment the scheme has finished its use of them. This will then allow the system to reach sleep. If sleep will be reached in every period the graph in figure 9 will show a major dip in current every fifth second. However, what can be seen in figure 9 is that two dips occur with over a minute between them. This occurs regularly throughout the whole execution. The reason for this is believed to be the GPS which cannot be directly controlled by the scheme. The gpsd software or the GPS in itself is believed to continue communication over the serial port even after receiving the command to stop. It would be desirable to prolong the dips and make them occur more often. However, the scheme does what it can to minimize the use of the GPS and no further approaches could be thought of that might reduce the GPS usage.

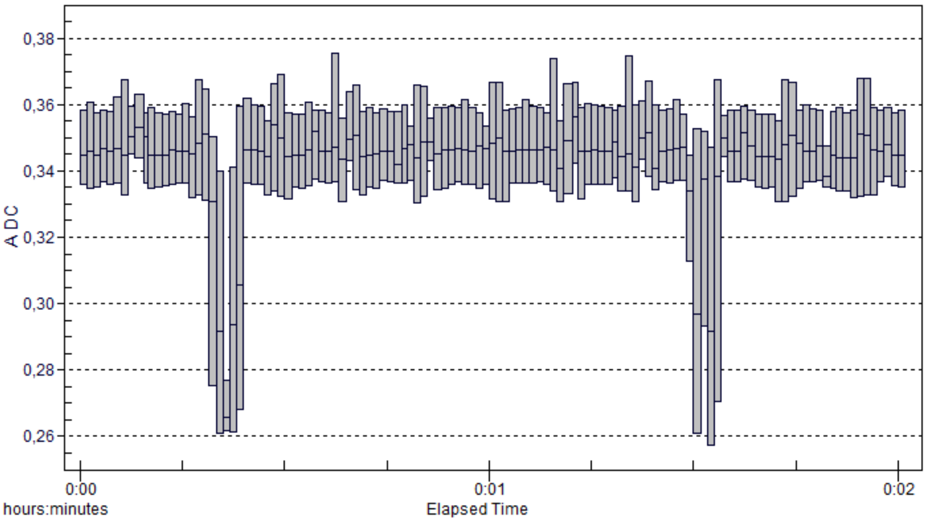

Figure 10 Graph of Normal state, standard version

In figure 10 a graph of the Normal state is depicted. Normal has a period of ten seconds and is otherwise the same as Medium. By comparing the Normal graph in figure 10 with the Medium graph in figure 9 it can be seen that they are very similar. The difference is that the dips in the Normal execution stretch over a longer period of time. Other than more time spent in sleep the same pattern is present. There is approximately one dip per minute and what occurs in between the dips does not stand out in such a way that no certain correlation to the scheme can be made.

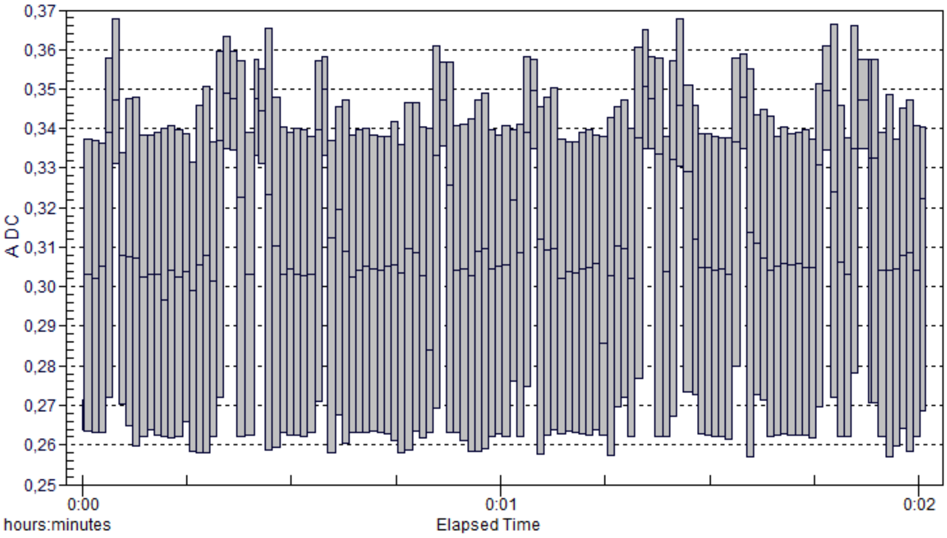

Figure 11 Graph of Still state, standard version

In figure 11 a graph of the execution in the Still state is shown. When the scheme is in the Still state the GPS is turned off and the period is set to thirty seconds. This leaves Bluetooth as the only component that will keep the system from reaching sleep. Every time Bluetooth is used communication will occur over the internal serial port which will prevent the system from reaching sleep. The scheme sends data over Bluetooth one time per period. This means that if the scheme was the only thing using Bluetooth the graph in figure 11 would show four peaks where the Bluetooth would become active and prevent the system from reaching sleep. The graph shows several occasions where sleep is prevented that cannot be explained by the scheme. As TENA is in charge of the external connections and communication, TENA is believed to be the cause of most of these. The time between the uses of Bluetooth the system is in a state where it can easily reach sleep when no processes need to be executed. This probably occurs many times per second and can be seen in figure 11 where the range of the current, when sleep is not blocked, is between 260 mA and 340 mA. When the system sleeps the energy consumption lays around 260 mA and when awake and working it stays around 340 mA. The average current when the system is allowed to sleep is around 300 mA and the average current when sleep is blocked is close to 350 mA. These values are similar to what is seen in previous graphs. The difference is the amount of time sleep is allowed.

6.3 Analysis of No-GPS version

As seen in the previous subchapter, the GPS is believed to cause significant energy consumption by preventing the system from going to sleep. By performing the same tests again with the GPS disabled it can hopefully be seen how much energy can be saved without the GPS. The functionality of the scheme will not be affected by disabling the GPS. There will however be a loss of data as the data collected from the GPS will not be available. The tests were performed in exactly the same way as for the standard version. A summary of the results can be seen in table 5. The average is the average current over the whole execution. The right column shows the reduction in average current compared to the High state.

Table 5 Average current of the scheme's states, No-GPS version

State	Average	Reduction compared to High
High	343 mA	-
Medium	329 mA	4.1%
Normal	322 mA	6.1%
Still	309 mA	9.9%

To look closer at the numbers presented in table 5 graphs of the execution of each state is presented in figure 12-15. As in previous chapter these graphs is subsets with the length of two minutes taken from the longer executions. They were chosen to illustrate patterns and are representative of the whole executions. The graphs show maximum, minimum and average for every sample taken. One sample per second is taken.

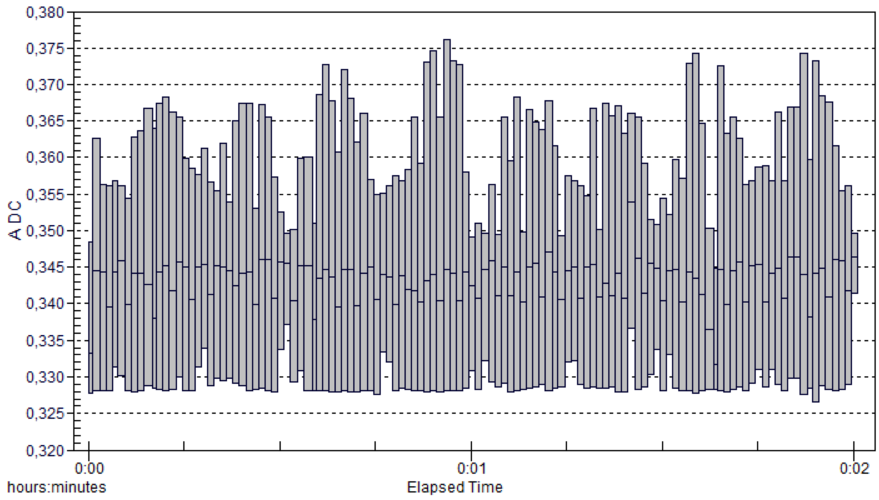

Figure 12 Graph of High state, No-GPS version

In figure 12 a graph of the execution of the High state is shown. The scheme has a period of one second in the High state which means that the scheme should have the same effect on every sample taken. This means that the frequent communication of data over Bluetooth prevents the system from ever reaching sleep. The patterns in the graph cannot be correlated to the scheme. The patterns and fluctuations shown in the graph are believed to be caused by the rest of the system.

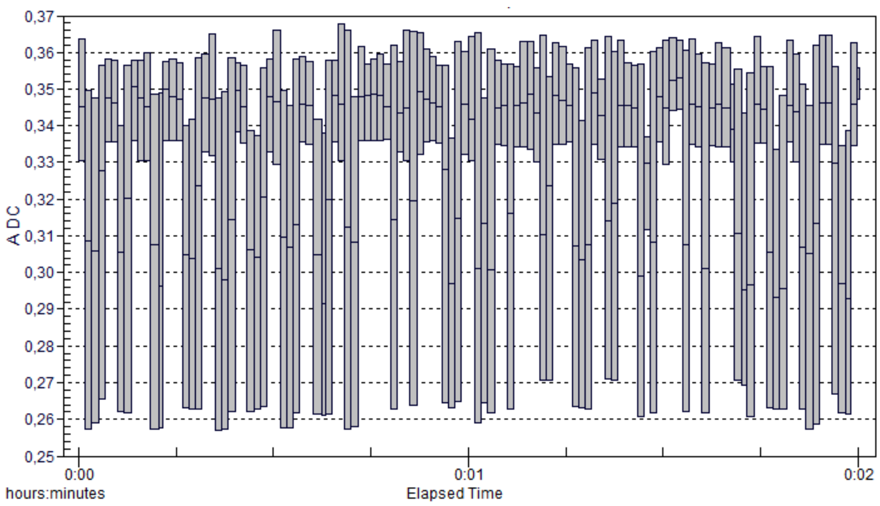

Figure 13 Graph of Medium state, No-GPS version

A graph of the Medium state is presented in figure 13. Medium has a period of five seconds which means twelve periods per minute. As the scheme communicates twelve times per minute, the scheme will also cause the system to be blocked from sleep twelve times per minute. In addition to this the OS and TENA is believed to cause some blockage by using Bluetooth to communicate and maintaining external connections. The graph in figure 13 clearly shows the system going between states where sleep is either allowed or blocked. As the only thing blocking the system from reaching sleep is Bluetooth it is clearly visible whenever Bluetooth is used. The pattern visible in the graph fits well into what is expected to be seen with the scheme communicating regularly and some less regular communication caused by other entities.

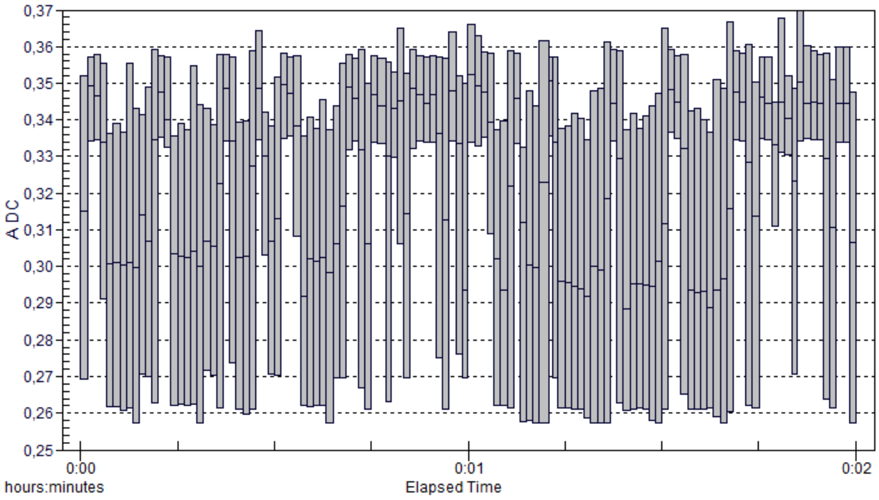

Figure 14 Graph of Normal state, No-GPS version

The graph displayed in figure 14 shows an execution of the Normal state. This execution is expected to be similar to the Medium but to be allowed in sleep for longer periods. With a period of ten seconds the scheme communicates six times per minute. As before, TENA or the OS is expected to use Bluetooth which causes irregularities. The graph shows much of what is expected. The system is allowed to reach sleep for longer periods and a regular use of Bluetooth in combination with some irregularities can be seen the moments where sleep is blocked.

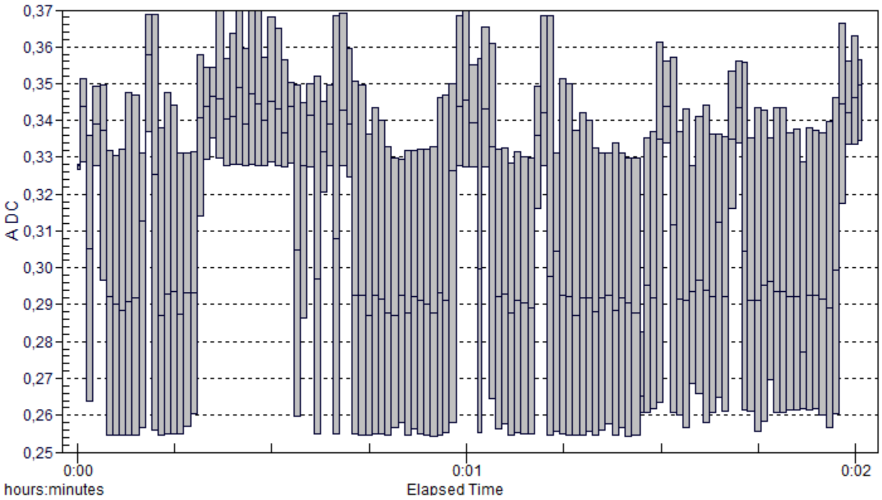

Figure 15 Graph of Still state, No-GPS version

Figure 15 displays a graph of the Still state in which the scheme has a period of 30 seconds. The communication caused by the scheme only occurs twice per minute which means that the system will only be blocked from sleep by the scheme two times per minute. In figure 15 the graph shows that this occurs more often. The long period that can be seen in the first half of the graph where sleep is prevented is believed to be caused by elements outside of the scheme's control. The reason for the long period of blocked sleep can only be speculated in as it does not occur with the same length again during the test execution. It is believed to be caused by TENA or the OS which are communicating with the Bluetooth module.

By comparing the graphs in figures 8-11 with the corresponding graph in figure 12-15 the impact of the GPS is indicated. In the High state the difference is restricted to the change in average current believed to be caused by the disabling of the GPS. The patterns of the executions are otherwise the same with the system always prevented from reaching sleep. In the Still state there is one major difference that stands out. In figure 15 there is a longer period of time where sleep is blocked that cannot be seen in the corresponding graph in figure 11. This is believed to be caused by elements outside of the scheme's control. This is also an irregularity that does not occur again within the time of the test execution. The other differences in the graphs of the Still state are minor as the GPS is not used in any of them. Big differences can be seen in the Medium and the Normal states. In the standard version the system was only allowed to reach sleep once per minute. When disabling the GPS, the system is allowed to reach sleep in every period. The use of Bluetooth then becomes very important as it does not only cause the energy consumption of its own activities but also prevents the system from reaching sleep.

6.4 Analysis of the scheme's effect

The test results presented in the previous chapter show the energy consumption of each state and the reduction for each state compared to the High state. To properly evaluate the scheme these values need to be put into a fitting context. The purpose of the system is to be deployed in a military vehicle where it will be monitoring the condition of the vehicle during various kinds of activities. A fitting context would therefore be to create an execution pattern based on how the system would react if it was deployed in a real situation. By putting the values of the test results into this execution pattern the energy consumption the scheme would have in that scenario can be calculated. This will then be compared to the baseline approach in order to get an answer to the question posed in the problem definition.

The baseline approach is set to be the same as the High state. The High state is where the scheme needs to be if something in the vehicle's condition is faulty or out of the ordinary. The baseline approach is therefore to stay in the high state during the complete lifetime of the system.

To find an execution pattern that is representative of the real world scenarios the system later will be deployed in, information was acquired on how the vehicle is handled. The information was acquired from an expert dealing with the vehicles on a regular basis. It was based on how the vehicle was handled over the course of a typical day during a major exercise with the goal to educate new drivers. The information was given in the percentage of the total time spent in different activities.

Over the course of 8 hours the vehicle's different activities were listed and the amount of time spent in each activity was given in percent. 35% of the time was spent with the vehicle standing still and the motor turned off. 5% was spent standing still with the engine on to prepare for driving. 10% was spent standing still with the engine on to get instructions from the leaders. 30% of the time was spent driving on roads and 20% was spent driving in terrain.

When mapping this to the states of the scheme some assumptions are made. The 5% the vehicle is standing still with the motor running in preparation of driving is assumed to be spent in the Medium state. The basis for this assumption is that the vehicle has been standing still and the measured temperature is probably low which causes the system to enter the Medium state. Of the total 50% spent driving on road or in terrain most of the time will be spent in the Normal state. The High state will be reached when the engine is under much stress and has to work harder than normal. In a vehicle where everything functions as intended this will not occur often. However, the assumption is made that the vehicle used is not functioning perfectly and will therefore occasionally cause the system to enter the High state. The assumption is made that one fifth on the time the vehicle is driven it will be driven in such a way that causes the system to enter the High state. This is a rather high number and it is unlikely that more time will be spent in the High state. This results in 10% of the total time spent in the High state and 40% in the Normal state. The final 10% where the vehicle is standing still with the engine running is assumed to be spent in the Normal state as the engine at these times is warm enough to have left Medium and the High state will not be reached when the vehicle is not driven. In table 6 the percentage of the total time spent in each state is summarized.

Table 6 Typical execution pattern

State	**Still**	**Medium**	**Normal**	**High**
Time in state	35%	5%	50%	10%

When combining the numbers in table 6 with the results from table 4 in chapter 6.2 the energy consumption can be calculated for the scheme in a typical setting. The average current of the baseline approach is the same as the High state, 351 mA. The average current during the complete execution in a typical setting for the standard version of the scheme is calculated to be 334.45 mA and results in a gain in energy saving by 4.7%.

By doing the same for the version of the scheme where the GPS is disabled the values from table 6 and the values from table 5 in chapter 6.3 are used. To get a valid reference point the baseline also has to have the GPS disabled. The baseline therefore has an average current of 343 mA which is the same as the High state. The average current of the version without GPS is calculated to be 319.9 and results in a gain in energy saving by 6.7%. These results are shown in table 7.

Table 7 Effect of the scheme

Version	**Baseline**	**Scheme**	**Reduction**
Standard	351 mA	334.45 mA	**4.7%**
No-GPS	343 mA	319.9 mA	**6.7%**

The reduction of energy consumption to be had by using the scheme is shown in table 7. The next step is to relate this to the change in measurement accuracy and decide whether the potential loss of measurement accuracy can be motivated by the gain in energy savings. The risk that exists with the scheme is that some sudden change in the measured conditions occurs while the scheme is sleeping and that it will not wake up in time to follow the change of the sensed values in a desired way. As long as the scheme is functioning as intended in normal circumstances the longest time the scheme is believed to be able to sleep without detecting a critical change is ten seconds. Ten seconds is the period of the Normal state and if some critical change in the conditions occurs just as the scheme goes to sleep it will take almost ten seconds for the scheme to wake up and register the critical values which most likely will cause the scheme to go to the High state. The possibility also exists that the scheme is in the Still state when critical values occurs at the same time as the vehicle starts to move. If this occurs just after the scheme goes to sleep it will take about thirty seconds for it to wake up and notice the change. This is a very unlikely scenario and even if it did occur the loss would not be that great. The purpose of the system is to measure the conditions of the scheme over time so that the sensed values can be related to each other. As there are no huge risks in using the scheme and the potential loss of measurements will not cause much problem the use of the scheme could be said to be motivated even though the reduced energy consumption is relatively small.

6.4.1 The effect of the platform

The results presented in table 7 shows that the effect of the scheme on the platform used in this thesis stays at a reduction in energy consumption of 4.7%. If the GPS is disabled the reduction can be increased to 6.7%. These results are far more modest than what was hoped for when the work began and it is clear that the platform the scheme is implemented on has a huge impact on how effective the scheme can be. To investigate how significant the platform's effect on the results is, the effect the scheme would have on an ideal platform can be calculated. As an ideal platform like this does not exists and probably never will this is highly theoretical but it gives an indication on how much the platform matters for the results.

The main difference between the four states of the scheme is the frequency in which they are active. High has an update period of one second, Medium has a period of five seconds, Normal has 10 seconds and the update period of Still is thirty seconds. With High used as the point of origin the relative frequency of the updates can be described in percentage. High then has an update frequency of 100%. Medium has a period of 5 seconds which gives a relative frequency of 20% or one fifth of the High frequency. With a period of ten seconds Normal has a frequency of 10%. Still has a period of thirty seconds which gives a relative frequency of about 3.3% compared to High.

These frequencies give a picture of the difference between the states of the scheme. If this is mapped to the typical execution pattern displayed in table 6 the relative frequency of the scheme compared to High can be calculated. By multiplying the amount of time spent in each state with the relative frequency of that state and adding these sums together the relative frequency of the scheme can be calculated. This simple calculation can be seen below:

$$0.10 * 1.00 + 0.05 * 0.20 + 0.50 * 0.10 + 0.35 * 0.033 = 0.172$$

As the frequency of High is the same as the static baseline approach a comparison between the scheme and the baseline approach can be made. The relative frequency of the scheme compared with the baseline approach is 17.2%. This means that the scheme reduces the update frequency by 82.8%.

On an ideal platform the energy consumption would follow the variance of the scheme. On the ideal platform the energy consumption would be zero when the scheme sleeps and energy would only be consumed when the scheme is active. To compare the ideal platform with the one used in this project the energy consumption of the baseline approach can be set to be the same in both. As seen in table 7 the baseline approach had a consumption of 351 mA in the standard version. On the ideal platform the reduction of energy consumption would follow the reduction of the update frequency and therefore be 82.8%. This would result in an average energy consumption of 60 mA. The difference between the scheme's effect on the ideal platform and the platform used in this project is around 274 mA. This gives an indication of how much the platform effects the results.

It is difficult to draw any definite conclusions from this. The argumentation here is based on the length of the different periods for the scheme's states. The fact that the GPS is turned off in one of the states is not considered and other aspects of the scheme and the application, such as TENA, is also left without consideration.

What it does show is that there is considerable amount of improvement to be had between the ideal case and the case of this project. A different, more optimized platform should be able to bring the results closer to, but never further than, the ideal 82.8%. As the results gained on the platform used in this project is only 4.7% there is much room for improvement.

6.5 Analysis of work and process

The work to reach an answer to the stated problem has progressed quite well and no major difficulties have been encountered. Although much could have been done more thoroughly and in depth, given the time limitations on the project, a sufficient amount of work has been done to reach an answer to the stated problem and to give a decent evaluation of the project's success.

To reach an answer to the problem, a set of objectives was declared.

The first objective was to perform a limited literature survey of related works and existing approaches. This survey can be found in chapter 4 and is considered to fulfill the first objective. The limited literature survey captured and summarized approaches and related work found in the area of context aware systems. Much of the related works in the field was processed to some degree. In the area of wireless sensor networks more could have been done. As the area is vast with huge amounts of published work it was hard to find significant related works. However, with the help of published surveys that summarized the works of energy efficiency in wireless sensor network a sufficient degree of knowledge and understanding was reached to be able to continue in developing a scheme with solid grounds.

The second objective was to develop and implement an improved scheme with inspiration found during the literature survey. As no existing schemes could be found that were transferable to the system used in this thesis a scheme was developed from scratch based on techniques and ideas found in related works. The scheme could therefore be said to be an improvement as techniques and ideas used in various other works was applied in a new situation. The developed scheme was based on a fairly simple algorithm that tried to use its resources in the most efficient way possible. Despite minor problems the resulting scheme and algorithm ended up in a way considered to fulfill the second objective and to bring an answer the stated question.

The third objective was to evaluate the scheme and compare it to a static baseline approach. Tests were performed on the different states of the scheme and by that on the static baseline approach as is the same as the High state. Here, time and resources were scarce and the performed test could not be performed to the desired extent. The tests were cut a bit short with only one execution of five minutes per state. However, as much care was taken to ensure that the quality of the performed tests was high, the tests can be considered to be sufficient and provide data with enough validity to give a decent basis for conclusions to be made. The third objective can therefore be considered to be fulfilled.

The fourth objective was to describe the results in the terms of system lifetime during a typical execution plan in the setting. This was done by acquiring relevant information from experts on how the vehicle was handled. From this the execution pattern of the scheme was constructed and the energy consumption of the scheme could be calculated and compared to the baseline approach. This gave an answer to how much the scheme reduced the energy consumption on the used platform. Even though reduced energy consumption is not equivalent with prolonged system lifetime it is considered to give an answer. Because of this the fourth objective is considered to be fulfilled.

The question stated in the problem definition is on a general level and the results gained from the tests are very much bound to the platform used. To provide a picture on how much the platform affects the results, the results the scheme would have on an ideal platform were discussed. By comparing the ideal case with the results gained from the test on the platform the scheme was implemented on and discussing this, the question stated in the problem definition is considered to be answered.

7 Conclusions

A dynamic measurement scheme was developed during the course of this project in accordance with the objectives and methods stated. The purpose of the scheme was to reduce the energy consumption in a mobile sensor unit. The scheme was developed with inspiration from existing works and techniques found in literature. The base of the scheme consists of four states that differ in the use of sensors and the frequency of communication. Transitions between the states were based on sensor readings. The sensors used were temperature, pressure, accelerometer and GPS.

To evaluate the scheme, tests were performed in a lab on the different states of the scheme. This was done to investigate the differences between the states and to see what effect each state had on the system as a whole. In an early stage it was discovered that the functioning of the GPS had a severe impact on the energy consumption. This was caused by the way the system interacted with the GPS. As the scheme functions even without the GPS, test were also performed on a version where it was disabled.

The test on both versions generated numbers which showed the energy consumed and the differences between the states of the scheme. These values were then mapped to a typical execution pattern. The circumstances if the scheme was deployed in a real world situation were investigated. From this a typical execution pattern could be constructed where the amount of time the scheme would spend in each state was known. The consumed energy of the scheme was then calculated and compared with the static baseline approach. Finally, the effect of the scheme could be expressed by how much it should be able to prolong the lifetime of the system.

From the work done in this project and the results gained, a few conclusions can be drawn.

- The consumed energy can be reduced by 4.7% or 6.7% depending on which sensors are used. The tests show that on our platform with the set of components at our disposal the developed scheme can reduce the energy consumption by 4.7% compared to a static baseline approach. If the GPS is disabled the scheme can reduce the energy consumption by 6.7%. These numbers are valid on the platform used in this project and will probably be different on other platforms.

- The platform the scheme is implemented on has a significant effect on the efficiency of the scheme. In the case of this project it was most noticeable by the GPS. Interaction with the GPS caused energy consumption not only by what the GPS used for itself but by preventing the rest of the system from using techniques to save energy. The same issue was experienced with Bluetooth, but it did not cause as much problem. This shows the importance of knowing your hardware and the effects the use of a component can have on the system as a whole.

7.1 Discussion

One weakness in this project is the testing of the scheme. The fact that, for each state, only one proper test was performed is a matter of concern. That each test is not measured for longer than five minutes adds to these concerns. This implicates the reliability of the results. An alternative test procedure or a more thorough testing with longer and several tests per state would be desirable and would increase the reliability of the results. The optimal measurement condition would be to perform measurements in the field with the system deployed and functioning as intended. To make this a reality a lot of time and resources would be needed. This was simply not possible and as time was in short supply test could not be performed to the extent we would want to acquire firmer and more reliable numbers. However, as the tests were performed, it was deemed that the quality of the tests were more important than the quantity of the test data. To best make use of the available time efforts were concentrated to make sure that the test values acquired were valid and representative. The values acquired from the testing are believed to be accurate enough that they can give a valid picture of the scheme and its various states. However, more testing would be good to increase the reliability of the results and in hindsight the tests should probably have been able to be somewhat extended even within the limited time.

During the course of this project it has become clear that the platform and hardware being used has a major impact on how effective a scheme like this can be. The platform used in this project is not optimal for the purpose of a scheme like this. A better result would presumably be achieved if a different platform more suited for the task could be used. However, for this project the platform was determined beforehand. The problems were unknown at the start of the project and were encountered as the project progressed.

Problems with the platform were mainly encountered in the interaction with the GPS. The fact that the GPS communicates over a serial UART port which prevents the system from reaching the sleep mode that consumes less power, makes the usage of the GPS to an important component when trying to reduce energy consumption. When directly control over the GPS is not possible and a service daemon has to be used, it becomes harder to analyze and decide the best way to control the GPS. If another kind of GPS was available or a different way of communicating with it could be had, results might have been different.

The serial UART ports have a major impact on the results. Much energy is wasted when the ports block the unit from going to sleep. As the GPS and Bluetooth are connected to the system through serial UART ports, every time they are used they will consume the energy they need for their own devices but they will also cause energy consumption by blocking the system from sleep for an extended period of time. The more critical of these two problems is the blockage of sleep. If this scheme was implemented on a system where the GPS and Bluetooth were connected to the system by means that did not block sleep, it would probably generate different results where the effect of the scheme would be diminished.

Communication is another part of the system that generated a lot of problems. The Wi-Fi in particular was an issue as the available drivers were faulty and caused numerous problems. When trying to disable the Wi-Fi and cut the power to it, the system became unstable. This is unfortunate as the Wi-Fi module is a major energy consumer in the system. At five volts the Wi-Fi consumes around 100 mA which is almost a third of the total energy consumed by the system. If there had been no problems with the Wi-Fi it would have been disabled and all measured test results in this project would presumably have values that were around 100 mA lower. However, this is a speculation and testing would be needed before making any reliable conclusions.

The operating system is also a factor that might have an impact on the results. In this project a Linux version was used. Linux gives the device a fair amount of customizability and variety but with it comes several processes and functionalities that our scheme has no use of. The use of a different, more lightweight operating system with the smallest amount of necessary functionalities could possibly give more control to the scheme and reduce energy consumption of the system further.

The reduction of energy consumption in sensor units like the ones used in this project could have several positive effects. First, a longer lifetime of the sensor units could possibly lead to less work and time spent on maintenance. This might give the operator an easier workload and decrease time spent on the field to perform maintenance. It might also result in a lowered cost for the organization deploying the sensor units as less work and resources has to be spent on taking care of the sensor units when they can take care of them self for longer periods. Second, a longer lifetime might also result in less frequent change of batteries. If the batteries last longer fewer batteries will be used which will have a positive effect on the environment with less waste and on cost as fewer batteries have to be purchased.

This system is designed to monitor the conditions of a military vehicle use for personnel transport. There are several ethical aspects that can be considered in relation to this. In the long run a variant of this system might end up deployed in the field to monitor the vehicles when they are on missions. The system would provide information on when it is time for the vehicle to be taken in for maintenance. In such a situation the vehicle might have broken down without the system, possibly in hostile territory. That situation could possibly be prevented by using the system as the vehicle might have been called in for maintenance well before its condition turned bad enough for the vehicle to break down. This would give the soldiers involved more security as dangerous situations could be avoided by having access to more reliable equipment.

7.2 Future work

The dynamic measurement scheme developed for this project is customizable in the way that the conditions that regulates the state-changes can be adjusted. The scheme is developed with the specific use of monitoring the conditions of a certain type of military vehicle. It is not general enough to be applied in other situations. An interesting and useful direction to develop it further would be to make it as general as possible. By restructuring the algorithm and providing an input mechanism to configure the scheme a general application could be developed that could be deployed in many different situations.

In this project we have studied the energy consumption and the possible reduction of it that could be had by using a dynamic measurement scheme. This study has been restricted to a specific platform and all the limitations that brings. An interesting study to perform would be to compare different platforms by implementing the same scheme on all of them. This would be done to investigate how much the platform matter for the energy consumption and to see the impact a dynamic measurement scheme can have if the platform differs in various ways.

Another study that could provide a different point of view on the problem would be to evaluate a developed scheme by simulating an ideal platform and calculate how the scheme would act. This would be similar to what was done in chapter 6.4.1 in this project. This could be done on a greater scale with more consideration on how an ideal platform would look like and react.

References

Anastasi, G., Conti, M., Di Francesco, M., Passarella, A., (2009) Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*. 7(3), 537-568.

Baldauf, M. & Dustdar, S. (2004) A Survey on Context-aware systems. *International journal of ad hoc and ubiquitous computing*.

Benini, L., Bogliolo, A., De Micheli, G. (2000) A Survey of Design Techniques for System-Level Dynamic Power Management. *IEEE transactions on very large scale integration(VLSI) system. 8*(3), 299-316.

Boström, H., Andler, S.F., Brohede, M., Johansson, R., Karlsson, A., van Laere, J., Niklasson, L., Nilsson, M., Persson, A., Ziemke, T. (2007) On the Definition of Information Fusion as a Field of Research. *Technical Report: HS-IKI -TR-07-006, School of humanities and Informatics, University of Skövde.*

Britt, W.R., Miller, J., Waggoner, P., Bevly, D.M., Hamilton, J.A. (2011) An embedded system or real-time navigation and remote command of a trained canine. *Personal and Ubiquitous Computing 15* (1), 61-74.

Buede, D.M. & Waltz, E.L, (1990) Issues in Sensor Management. *5th IEEE International Symposium on Intelligent Control, 2*, 839-842.

Gellersen, H.W., Schmidt, A. & Beigl, Michael (2002) Multi-Sensor Context-Awareness in mobile Devices and Smart Artifacts. *Mobile Networks and Applications 7*(5), 341-351.

Kang, S., Lee, J., Jang, H., Lee, H., Lee, Y., Park, S., Park, T., Song, J. (2008) SeeMon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In: *MobiSys'08 - Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, Breckenridge, p. 267-280.

Liu, T., Sadler, C.M., Zhang, P., Martonosi, M., (2004) Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and ZebraNet. *MobiSys 2004 - Second International Conference on Mobile Systems, Applications and Services*. 256-269.

Lorch, J.R., & Smith A.J. (1998) Software Strategies for Portable Computer Energy Management. *IEEE Personal Communications*. *5*(3), 60-73.

Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T. (2008) Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In: *SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems*, New York, p. 337-350.

Murao, K., Terada, T., Takegawa, Y., Nishio, S. (2008) A Context-Aware System that Changes Sensor Combinations Considering Energy Consumption. In: *Pervasive '08 Proceedings of the 6th International Conference on Pervasive Computing,* Berlin, p. 197-212.

Musah, A. & Dykstra, A. (2008) Power-Management Techniques for OMAP35x Applications Processors. Available on Internet: www.ti.com/lit/an/sprt495/sprt495.pdf [Accessed 28 February 2012]

Ng, G.W. & Ng, K.H. (2000) Sensor management - what, why and how. *Information Fusion 1* (2), 67-75.

Noseworthy, J.R. (2008) The test and training enabling architecture (TENA) - supporting the decentralized development of distributed applications and LVC simulations, *12th 2008 IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, 259-268.

Tennies, N. (2003) Software Matters for Power Consumption. *EE Times*. Available on internet: http://www.eetimes.com/design/embedded/4024506/Software-Matters-for-Power-Consumption [Accessed 28 February 2012].

Wang, Y., Lin, J., Annavaram, M., Jacobson, Q.A., Hong, J., Krishnamachari, B., Sadeh, N., (2009) A framework of energy efficient mobile sensing for automatic user state recognition. In: *MobiSys'09 - Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services*, Krakow, p. 179-192.

White, F.E. (1998) Managing data fusion system in joint and coalition warfare, in: *Proceedings of the European Conference on Data Fusion*, Malvern, UK, p. 49–52.

Xiong, N. & Svensson P. (2001) Multi-sensor management for information fusion: issues and approaches. *Information Fusion, 3*(2), 163-186.

Appendix A - The algorithm

Here is the core algorithm of the scheme presented in pseudo code:

```
if movingContext is still
    if no GPS fix and GPS is not disabled
       Start timer
       Initialize GPS
       Get GPS data within timeout
       Shutdown GPS
       Stop timer

    Get temperature and pressure data

    if moving
       set movingContext to moving

else if movingContext.Moving

    if GPS is disabled
       Get temperature and pressure data

    else if environmentContext is Normal
       Start timer
       Initialize GPS
       Get GPS data within Normal timeout
       Shutdown GPS
       Stop timer

       Get temperature and pressure data

    else if environmentContext is Medium
       Start timer
       Initialize GPS
       Get GPS data within Medium timeout
       Shutdown GPS
       Stop timer

       Get temperature and pressure data

    else if environmentContext is High
       Start timer
       Initialize GPS
       Get GPS data within High timeout
       Stop timer

       Get temperature and pressure data

    if not moving and timeout is not reached
       set movingContext to still

    else

       if temperature < temperature low limit
           set temperatureContext to medium
       else if temperature >= temperature low limit and
                temperature <= temperature high limit
           set temperatureContext to normal
       else if temperature > temperature high limit
```

```
            set temperatureContext to high


        if pressure < pressure low limit
            set pressureContext to high

        else if pressure >= pressure low limit and
                 pressure <= pressure high limit
            set pressureContext to normal

        else if pressure > pressure high limit
            set pressureContext to high


        if temperatureContext is Normal and
                   pressureContext is Normal
            set environmentContext to normal

        else if temperatureContext is High or
                   pressureContext is High
            set environmentContext to high

        else if temperatureContext is Medium or
                   pressureContext is Medium
            set environmentContext to Medium


sleep the time left of period
```