Tests and further development of a wireless body area network system

Master of Science Thesis

JIM RYBERG

Department of Signals and Systems
Division of Communications Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010
Report No. EX055/2010

Tests and Further Development of a Wireless Body Area Network System

Jim Ryberg

December 1, 2010

Master of Science Thesis (30hp)

CHALMERS UNIVERSITY of TECHNOLOGY

Department of Signals and Systems
Division of Communication Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010
Report No. EX055/2010

Abstract

This is a master thesis of which content is tests and development of an existing Wireless Body Area Network (WBAN) prototype. The prototype is a part of the Time and Resource Demanding Control with Surveillance of an Individual's Status (TRIS) project at the Swedish Defence Research Agency (FOI). The purpose of the TRIS project is to investigate the possibilities of developing a system concept for a decision support system for monitoring of an individual's status. The purpose of this thesis is to support the TRIS project in its aim to deliver a proof of concept prototype with all the main requirements of a final WBAN system product. The aim of the thesis is to - on an already existing WBAN system prototype - reduce the energy consumption, make it more user-friendly and flexible and gain information of the system properties by performing tests. The thesis is divided into system tests and system development. From the system tests it has been shown that the ZigBee sensor nodes has communication coverage when the system is worn by a monitored person and that the batteries last for eight hours. It has also been shown that it is possible to perform activity classification on a monitored person wearing the WBAN system. After system development a user of the system now can give commands to the WBAN system as a step to make the system more user friendly and flexible. The commands can be given via settings placed in a file at the remote server or by sending SMS containing settings to the personal server. Furthermore, system adaption has been implemented in order to save energy by reducing the transmitted amount of data. The WBAN system has the possibility to only transmit new GPS positions if the GPS position has changed. Finally local data processing of the accelerometer data has been implemented, also with the purpose to save energy by reducing the transmitted data.

Contents

Chapter 1

Introduction

This report is the result of a master thesis of which content is tests and further development of a wireless body area network (WBAN). The tests and development has been performed on a previously developed WBAN prototype. The prototype is a part of the TRIS project at Information Systems at FOI in Linköping.

1.1 FOI

The Swedish Defence and Research Agency (FOI) is operating under the ministry of defence in Sweden and is one of the leading research institutes in the defence and security area in Europe. The head office is in Kista in Stockholm, but the agency also has offices in Linköping, Umeå and Grindsjön outside Stockholm. During 2001 FOI merged from - at the time existing - agencies, National Defence Research Establishment (FOA) and Aeronautical Research Institute (FFA). FOI has about 1000 employees, of which 800 are research workers [1].

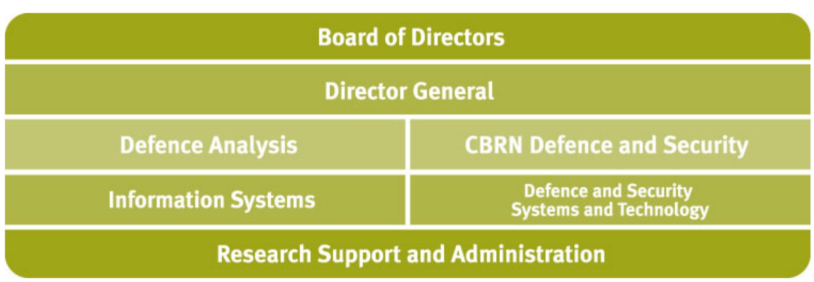

Figure 1.1: Organization of FOI

The organization of FOI consists of five divisions, of which four are research divisions within a specific area of expertise. The fifth division is a support division and carries out tasks such as administration, IT and communications. The organization can be overviewed in Figure 1.1.

1.2 Purpose of the TRIS project

The purpose of the TRIS project at FOI is to investigate the possibilities of developing a system concept for a decision support system for monitoring of an individual's status. TRIS is a Swedish abbreviation of (Tids-/resurskritisk ledning med övervakning av individstatus), which is translated as Time and Resource Demanding Control with Surveillance of an Individual's Status. The system is intended to acquire, reduce, fuse and visualize information of physiological data, contextual information, and observation reports from field observers in order to support a user's decision making [2].

The users could be medical personnel for distant medical management or commanders in the military. The medical personnel could use the system in order to acquire automatic status reports of an ill person at home. As a military application it could provide status reports of soldiers before, during and after missions. The system could also be applicable to many other areas, not mentioned here [3].

1.3 Purpose and Aim of the Master Thesis

The purpose of the master thesis is to test and further develop an existing WBAN prototype with the aim to support the TRIS project at FOI in its investigation of the possibilities of developing a system concept for monitoring of an individual's status. The aim with the tests is to find system properties and features of the WBAN prototype and to support the system development. The aim with the further system development of the WBAN prototype is to; increase the flexibility, make it more user-friendly and make it more energy efficient.

1.4 Method

The prototype developed so far is declared to meet a certain amount of requirements stated by the TRIS project group. In order to make sure the prototype is able to carry out those requirements, basic system tests will take place. The tests will evaluate the nature of the collected data, if all required data is transmitted as intended and the stability of the prototype's functionality. If the outcome of some tests is faulty or unexpected, the software will be debugged to find the malfunctioning parts, then decisions can be made upon which actions that is appropriate. Further system tests will be performed as new functions are implemented in the software of the system. It will be an iterative process with system development and system tests.

1.5 Demarcations

Choices of which system functionality that will be implemented will be according wishes of the TRIS project group at FOI and thereby not be argued for in this report. The prototype's already implemented software and hardware will be used during the thesis work. The software will be altered and further developed in the C programming language and run on a Linux operating system. Furthermore, the communication system of the prototype uses the ZigBee and GSM/GPRS technologies and will not be altered during the thesis work.

Chapter 2

Theory

The technology of WBAN systems is a relatively young technology [14]. It is based on wireless sensors that - depending on application - transmit various information to a user. There are some well-known hardware and communication protocols to obtain this. The hardware, communication technology and software used in this WBAN prototype will be overviewed in the first parts of this chapter. Finally a short overview of the theory of sensor data classification is presented.

2.1 ZigBee

ZigBee is a worldwide standard specification of a suite of communication protocols which has been developed by the ZigBee Alliance [5]. The ZigBee Alliance is an association of international companies that develops the standard. The suite of communication protocols is based on the IEE 802.15.4-2003 standard for wireless personal area networks. The goal with the ZigBee wireless standard is to provide a short-range, low-power, low-rate, low-cost and secure networking technology [4].

One feature that supports the low-power consumption of the technology is that ZigBee can spend time in sleep mode most of the time in some applications. A device can enter sleep mode when there is no data to receive or transmit. The wake up delay – going from sleep mode to wake mode – is very small in ZigBee, i.e., 15mS. Particularly compared to Bluetooth which has a wake up delay around three seconds [5].

IEE 802.15.4-2003 which ZigBee has as foundation uses three different frequency bands which together are divided into 26 channels. One of the operating frequency bands is 2.4GHz and is used worldwide. The other two have lower frequencies and are specific for Europe, America and Australia [6]. The constellation of channels, frequency bands and data rates is depicted in Figure 2.1.

Figure 2.1: Frequency bands and channels for ZigBee with geographic explanation

In a ZigBee wireless network there can co-exist three different nodes named ZigBee Coordinator (ZC), ZigBee Router (ZR) and ZigBee End Device (ZED). There can only be one ZC in each network. The ZC can relay messages between devices in the network, but it is also responsible for startup and initialization of the network, select appropriate frequency channel, allow other nodes to join the network and handle security management. In comparison, the ZR has no other responsibilities but relaying messages between other devices in the network. As implicated from the name, the ZED is located at the end (or edge) of the network and it only transmits and receives messages to and from routers and the coordinator.

The real communication coverage distance for ZigBee varies depending on the application. Some factors that affect the maximum communication distance are; type of antenna, antenna gain, transmit power and background noise [5].

The ZigBee standard support three different network topologies, these are:

- Star Topology
  In this configuration, which is the simplest and most limited, nodes use the coordinator as a central hub to exchange messages. Thus, there is only one coordinator that is responsible of relaying messages between communicating nodes.

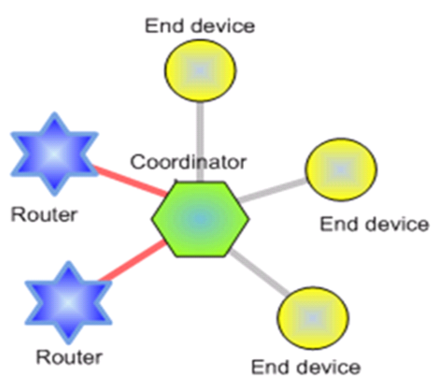

Figure 2.2: Star Topology

- Tree Topology
  In this topology, the coordinator operates at the root of the tree as a parent and can have many nodes (Routers/ End Devices) as children. At lower levels of the tree hierarchy routers and end devices can be found, where only routers can act as parents and have children of their own further down in the hierarchy. Each child node can only communicate with its parent node which, in turn, can communicate with its parent node.

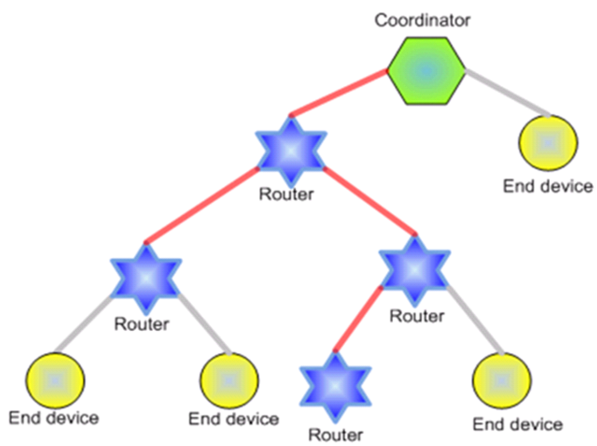

Figure 2.3: Tree Topology

- Mesh Topology
  This topology builds upon the tree topology to add more flexibility where nodes on the same level of the tree or within range can communicate directly, without the need to exchange messages through the closest ancestor. The Mesh topology gives rise to more efficient message propagation, and means that alternative routes can be found if a link fails or there is congestion.

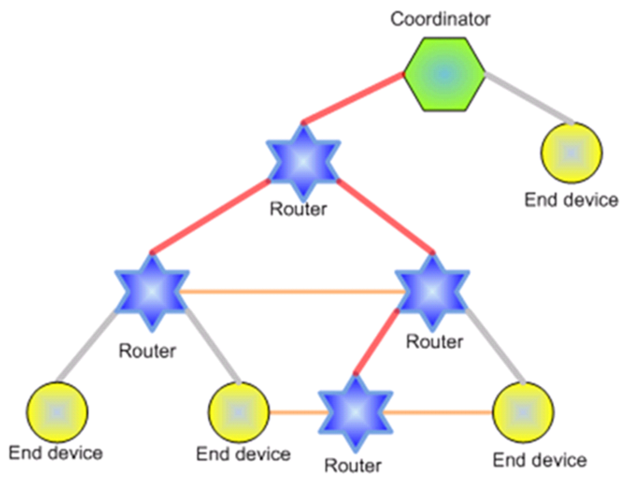

Figure 2.4: Mesh Topology

2.2 GSM/GPRS

GSM is originally the abbreviation of Group Spécial Mobile but is nowadays the abbreviation of Global System for Mobile Communications. It is the most popular standard for mobile communications with over 3 billion users across the world and is referred to as the second generation (2G) mobile phone system. The main difference between the second and the first generation of mobile phone systems is that in the 2G system signaling and voice channels are digital compared to being analog in 1G [7].

GPRS is the abbreviation of General Packet Radio Service and is a later added service to the GSM standard. GPRS is a packet based service compared GSM which only uses circuit switched services like voice calls and circuit switched data. A difference between circuit switched and packet based data traffic is which type of billing that is used. Packet based traffic is typically charged per megabyte of transferred data while circuit switched traffic is charged per minute of connection time. Another difference between the two technologies is that in GPRS there is a best-effort service while GSM guarantee a certain quality of service (QoS) [8].

There are three different classes of devices supporting GPRS, these are:

Class A: Has the ability to simultaneously connect to both GSM and GPRS services.

Class B: Has the ability to use both GSM and GPRS services but only one at a time. During a GSM connection GPRS is suspended and later resumed again automatically.

Class C: As class B but the switching between GPRS and GSM must be done manually.

Furthermore, a GPRS connection is established by specifying an access point name (APN).

2.3 GPS System

The first GPS satellite was launched in 1978 and the GPS system was operational in 1993 [9]. The system consist of over 24 GPS satellites with a medium earth orbit (MEO) of 20 000 km. There are six orbital planes with at least four satellites in each plane. The idea is that each satellite will pass over the same point two times every 24 hours and that at least four satellites will be in the line of sight (LOS) over the horizon wherever you are on the planet. This is almost true since the system has a real-time coverage of 99.9% [10].

Each satellite broadcasts two different L-band frequencies which are used when the distance to a GPS receiver is calculated by using the simple formula distance = (speed of light)*time. Four atomic clocks are used on each satellite to give an as precise time estimation as possible. To be able to triangulate a GPS receiver and obtain the correct coordinates there has to be four satellites within reach of the receiver, which will be the case 99.9% of the time as long as the receiver resides outdoor in LOS to the satellites [11].

When the position of the receiver is calculated, two different techniques can be used which are called pseudo-range and carrier-phase measurements. The most commonly used is the carrier-phase measurement which is using the two unmodulated carriers of the two frequencies that are transmitted from the GPS satellites. It is the phase difference between the signal(s) transmitted from the GPS satellites and an internally generated signal in the receiver that is measured. The phase difference is used to decide the distance from the receiver to each of the GPS satellites and when that is done the position, or coordinates of the receiver, can be triangulated [12].

The signal generated internally in the receiver is an exact copy of the carrier wave transmitted from the GPS satellite. Thereby in an ideal world, there should be no phase difference between the two signals. But there is a difference due to the Doppler Effect which Doppler shifts the signal when it propagates to the receiver [11].

Most handheld GPS devices (receivers) around the world has a position accuracy of up to 10 meters horizontally and $15 - 20$ meters vertically which is the case when absolute measurement is used, meaning one receiver and four satellites. The accuracy can be increased by doing relative measurement which is when one extra fixed receiver with a known position is used as a reference. From the difference between the position of the fixed receiver and the handheld receiver a baseline is computed. By doing this almost all error sources can be eliminated and thereby increase the accuracy [12].

The data transmitted between the GPS receiver and another technical unit, such as a computer or a mobile phone, is communicated by using the standard

NMEA 0183 or NMEA 2000 communication protocols. Most GPS receivers support the NMEA 0183 protocol. NMEA is the abbreviation of National Marine Electronics Association [13], which is an association that defines communication protocols used in marine communication technologies.

There are several different NMEA messages transmitted from GPS receivers. In the NMEA 0183 standard, pre-defined ASCII sentences, are used in a way that makes it simple to parse the content accurately. Some of the most popular sentences can be found in Table 2.1.

Table 2.1: Some of the sentences used in the NMEA 0183 standard

Sentence Name	Sentence Content
$GPRMC	Recommended Minimum Specific GPS/TRANSIT Data
$GPGGA	Global Positioning System Fix Data
$GPZDA	UTC Date/Time and Local Time Zone Offset
$GPGLL	Geographic Position, Latitude and Longitude
$GPVTG	Course Over Ground and Ground Speed
$GPGSV	GPS Satellites in View

The most widely used NMEA sentence in Table 2.1 is $GPRMC, which contains most of the useful data such as; speed, geographical position and UTC time. UTC is the abbreviation of Coordinated Universal Time which is used for civil time keeping and is a time standard based on the international atomic time (TAI). The geographical position is given in longitude and latitude and the speed in meters per second.

2.4 Prototype Solution

The prototype uses three ZigBee sensor nodes containing a three axis accelerometer each. The sensors are attached on strategic positions on the body of a person who is about to be monitored. Those strategic positions are above the elbow on the right arm, above the knee on the outside of the right leg and on the torso respectively. The positions are strategic in a sense that a remote user can see the angle of the arm, leg and torso separately and thereby find the monitored persons body position but also the combined movement pattern.

All the sensor nodes communicate with the ZigBee Network Coordinator (ZNC) according to the star wireless network topology. The ZNC is integrated into a unit called the personal server (PS). The PS is a sealed plastic box containing five modules. Those are; the ZNC, a GSM/GPRS modem, a GPS receiver, a micro processor and a package of batteries. In addition to the modules the PS also is supplied with a power switch and a magnetic switch for the ZNC.

Furthermore, the PS communicates via the GSM/GPRS modem and GSM network with a remote server (RS). The RS could be anything connected to the Internet, e.g., a laptop. At the RS the user of the system will receive status information of the person who is monitored by wearing the PS and sensor nodes. A simple overview of the system concept can be seen in Figure 2.5.

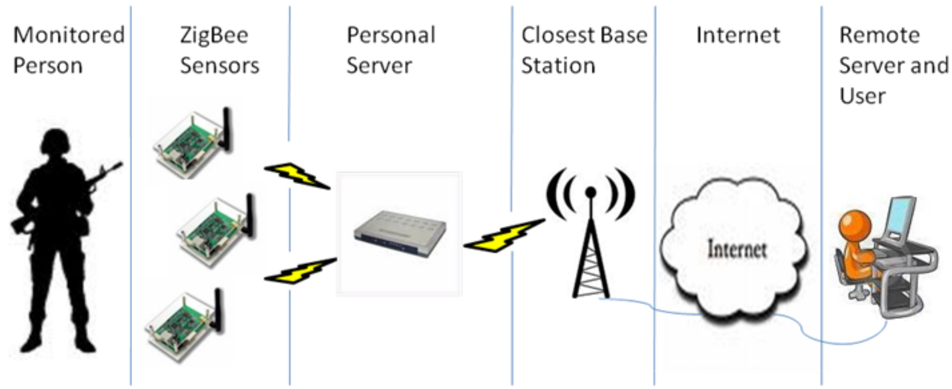

Figure 2.5: An overview of the WBAN system concept

For more information of the hardware details of the ZigBee sensor nodes, ZNC, GPS receiver and GSM/GPRS modem, please read the report "Wireless Body Area Network - Monitoring Application System: Nodes & Data Processing" by Magnus Jobs [17].

The software of the PS consist of a Linux core which run a program developed in the C programming language. The program is multi-threaded and handles an upload thread, a ZigBee data thread and a GPS data thread. The upload thread manages the following tasks:

- Establishes a GPRS connection between the PS and RS
- Creates a secure tunnel using secure shell (SSH) with key based authentication
- Transfers data from the GPS and ZigBee data threads via the secure tunnel and GPRS to the RS by using rsync.

Rsync is a free software in Linux systems that has the ability to synchronize files and directories from one computer or server to another. Some of the options that can be used with rsync are compression of files, use Remote Shell (RSH) or SSH and recurse into directories.

The ZigBee data thread manage the following tasks:

- Read messages (data packages) from the ZNC device through a serial port
- Parse header info from the data packages
- Identify and remove invalid messages

- Save data into files and time stamp the file names

The header of the packages contain information of which sensor node and which axis the packet is a part of. This information is also added in the filename together with the time stamp in order to make it easy for the user to collect the appropriate files at the RS. For example, if the user wants to use the files as input to a sorting algorithm in a graphic presentation program.

The GPS data thread manage the following tasks:

- Read GPS NMEA 0183 sentences from the GPS receiver through a serial port
- Collect the most widely used NMEA sentences named $GPRMC
- Parse the collected NMEA sentences, convert them into binary format, and store the data into time-stamped files

Furthermore, the program reads configuration settings from a configuration file placed in the file-system of Linux on the PS. Some of those settings are; which host that is the RS, which network operator that will provide Internet access and upload directories. Those settings can be changed by connecting the PS to a computer via USB and then download a new configuration file.

Added thread functionalities and settings have been implemented upon this software structure during this master thesis, of which can be found in section 4 on page 30. For a more in depth understanding of the software, parsing or serial port programming, please read the report "A Wireless Body Area Network System for Monitoring Physical Activities and Health-Status via the Internet" by Bestoon T. Hussain Jaff [18].

If a user of the WBAN system wants to configure or further develop the software of the system, the configuration setup of WBAN prototype in appendix A can be used as a support. For a better understanding of the software's communication with the GPRS modem, read the modem connection chat content in appendix B.

2.5 Accelerometer Data Classification

An accelerometer is a device that measures acceleration forces. These forces may be static, like the constant force of gravity, or they could be dynamic caused by movements or vibrations acting on the accelerometer. Acceleration is quantified in the SI unit meters per second per second $\left[m/s^2\right]$.

Earth's gravity, denoted g, refers to the acceleration that the Earth conveys to objects on or near its surface. It has an approximate value of 9.81 m/s^2. For a stationary object resting on the Earth's surface there will be an identical but opposite acceleration, i.e., directed upwards.

Acceleration is often expressed as a number of Gs. This G-value is obtained by dividing the measured acceleration with the Earth's gravity of 9.81 m/s^2.

This means that the Earth's gravity results in a +1Gz acceleration on an object resting on the Earth's surface.

A three axis accelerometer has the ability to return real value estimations of the acceleration along the x, y and z axes. Each axis only measure the acceleration in two directions, up and down. For example, if an accelerometer is standing still with the z-axis pointing straight upwards, then the theoretical value for the z-axis would be +1G. In turn the x and y axes would have the values 0G since those axes are perpendicular to the z-axis and do not detect any g-forces sideways.

Accelerometers are widely used in different areas of society. Some of those areas are; gaming controls, smart phones, and spacecrafts. A relatively new area is to use three axes accelerometers on persons in order to recognize a person's body activity. This is possible thanks to recent advancement in micro sensor technology and low power wireless interfaces [15]. Accelerometer classification is a step in the efforts to recognize user activity from accelerometer data and is a part of the framework context awareness [16]. The activity could be, e.g., that the person is lying down, sitting, walking or running.

Data processing of the raw data from the accelerometers is needed when the classification take place. Some of the most useful features to extract from the raw data are:

- Mean
- Standard Deviation
- Energy
- Correlation

The mean can be used to find the DC component of the different axes of the accelerometer and thereby find the user body position. Different movements such as standing still or running have different ranges of possible acceleration values. Standard deviation is a good measure of how big this range of acceleration values is. Energy is useful when activities have different periodicity and can be calculated by taking the sum of the squared discrete Fast Fourier Transform (FFT) component magnitude of the signal generated by the data stream from the accelerometer. In turn, the correlation is useful to differentiating activities that involve translation in one dimension [19].

The data processing involve that calculations based upon a given sample window size are performed. The window size differ depending on the time period over which the activity is about to be classified and on what sample rate the accelerometers use. It has proven to be efficient to use a window sample size representing a time period of five seconds [19]. The energy would be:

$$E = \frac{\sum_{i=1}^{ws} |x_i|^2}{ws} \tag{2.1}$$

Where ws is the window size and x_i are the time samples of the window.

When the different calculations have been taken place for each of the axis and accelerometers, an activity recognition algorithm is needed. Each activity should have a corresponding signal pattern which can be studied in order to learn the signal pattern. This pattern can then be recognized by using the mentioned calculated features implemented in an activity recognition algorithm.

Chapter 3

Tests

System tests is performed as an initial phase of the master thesis with the purpose to find bugs in the software and to find the data rate of the system. To support further development of the system, to learn the nature of the data and to find system properties and features, more tests have been performed during the thesis work. These tests are divided into five parts; initial system tests, system properties tests, GPS tests, temperature tests and accelerometer tests, which will be overviewed in this chapter.

3.1 Initial System

After initial system tests it is noticed that the system has data transfer interruptions. It is decided that monitoring of the data flow is needed in order to understand where the problem is. A system test is performed with two ZigBee sensors activated, lying on a Table close to the PS. To monitor the data flow, the PS is connected to a computer via USB. From the output of rsync the amount of transmitted data and the time between several synchronizations (transmissions) are studied. In Table 3.1, the outcome from the test can be seen.

Table 3.1: Amount of data transmitted with two ZigBee sensors activated

Sync. No.	Data Send [Bytes]
1.	13377
2.	24625
3.	36048
4.	48759
5.	50036

It can clearly be seen that the amount data transmitted each time constantly increases with about 1200 bytes. This causes the system to constantly extend the transmission time and after a while, unexpectedly interrupts the data transfer. This explains why there is a loss of data. A second test with only one sensor is performed to see if the behavior still is the same. In Table 3.2, the outcome from the second test can be seen.

Table 3.2: Amount of data transmitted with one ZigBee sensor activated

Sync. No.	Data Send [Bytes]
1.	14816
2.	15030
3.	14000
4.	14926
5.	14300

When only one sensor is used it can be seen from Table 3.2 that the size of the transferred data is kept at a relatively constant level and no disruption of data occurs. One may draw the conclusion, as could be seen from Table 3.1, that two or more sensors implies an increased amount of transmitted data for each transmission. This causes the system to become overloaded after a period of time and thereby interrupts the data transfer. The question is if it is the amount of data that is making the transfer slow or if it is something else specific for how rsync operates. Further understanding is needed before any conclusions can be drawn.

It is known that rsync synchronize files and/or directories between two computers or servers. At the PS plenty of small data files are created locally before they are synchronized, i.e., transmitted to the RS. At this point it is wanted to understand how the amount of data and number of files respectively affect the transfer time. It is decided to perform two tests. One of how long the transfer time is for a specific amount of data contained in two files. And a second of how long the transfer time will be for the same amount of data but with file sizes the same as the size of the files created at the PS. The amount of data is chosen to be quite large in order to get a better estimate of possible differences in transmission time between the two tests. The results turned out to be as shown in Table 3.3.

Table 3.3: Transfer test with 750 kB (kilobyte) of data with varying amount of files.

Number of Files	Up Speed [kB/s]	Down Speed [kB/s]
2	9.20	7.40
5697	0.95	0.97

As one can see, increased number of files affected the transfer time in a negative way. The transfer time increased with approximately eight times. The conclusion is that rsync has two parts that affect the transfer time; the amount of data and the number of files. Rsync synchronizes one file at a time and that takes equally long time for each file independently of the file size. Hence it is decided to compress all GPS and ZigBee files into two libraries (.tar.gz) before transmission. Tests has shown that this solution has solved the problem with increasing transfer times and now the WBAN system is working without any disruptions.

3.2 System Properties

Interesting system properties that one would like to know are; what is the average transmission time from the PS to RS, what is the maximum communication coverage distance between the ZigBee sensor nodes and the PS, what is the relation between the amount of collected GPS and ZigBee data and for how long will the system transmit data to the RS? To find these properties it is decided that several properties tests should be performed and of which now will be presented.

3.2.1 Transmission Time

It would be interesting to know how long it takes for the created data files on the PS to reach the RS, especially for the user of the system. In this case, transmission time means the time it takes between each synchronization. That includes how long it takes for rsync to finish its compression, synchronization and secure transfer of files. Notice that the existing software constellation has a periodic transfer (synchronization) of files, not a continuous.

To get an accurate estimate of the time period between each synchronization, the output from rsync will be studied. This is done by connecting the PS to a computer via USB and log on to the Linux system of the PS and then run the WBAN software in a terminal. Since rsync is a software part of the Linux system it will give continuous output in the terminal. Among other things it will indicate when a synchronization starts. The time between the start of each of 20 synchronizations is documented and then the average time period between the synchronizations is calculated.

The result shows that most of the synchronizations take about 22 seconds, but two of the synchronizations take 38 and 54 seconds. However, the average time period between each synchronization is calculated to 25.5 seconds.

3.2.2 Coverage

One crucial factor in the WBAN system is the communication coverage distance between the ZEDs and the ZC. If one ZED drops the connection with the PS, no data from that ZED will be transmitted to the RS. With the current functionality of the ZC it only accepts connections the first 30 seconds from when it is switched on, so if a ZED drops connection after 30 seconds it can not re-connect again. The maximum coverage distance was previously unknown for the WBAN system but were investigated by doing some tests.

The first test is performed indoor with the PS and one ZED lying on a long Table. The ZED is in line-of-sight (LOS) with the PS. When the system is started the ZED is lying very close to the PS. The ZED is then slowly moved further away from the PS and when data stops to arrive at the RS, it is obvious that the ZED has dropped its connection with the ZC.

The test indicates that the ZED drops connection with the PS (or ZC) already at 2.5 meters. It is a short distance but should be enough for this application where the ZEDs will be placed together with the PS on a person, which probably will give a maximum communication distance of about 1.5 meters. But what will happen if the PS is placed in a bag on the back of a person while the ZEDs are attached on the arm, leg and torso respectively? The communication distance will not be over 2.5 meters but now it would not be a LOS environment anymore. To further investigate the coverage of the system, it is decided to reproduce the bag-scenario in a live test. The test would also represent the most probably future scenario of the WBAN system; that the PS is enclosed inside a textile cover somewhere on the body, with body parts in the way between the PS and ZEDs.

The test is performed by first having the PS and ZEDs in LOS at start up and staying so in five minutes. Then - when the system is running - the PS is put in a bag and placed on the back of the test person and the ZEDs are attached on the arm, leg and torso respectively. Then the test person is walking in circles inside a room during five minutes before the system is shut down again. Afterwards the time stamps of the received files at the RS are checked to see if data has been received from all three ZEDs during the test. The result show that files with data was received from all three ZEDs during the 10 minutes the system was running. Thereby it has been shown that the WBAN prototype has communication coverage between the ZEDs and the PS in its specific field of application.

3.2.3 Data Relationship

The amount of data transmitted from the PS to the RS will always be a hot topic since the GPRS communication is the most energy consuming part of the WBAN application. For this reason, it is wished for to keep the amount of transmitted data as low as possible. Therefore it is also interesting to see if it is the GPS or the ZEDs that generate the most of data. If it is known where most of the data comes from one can focus on to reduce the amount of data from that part. But if both the GPS and ZEDs produce equally amounts of data, then it is equally worth to reduce the data from both.

When the relation of the amount of data from the GPS and ZEDs is to be tested, it is decided to collect data from all three ZEDs and GPS outdoor with full GPS connection. It is a 10 minutes test with full data collection capacity.

Afterwards the amount of ZigBee and GPS data is compared and it is shown that the accelerometer and temperature data from the ZEDs represent 98.5% of the totally amount of transmitted data to the RS.

3.2.4 Endurance

An endurance test has the purpose of testing for how long the system can transmit data to the RS. The configuration of the test is simply that all three ZigBee sensor nodes are lying still on a Table, close to the PS. The PS and ZEDs are loaded with fresh batteries. Then the PS is switched on and is tested for how long the PS is transmitting data to the RS. It is also expected that the data from the accelerometers should be stable over time since the sensors has been lying completely still. The result from one of the sensors can be seen in figure 3.1.

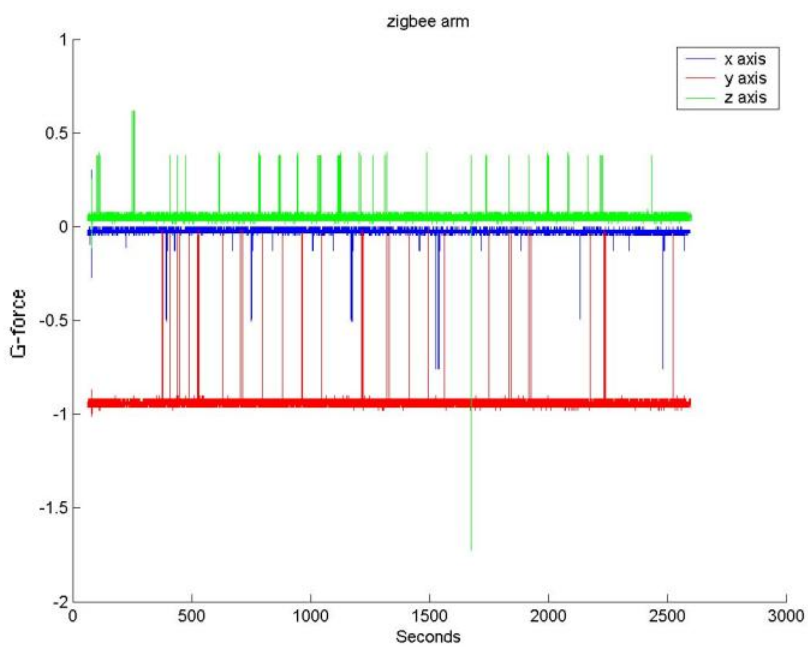

Figure 3.1: Endurance test of a ZigBee sensor node lying still on a table with the y-axis face down

It can be seen from Figure 3.1 that the PS has transmitted data from the sensor approximately 8 hours. The other two sensors had data transmitted for an equally long time. It was shown that it was the batteries of the PS that only lasted for 8 hours.

Another interesting result from the endurance test is the, what it seems, random appearing peaks in the data flow of the sensor in Figure 3.1. The amplitude of the peaks seem to be consistent for the y-axis but more random for the x-axis so it is hard to find a pattern. In the ideal case there should be no peaks at all for a sensor lying still on a table. Obviously an error has been found. In Figure 3.2, one of the peeks on the y-axis has been zoomed in. One can see that the pulse has the shape of a square pulse.

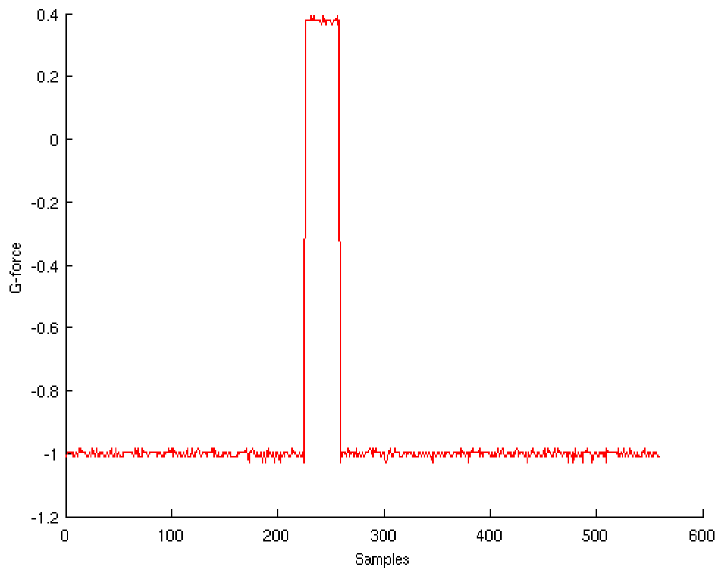

Figure 3.2: An unwanted square pulse in the sample data of an accelerometer that has been lying still on a table.

If calculations are about to be performed on the sample data, it is obvious that the unwanted square pulses will affect the result. This is especially true if one would like to calculate the fast Fourier transform (FFT) for a small given sample interval containing one of the square pulses, which would give an erroneous picture of reality. During debugging of the system it is found that the error occurs before the ZigBee data packet is received at the PS and is thereby not correctable during this master thesis work.

3.3 GPS

The GPS needs to be tested in order to verify that it operates as expected, i.e., that the GPS get contact with satellites, deliver the correct geographic coordinates, and to verify its accuracy.

During the first tests of the GPS the PS was inside a building. As expected the quality of the data was invalid since there was no line of sight (LOS) between the receiver and the satellites, and thereby no GPS positions were received. The second test of the GPS was set-up according to the following routine. One person is wearing the PS while walking a pre-defined round trip lasting for about 17 minutes. The round trip is performed outside and has the same start and end position. In this test there was LOS and hence GPS positions with correct geographic coordinates were received. Furthermore, the coordinates also had an approved accuracy within the theoretical maximum limit of 3 meters specified for the GPS receiver used in this prototype. But it was detected that the time stamps of the GPS NMEA messages did not match the time stamps of the PS system time. According to the GPS time stamps the round trip had only lasted for 9 minutes when it actually had lasted for 17 minutes. When the retrieved geographic coordinates were plotted on a map it could be seen that only about half of the round trip positions had been transmitted to the RS. It was concluded that a major system bug had been found and that debugging was needed.

After thoroughly debugging of the system the problem was found to be multifaceted. Firstly, the GPS data thread was set to sleep one second after each read NMEA sentence and thereby sentences was read in a slower phase than they arrived at the receiver and the serial port. When the sleep state was removed the program crashed instead. It followed that the crash was due to improper reading from the serial port that is connected to the GPS receiver. As a result of an erroneous control function of the reading from the serial port, the program tried to save data outside an arrays allocated memory, which is possible in C-programming. Hence other random data in the memory was destroyed which also caused the crash. The problem was solved by improving the programs read procedure at the serial port.

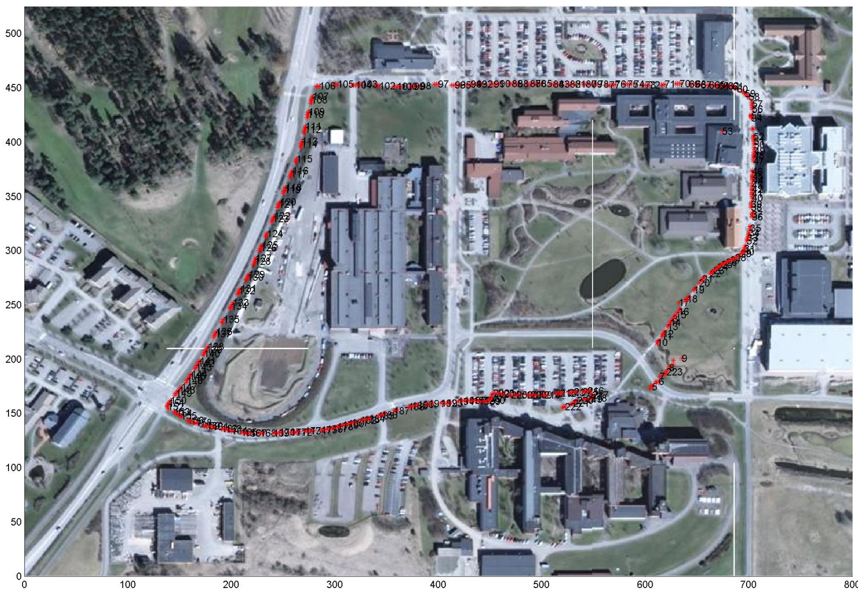

Figure 3.3: GPS round trip test at FOI and the University in Linköping

When the round trip test was performed again, the result was better. In Figure 3.3, the received geographical coordinates are plotted with red dots upon a satellite picture over the area of interest. The start and the end of the round trip have the same position and is marked with a green circle, and the round trip was anti-clockwise. As can be seen from the picture there are some positions missing in the beginning of the trip and then there are a couple of positions that are inaccurate by being misplaced about 20-30 meters eastwards. This is no error caused by the software but can be explained logically. After the WBAN system was started it took a short while before the system had initialized and started the GPS data collection which would explain the missing positions. When the GPS finally had connected properly it had not established enough satellite connections to provide accurate positions which would explain the misplaced positions. But after enough satellite connections the positions arrived accurately the rest of the round trip.

3.4 Accelerometer

Accelerometer tests are needed in order to learn how to interpret accelerometer data and classify activities when a person is wearing the ZigBee sensor nodes, containing the three axis accelerometers. The result of classification could be determination of a person's posture or activity such as sitting, walking or running.

3.4.1 Calibration

To understand and interpret the accelerometer data from a moving person, a good reference point would be to compare the data with data from an accelerometer sensor that has been lying still on a table. This way one may find possible off-sets of the sensor data which later can be compensated for. Therefore, first a calibration test is needed. In the calibration test the sensor is lying still on a flat table for a while and then rotated so the top side is down, this continuous until all sides of the sensor has been lying face down on the table for an equal period of time each. After the test, the gravitation vector is expected to be +1G and -1G for each axis. Hence the absolute value between the highest and lowest value should be 2G.

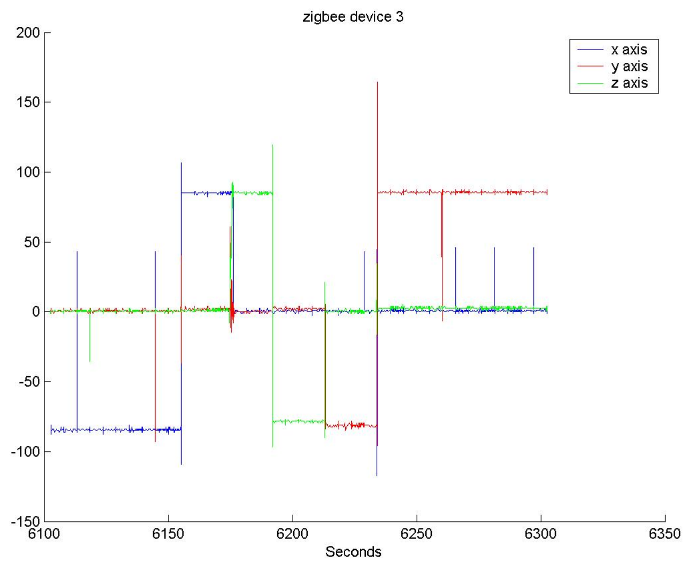

Figure 3.4: Calibration of Sensor 3 with a relative G-force scale on the y-axis. The values 90 and -90 on the y-axis corresponds to +1G and -1G respectively.

The result of the calibration test of sensor 3 can be seen in Figure 3.4. It shows that the gravitation vector is varying between +1G and -1G as expected. It can also be seen that there are peeks in the start and the end of each period that one side of the sensor has been lying face down on the table. Those peeks are going above and below +/-1G which is the result of the acceleration and retardation of when the sensor is lifted up respectively put down on the table again. This indicates that the sensor seem to have a quite good resolution which may capture even small movements of the sensor. Furthermore, the result shows a relatively small amount of noise.

3.4.2 Movement Classification

Next step in the study of accelerometer data is to investigate the nature of the data from ZigBee sensors that are moving. The idea is also to investigate the possibility of classifying sensor data as a specific body activity. This knowledge can then be used both at the RS where the user will interpret the sensor data and at the PS where the program automatically could interpret the data and then transmit this information to the user. All the movement tests will be performed with the sensors attached to a real person. The sensors are attached with Velcro tape and placed on strategic positions of the body as explained in section 2.4 on page 11.

The movement tests follow a pre-defined test protocol and one of the first tests is to collect data from a person who is standing still periodically swinging the right arm and periodically holding it still straight down. The purpose is to learn the amplitude of the G-vector and the frequency of a swinging arm. Figure 3.5 shows the result from this test.

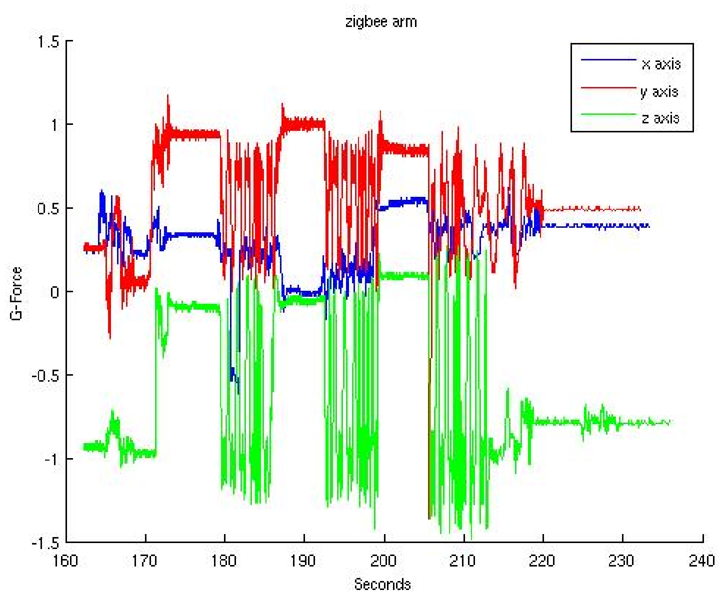

Figure 3.5: Accelerometer data from a sensor attached on the arm of a person who is standing still holding the arm periodically still straight down in 10 seconds and swinging the arm in 10 seconds, repeating this pattern three times.

As can be seen in Figure 3.5 the result show that the movement of the arm clearly can be seen. Especially if compared with data from a sensor that has been lying still as was seen in (Figure 3.4 on the preceding page).

A final full scale body movement test is performed where the data from a person who is walking and running is studied. The test has four sequences described by a person who is; walking slow, walking fast, running slow and running fast. From this test it is expected to learn how the impact of the G-vector and frequencies differ when a person is walking and running fast and slow. In Figure 3.6, the result of the test for the sensor attached to the torso can be found.

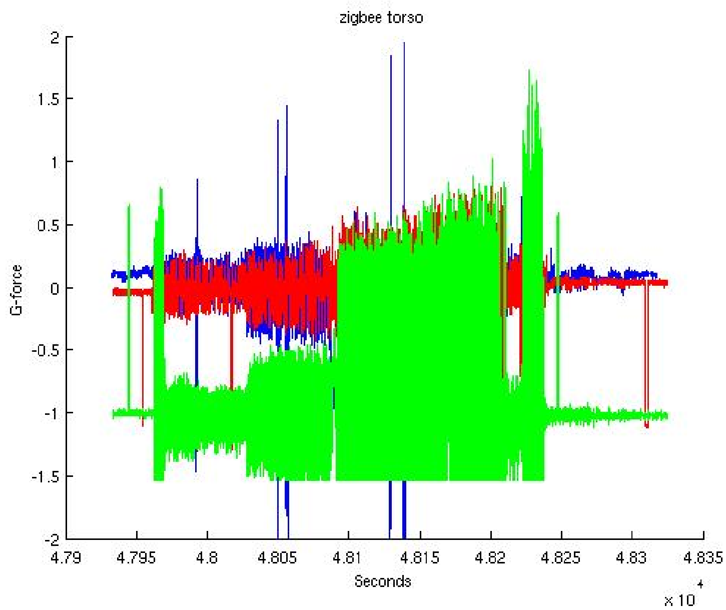

Figure 3.6: Accelerometer data from the torso sensor when a person is standing still, walking and running

A clear increase of the amplitude of the G-vector can be seen in Figure 3.6, especially for the z-axis. That is not unexpected since the z-axis is the axis that has an angle of 180^0 to the ground and is pointing upwards. Hence the z-axis easily picks up the acceleration and deceleration from when the feet hit and leave the ground. Furthermore it can be seen that no values lower than -1.5G is detected for the z-axis which is as expected since the dynamic range for the build in three-axis accelerometer is between -1.5G and +1.5G.

From the result it is clear that by calculating the standard deviation for the z-axis one could get a rough estimate of if the person is standing still, walking or running. Previous research on activity classification has also shown that by summarizing energy in certain sample windows by using Fourier analysis could give a quite good estimate of which activity that occur. In Figure 3.7 the y-axis represents the summarized energy in small sample windows from a Fast Fourier Transform (FFT) for an appropriate chosen frequency which is shown together with the raw accelerometer data that the calculations have been performed on. It can be seen that the summarized energy is as lowest when standing still and increasing with increased activity. This indicates a good possibility for activity classification by looking at the energy for a certain frequency.

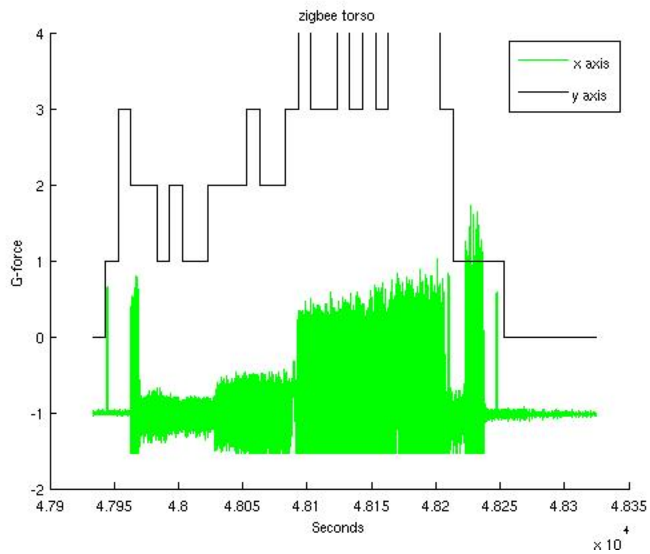

Figure 3.7: Accelerometer data and summarized energy from FFT for the z-axis of the torso sensor.

3.5 Temperature

The temperature sensors are put inside the ZigBee sensor nodes. The temperature data is not given in Celsius or Fahrenheit, but has a relative scale. To interpret this scale one need to study the data for different known temperatures. It is also of interest to study how fast and how much the temperature data is changing. It follows that temperature tests are needed.

One initial test is to study the temperature data when a temperature sensor resides in a relatively constant room temperature. From this test it is expected that the temperature data should be kept on the same level during the whole time interval of the test. Possible variations of the received data will give a good estimation of how noisy the temperature data is. Figure 3.8 shows the result from this test.

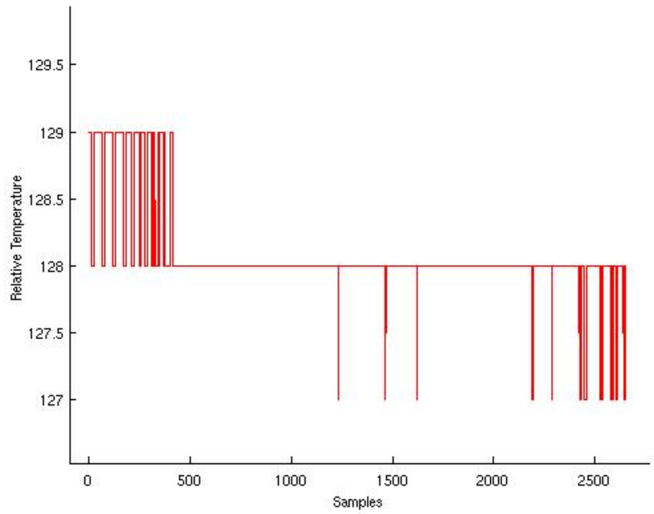

Figure 3.8: Constant indoor temperature measurement during five minutes.

The nature of the temperature data when a sudden change of temperature occur is also tested. This is tested by collecting data from a sensor that suddenly is taken from indoor room temperature, $+23°C$, to an outside temperature, $-5°C$. Figure 3.9 shows how the temperature data is changing during five minutes.

Unexpectedly it can be seen that the temperature data in Figure 3.9 is increasing. The conclusion from this must be that either the temperature scale should be inverted or that the temperature sensor actually measures the micro temperature from the small environment around the ZigBee sensor node's microprocessor. But since the temperature was stable during the indoor temperature test with constant temperature, one can conclude that the temperature scale should be inverted.

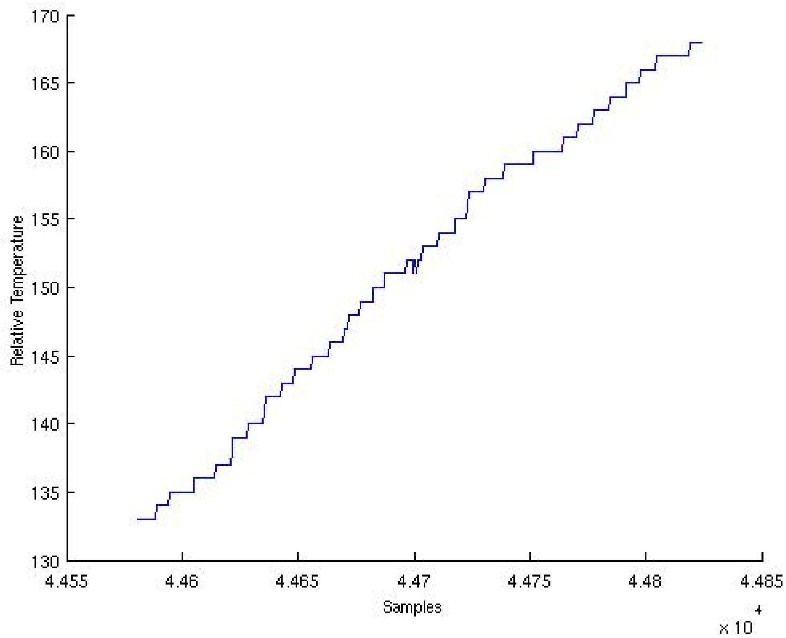

Figure 3.9: Temperature test of a room tempered sensor lying on the ground outside during five minutes in a temperature of $-5°C$.

Furthermore, the indoor- and the outdoor tests show initial temperature values of 128 and 133 respectively. In both cases these values should represent a room tempered environment. As a consequence of the inverted temperature scale, the higher initial value in the outdoor test would indicate a lower initial temperature. This could be explained by that the test did not start until after the sensor had been outdoor for about one minute and thereby already had lowered it's temperature.

Chapter 4

System Development

Further system development was requested for the WBAN system. The further development should support the aim to lower the energy consumption and increase the flexibility and user friendliness of the WBAN prototype. To increase the flexibility and user friendliness it has been chosen to implement two kind of manual settings for the user of the system; manual settings via RS and manual settings via SMS. In order to lower the energy consumption, automatic system adaption and local data processing is implemented. These implementations will be explained respectively further in this chapter.

4.1 Manual Settings via RS

The first version of the WBAN prototype collected all data from the GPS and ZigBee sensors and then continuously transmitted it to the remote server. Since energy consumption is needed to be kept as low as possible, it is logical to keep the transferred amount of data as low as possible as well. For example, if the user only is interested in the GPS position, then it is unnecessary to collect and transmit all ZigBee data. If there was a possibility for the user to choose which kind of data that is needed at the moment, then the energy consumption of the system would be reduced a great deal. But there would also be a flexibility for the user to operate the WBAN system and periodic collection of data would be possible. Therefore an implementation of manual settings for the user was asked for.

Since the system already uses rsync to transmit data from the PS to the RS, it is decided that rsync also should be used for downloading of user settings from the RS to the PS. The settings will be inserted into a configuration file with the same format as the one with standard settings placed on the PS, with the difference that the user can change the settings of the file at the RS. The PS will then periodically check for changes in the configuration file at the RS and if changes are detected, the PS will download the new settings and use them until the settings change again. It was decided that the downloading of settings will occur periodically with a time interval decided by the setting “RsSettingsTimeInterval”.

The periodically downloading of the configuration file is, a little bit contradictory, implemented into the “data upload thread”. It could also have been implemented into the “GPS thread” or the “ZigBee thread. But it was reasoned that the GPS and ZigBee threads should be kept as data collection threads only in order not to loose any data transmitted from the GPS or ZigBee sensors. Another reason why the upload thread was chosen is that it is assumed that the periodicity of the downloading of new settings will be much less frequent than the uploading of data. Thereby the data upload thread still has the main purpose to upload data to the RS and no loss of data will occur since the GPS and ZigBee threads store the data in files locally at the PS until the upload thread has time to transmit them.

Some of the most important settings - according to the TRIS project group - have been implemented in this version of the prototype, but further settings can easily be implemented in the future. The implemented settings with explanations of what it sets and what its parameters are explained in Table 4.1.

Table 4.1: Manual settings for data collection

Setting	Parameter	Explanation
DataCollection	1	Collect All Data Possible
DataCollection	2	Automatic System Adaption
DataCollection	3	Manual Collection of Data
RsSettingsTimeInterval	ss	Settings download interval [s]
GpsCollectionOn	yes/no	Collect GPS Data
TransmitGpsQuality	yes/no	Transmit GPS Quality
TransmitGpsPosition	yes/no	Transmit GPS Position
TransmitGpsHeight	yes/no	Transmit GPS Height
TransmitSensor1	yes/no	Transmit Data from Sensor 1
TransmitSensor2	yes/no	Transmit Data from Sensor 2
TransmitSensor3	yes/no	Transmit Data from Sensor 3
TransmitSensorTemperature	yes/no	Transmit Sensor Temperature
TemperatureSampleRate	ss	Temperature sample time [s]
GpsPositionThreshold	nmi	Nautical Miles [NM]
ZigBeeSampleWindow	samples	Number of Samples
RsSettingsTimeInterval	ss	Seconds Between Downloads
SmsSettingsTimeInterval	ss	Seconds Between SMS Read

The setting “DataCollection” is the one that decides which type of data collection the PS will use. Manual collection of data, i.e., parameter 3, will give the user the possibility to manually decide which data that should be collected.

Parameter 2 of DataCollection will be explained further in chapter 4.2. The parameters of the settings are changed by the user in the "configuration file" at the remote server. The remote server could be any computer connected to the Internet, e.g., a laptop.

4.2 Manual Settings via SMS

To increase the flexibility for the user in the control of the WBAN system it is decided to implement a SMS application in the software of the PS. The vision is to be able to SMS the PS with new settings instead of changing the settings in the configuration file at the RS. All of the settings that can be set at the RS should also be possible to set with a SMS.

Since the PS is equipped with a GSM/GPRS module - with belonging subscriber identity module (SIM) slot - it is possible to send SMS to that module as long as the phone number is known. Hence, no additional hardware is needed when the SMS application is implemented.

It came naturally to implement the SMS functionality in the upload thread of the program on the PS since the upload thread also handles the downloading of new settings at the RS. The danger with this constellation is that if new settings are downloaded and new SMS are checked for very often, then rsync which handles the uploading of data, do not have much time to transmit new data to the RS. This could imply that data is buffered up at the PS and that rsync work slow since there is much data to transmit.

When the SMS functionality of the WBAN system has been implemented, Hayes AT commands has been used in the communication with the GSM/GPRS modem. Hayes AT commands are a common communication language for modems, which consist of a series of short text strings combined together to produce complete commands that the modems understand. In this implementation the AT commands are inserted into an already existing chat file that is used by the point to point protocol daemon (PPPD) in Linux.

In the prototype of the WBAN system the chat file only handles all the necessary AT commands needed to set up a GPRS connection between the PS and RS. In this updated version of the prototype, AT commands needed to read received SMS at the GSM/GPRS modem are added into the chat file. The AT commands used for the SMS section of the chat file can be overviewed in table 4.2.

Table 4.2: Hayes AT commands used in the SMS section of the modem chat file.

AT Command	Explanation
AT+CMGF	Put the modem in text mode.
AT+CMGL	List all SMS in memory.
AT+CMGD=1,4	Delete all SMS in memory

The modem's responses to the AT commands are saved into a chat log file. From this follows that also any newly received SMS will be found in the chat log file since the content of the SMS will be a part of the modem's answer to the command AT+CMGL. Thereby this log file is read by the WBAN program with the purpose to look for any new settings transmitted by the user of the system in the form of a SMS. Each time the program look for new SMS, the log file is read. Directly after the content of the log file has been extracted, the log file is deleted in order to not accidentally read old settings next time it looks for new SMS. The log file will be re-created with new answers from the modem next time the chat file is run by PPPD

4.3 Automatic System Adaption

Automatic system adaption is an alternative to the manual settings of data to be collected. If the setting DataCollection has the parameter = 2, the system is set to automatic system adaption, which implies that the system becomes intelligent and can adapt itself. Automatic system adaption means that the system by itself continuously takes decisions upon which data that should be transmitted. The idea is that if nothing new happens, then nothing is needed to be transmitted. This way the system saves energy in an effective way since the transmission of data is regarded as the most energy consuming part of the process [17].

In order to obtain system adaption, data processing is needed. The PS need to keep track of changes in the incoming data and then take decisions upon what to do, e.g., if the changes is noise or if it is something new that should be transmitted to the RS. As a proof of concept it is decided that simple system adaption should be implemented for detection and transmission of relevant change of the GPS position. The term relevant in this case means that the system should be able to distinguish real changes from changes caused by noise or inaccuracy.

One further step to reduce the amount of transmitted data from the PS to the RS is to only transmit a GPS position if it is a new position. It means that if the monitored person stands or is lying still, no GPS positions should be transmitted. This should be detected automatically by the system and in order to implement detection of a relevant change of GPS position, it is required to find a simple GPS position update algorithm. The algorithm implemented during this thesis work has a straight forward approach. When the system is started the algorithm will save the current GPS position and it will be transmitted to the user. The algorithm will not update the position and thereby not transmit a new GPS position to the user until the monitored person has moved a further distance than a given threshold value.

The GPS position threshold is implemented as a new setting of the system. This setting is convenient when the user of the system wants to set the resolution of the GPS positions. For example, if the user is not interested in detailed position changes but rather want to know if the monitored person is leaving a certain area. Maybe the user only is interested if the person has moved more than 100 meter or one kilometer from the latest position. But if the user wants to detect small changes of GPS positions, (more than three meters), then a threshold of 0.0015 can be used which in longitude and latitude is number of nautical miles (NM) also called distance-minutes. One nautical mile is defined as 1852 meters, which give 0.0015 NM = 2.77 meters. Keep in mind that this could bee a too small value since the GPS has an accuracy of 3 meters.

4.4 Local Data Processing

In this updated version of the WBAN prototype, automatic collection of data also implies that the PS perform simple calculations on the raw data received from the ZigBee nodes. The idea is that if some data processing could be done locally on the PS before transmission, then the amount of transmitted data will be reduced. This is also a part of the efforts taken to reduce the energy consumption of the WBAN prototype which would be the case if the amount of transmitted data is reduced. It is decided that the data processing will result in the variance, mean, min and max values for the accelerometer and temperature sensors for each ZED. These values will be calculated over a given “window sample size”. The window sample size is a setting given by the user of how many samples that should be collected before the mentioned values are calculated.

The mean is calculated according to:

$$\upsilon = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{s}{N} \tag{4.1}$$

Where υ, N, i and s respectively are the mean, total number of samples, the present sample and the sum of the values of all samples. In turn the variance is calculated according to:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \upsilon)^2 \tag{4.2}$$

Where σ^2 is the variance. From equation 4.2 follows:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} \left(x_i^2 - 2x_i\upsilon + \upsilon^2\right) = \frac{s^2 - 2s\upsilon + N\upsilon^2}{N} \tag{4.3}$$

Where s_{i^2} is the sum of all the squared samples. Equation 4.1 and 4.3 gives:

$$\sigma^2 = \frac{s^2 - 2s\upsilon + N\upsilon^2}{N} = \frac{s^2}{N} - 2\upsilon^2 + \upsilon^2 = \frac{s^2 - \upsilon^2 N}{N} = \frac{s^2 - s\upsilon}{N} \tag{4.4}$$

This gives a simple expression to use since then all samples don’t need to be stored in memory. Only the sum and the total number of samples are needed.

Hence, equation 4.1 and the right hand side of equation 4.4 is implemented in the data processing part of the software where the mean and variance respectively are calculated.

The ZigBee data is transmitted in form of ZigBee data packets by the sensor nodes and is received by the ZNC at the PS, which in turn forwards the packets to a serial port. The ZigBee thread of the program continuously checks for new data at this serial port. Each packet arrived at the serial port consist of a header and a dynamic range of samples from the accelerometers in the sensor nodes. The ZigBee packet header contains information of which ZigBee sensor node and axis the samples in the packet belongs to. This information is used when state information is saved locally into memory. The vector $\vec{A}$ holds state information of all 12 ZigBee sensor nodes and axis combinations. The number of the ZigBee sensor node and the number of the axis are used when the appropriate position for the state information is wanted from the vector. The correct data position is reached according the following vector position algorithm:

$$\overrightarrow{A}(n) = \overrightarrow{A}(dID + sID * N_{sID} + slot * N_{sID} * N_{slot}) \tag{4.5}$$

Where:

- $\overrightarrow{A}(n)$ = The value of position n in vector $\overrightarrow{A}$
- dID = Which sensor node (device ID) the data belongs to
- sID = Which axis (sensor ID) the data belongs to
- $slot$ = Which data slot the data belongs to
- N_{sID} = Total number of axis per device (Constant value, currently = 4)
- N_{slot} = Total number of data slots (Constant value, currently = 6)

The data refer to which value that is about to be saved into, or collected from the vector and is distributed according the structure in Table 4.3.

Table 4.3: Content of the data slots in the $mainVector$

Slot Number	**Data Content**
0	Start Position
1	Values Ready
2	Min Value
3	Max Value
4	Sum
5	Number of Collected Samples

Slot number 0 and 1 are used by the sample reading algorithm of the program, while slot 4 and 5 are used to calculate the mean and the variance according to the equations 4.1 and 4.4. The mean and variance together with the min and max values contained in slot 2 and 3 will then be written to file and transmitted to the RS. Then the values in the vector is reset to zero and the process restarts. The process of the data collection is depicted in Figure 4.1.

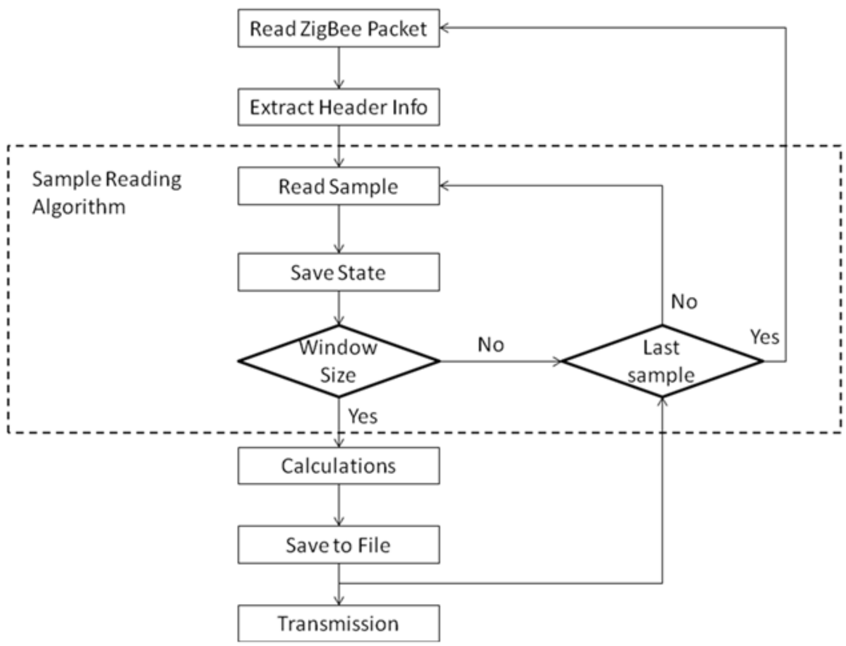

Figure 4.1: Flow chart over the ZigBee data collection process in automatic data collection mode

In order to not miss any samples, a simple “sample reading algorithm” has been developed and is a part of the data collection process in Figure 4.1. The algorithm read samples until the sample window size is reached or until the sample buffer is empty. According to design requirements, the sample window size is dynamically set by the user and is one of the settings in the configuration file. Even if the user is able to choose any sample window size in the range one up to infinity, it is advised to choose a window size that represents a time interval in the range one second up to one minute. A time interval shorter than one second will give inaccurate statistical calculations of values such as the mean and standard deviation. The reason is that the population (number of samples) is not large enough. On the other hand, a time window longer than one minute will give a lower resolution and maybe not capture some body activities performed by the person who is wearing the WBAN system. Recall from chapter 2.5 the effective time period of five seconds.

The content of the file with the saved values will have the following predefined format:

$$< mean >, < \sigma^2 >, < min >, < max >$$

For parsing purposes at the RS, the fields have "," as a delimiter. And in order to facilitate the interpretation of the received data at the RS, it is decided that the file name will contain information of which ZigBee sensor and axis the data belong to by having a file name according the following format:

$$< ZigBeeSensor >< Axis > \_ < WindowSize > \_ < date >< time > .hex$$

To verify that the local data processing actually has reduced the amount of transmitted data to the RS, a final test is performed. It is chosen to set a ZigBee sample window representing five seconds of samples since that is a good sample interval for activity classification. The result shows that the accelerometer data now represent 75% of the total amount of transmitted data, compared to 98.5% as it was before. That implies that the accelerometer data has decreased with about 16 times. Hence, the local data processing has proven to be efficient at reducing the amount of transmitted data.

Chapter 5

Discussion

Recall that during the system properties tests, several useful system features were found. One of the results was that the system has communication coverage between the PS and the ZEDs when the system was attached on the body of a person about to be monitored as intended for this application. But one should be careful when interpreting the result of this test. One should remember that the test only was performed once and that the system only had coverage up to 2.5 meters in a LOS environment. In another situation on another person and in another environment, the test may show another result.

On the other hand, one test that should be reliable is the one where the amount of transferred data was measured and when the relation of the amount of data between the GPS and ZigBee sensors was calculated. There could of course be less collected data if the automatic data collection is activated and the person stands still which would cause the system to not transmit any new GPS positions or if one of the ZigBee sensors drops connection with the PS. But as long as the WBAN system only uses three accelerometers, it will not transmit more data then was measured during the test.

Furthermore, it seems that by doing an FFT of the accelerometer data samples, it is possible to perform activity classification by looking at the total energy over certain intervals. At least this classification should be possible for the standing still, walking and running scenarios. In combination with the mean values it should also be possible to detect which body posture the monitored person has. For example if all three z-axes of the three accelerometers show a G-vector that has a mean value of zero and an energy equal to zero, then the user of the system knows that the person is not moving and that the leg, torso and arm is horizontal to the ground. Hence, the user can draw the conclusion that the monitored person is lying down.

When the manual settings part of the system was implemented, the functionality was implemented in the upload thread of the software. One may argue that the downloading of new settings from the RS and the checking for new SMS can cause data loss since the upload threads main purpose – to upload data – is disrupted. Data loss could maybe occur if the two settings; rsSettingTimeInterval and smsSettingTimeInterval is set to very low values which could cause the upload thread to check for new settings at the RS and check for new SMS as often as it uploads data. But by experience during the thesis work, it has been shown that if the time intervals is not lower than five minutes, the uploading of data will not be affected much at all. Maybe a minimum allowed time interval for the two manual settings alternatives should be implemented in the system to prevent any potential future problems.

Chapter 6

Summary

Some of the most severe bugs found during initial system tests caused the program to crash during run-time and thereby disrupted the transmission of data between the PS and RS. After debugging and fix of bugs, the WBAN prototype is able to run and transmit all data to the RS without disruptions, up to eight hours when the PS run out of batteries.

During the system properties tests some interesting features have been extracted. It has been shown that the maximum coverage distance between the ZigBee sensor nodes and the PS is 2.5 meters in an LOS environment. The coverage is also sufficient in a non-LOS environment where the WBAN system is attached on a body.

Furthermore, the tests have shown that the accelerometer data stands for 98.5% of the total amount of transmitted data from the PS to the RS. Since the GPRS communication between the PS and RS is the most energy consuming part of the system, this indicates that efforts taken to reduce the transmitted accelerometer data will pay off well in energy preservation purposes. The implemented local data processing of the accelerometer data implies that the accelerometer data has decreased with 16 times for five seconds sample intervals and thereby saves much energy of the WBAN system.

The movement tests with the accelerometers have shown that activity classification is possible. The activity classification by using FFT on small accelerometer sample intervals seems to be especially promising.

Finally, the flexibility and user friendliness of the system have been greatly improved by the implemented manual settings of the WBAN system. The user can give commands to the PS in two ways. Firstly, by changing setting parameters in a setting file at the RS which the PS periodically downloads. Or secondly by sending an SMS with settings content to the PS.

Bibliography

[1] FOI, http://www.foi.se

[2] Britta Levin, Dennis Andersson, Fredrik Lantz, Systemkoncept för Övervakning av Individstatus, R–2648–SE, FOI, Linköping, Sweden, 2008.

[3] Britta Levin, Dennis Andersson, Fredrik Lantz, Individual Status Assessment Using Wireless Body Area Networks, R–2878–SE, FOI, Linköping, Sweden, 2009.

[4] IEEE 802.15 WPAN Task Group 4, http://www.ieee802.org/15/pub/TG4.html

[5] ZigBee Alliance, http://www.zigbee.org

[6] Haroon Muhammad, Imad Rizk, Jim Ryberg, Sohaib Maalik, ZigBee Wireless Sensor Network, Chalmers University of Technology, Sweden, 2009.

[7] About General Packet Radio Service, http://about.qkport.com/g/general_packet_radio_service.

[8] Christian Bettstetter, Hans-Jörg Vögel and Jörg Eberspächer, GSM Phase 2+ General Packet Radio Service GPRS: Architechture, Protocols and Air Interface, Technische Universität, Munchen, 1999.

[9] Chris Risoz, Trends in Geopositioning for LBS, Navigation and Mapping, School of Surveying and Spatial Information Systems, The University of New South Wales, Sydney, Australia, 2005.

[10] Richard Fuller, Donghai Dai, Todd Walter, Christopher Comp, Per Enge, J. David Powell, Interoperation and Integration of Satellite Based Augmentation Systems, Department of Aeronautics and Astronautics, Stanford University, USA, 1998.

[11] Elliott D. Kaplan and Christopher J. Hegarty, Understanding GPS Principles and Applications, Artech House, 2006

[12] Johan Nilsson, GPS-Data Processing for Continental Drift and Trophospheric Measurements for North and South America Using GIPSY-OASIS Software, Department of Radio and Space Science, Chalmers University of Technology, December, 2009.

[13] National Marine Electronics Association, http://www.nmea.org

[14] Otto, C. and Milenkovic, A. and Sanders, C. and Jovanov, E., System architecture of a wireless body area sensor network for ubiquitous health monitoring, Journal of Mobile Multimedia, vol 1, no 4, pp 303-307, 2006.

[15] Winston H. Wu, Maxim A. Batalin, Lawrence K. Au, Alex A. T. Bui, and William J. Kaiser, Context-aware Sensing of Physiological Signals, Engineering in Medicine and Biology Society, 2007.

[16] Zappi, P. and Lombriser, C. and Stiefmeier, T. and Farella, E. and Roggen, D. and Benini, L. and Tröster, G., Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection, Wireless Sensor Networks, 17–33, Springer, 2008.

[17] Magnus Jobs, Wireless Body Area Network - Monitoring Application System: Nodes & Data Processing, Master Thesis, University of Uppsala, Sweden, 2009.

[18] Bestoon T. Hussain Jaff, A Wireless Body Area Network System for Monitoring Physical Activities and Health-Status via the Internet, Master Thesis, University of Uppsala, Sweden, 2009.

[19] Ravi, N. and Dandekar, N. and Mysore, P. and Littman, M.L., Activity recognition from accelerometer data, Proceedings of the National Conference on Artificial Intelligence, vol 20, issue 3, pp 1541, 2005.

[20] WISENET, http://www.wisenet.uu.se

[21] Hectronic AB, http://www.hectronic.se

Appendix A

Configuration Setup of WBAN Prototype

This is a crib sheet that is intended to describe a step by step procedure of how to connect the PS to a Linux laptop and configure the WBAN system. This may be needed when the software or other features are about to be updated on the PS. It is also described how to start-up the system prior tests and measurments.

1. Serial Port Setup

1. Connect a voltage source (5V) to the battery package in the PS. (To run with batteries is also OK)
2. Connect the serial COM socket to the computer serial COM port
3. Run the communication program minicom in Linux with the command: "minicom -s" to enter the setup menu. If you are not an authorized user of the program you may need to start the program as a super user. Minicom is a communication program for serial ports.
4. Configure the settings in the setup menu in minicom as follows:

   (a) Filenames and Paths - Download directory : /tmp/minicom/download

   (b) Filenames and Paths - Upload directory : /tmp/minicom/upload

   (c) Filenames and Paths - Script Program : runscript

   (d) Serial Port Setup - Serial Device : /dev/ttyS0

   (e) Serial Port Setup - Lockfile Location : /var/lock

   (f) Serial Port Setup - Bps/Par/Bits : 115200 8N1

   (g) Serial Port Setup - Hardware Flow Control : No

   (h) Serial Port Setup - Software Flow Control : No

   (i) Save setup as dfl (default)

   (j) Exit from Minicom

5. Now run minicom with the command “minicom”

6. You should now see a terminal connected to the PS but you can do nothing as long as the PS not is switched on.

7. Switch on the PS. Then you should see a counter counting down from 20 seconds in the terminal. After the 20 seconds, Linux will auto-boot, which will take about 10 seconds.

8. Login on the PS board; Login: root, Password: root

9. 30 seconds after Linux has booted, the WBAN software will auto-start, which implies that the system will create a GPRS connection to the RS, open serial ports to the GPS and ZigBee devices and start to transmit GPS positions and ZigBee data if there are any connected ZigBee sensor nodes. The RS is at the moment this is written (foi.tris.se).

10. To prevent autostart of the WBAN software kill the command “/bin/sh /etc/rc.d/rc.start/rc.6.wban.sh” which will be found in the process list in Linux by the command “ps”. Kill the belonging process ID (PID) with the command “kill -9 PID”, where PID is the process ID number. Now you are free to move around in the Linux system without being interupted by the WBAN software.

11. The /mnt directory of the Linux system is editable, and that is where the configuration files and the WBAN software can be found.

12. /mnt is flash-based which imply that the content doesn’t change at a ree-boot of the PS.

13. All other changes will not be saved at a ree-boot of the PS.

14. /mnt/telenor and /mnt/telenor-connect-chat are configured to be compatible with the present telenor subscription. /mnt/telenor contain settings used by the pppd software in Linux, which in turn is used by rsync when the WBAN software synchronizes files over the GPRS connection. /mnt/telenor-connect-chat contain AT-commands which are used in the communication with the GSM/GPRS modem. The content of the telenor-connect-chat file can be found in appendix B.

15. If the operator subscription is changed, some things may need to be edited in the telenor and telenor-connect-chat files as well. Most probably the APN address need to be changed in the telenor-connect-chat file and maybe a PIN code need to be entered by putting the AT-command “AT+CPIN=<PIN>” in the telenor-connect-chat file.

16. (The PIN code of the current SIM card has previously been inactivated (removed) with the AT-command "AT+CLCK="SC",0,"<PIN>"")

17. If the automatic start of the WBAN software not is killed at startup of the PS, the script wban.sh (under /mnt/) will run after 30 seconds. Wban.sh copies necessary files from /mnt/ into /etc/ and then runs the WBAN software which has the filename wban_<date>. Where <date> is the date when the sofware was created (or latest updated).

18. The files that are copied automatically by wban.sh are; telenor, telenor-connect-chat and wban.conf. Where wban.conf is the configuration file containing all the settings which the WBAN software read at startup. It is the same editable file contained also at the RS which the program is able to periodically download at runtime. All these files can be edited at wish.

19. Data that has been transmitted from the PS to the RS can at the moment this is written be reached by typing the command "ssh tris@tris" from a terminal (xterm) on your computer. The password to log into the tris-server; password: tris.

20. At the tris server the transmitted data can be found under: /opt/local/home/tris/psNewData/

21. The data files are contained in compressed tar libraries and can be extracted by running the script "./extractTarFiles"

2. USB Setup

1. Connect a voltage source (5V) to the battery package in the PS. (To run with batteries is also OK)

2. Connect the USB to the computer

3. Switch on the power button on the PS and wait 30 seconds during boot

4. Configure the USB interface in Linux on your computer according the following procedure:

   (a) Set an IP address and netmask for the USB0 port with the Linux command:
   "sudo /sbin/ifconfig usb0 192.168.90.5 netmask 255.255.255.0"

   (b) Add the PS as a known host with the Linux command:
   "route add -host 192.168.90.2 usb0"

   (c) Check if network is reachable with the Linux command:
   "ping -c 2 192.168.90.2"
   If the connection with the PS is up, the response to the ping command should show a 0% packet loss

5. Login on the PS with the following command:
"ssh root@192.168.90.2"
Password: root

6. If step 4 - 5 is performed quicker than 30 seconds, it is time left to kill the automatic startup of the WBAN software according to step 10 under the Serial Port Setup. For mor information of the PS content please read step 11 - 21 under Serial Port Setup.

3. Start-Up of WBAN System

1. Make sure there are AA batteries in the PS and AAA batteries in the three sensors

2. Press the power button on the PS, then wait 30 seconds for the system to initialize

3. Activate the ZNC by holding the magnet with the opposite side of the visible screw against the LED at the right hand side of the power button

4. The LED will turn yellow when activated

5. The ZNC will be open for sensor connections in 30 seconds only!

6. Activate a sensor by holding the magnet with the side with the visible screw against the opposite side of the lid on the sensor. The LED at the PS will blink red once when a sensor is connected.

7. Repeat step 6 for all three sensors within 30 seconds after the ZNC has been activated

8. After about one minute the WBAN system should have started to transmit data-files to the RS, where the RS destination is specified in the configuration file (wban.conf) at the PS

Appendix B

Modem Connection Chat Content

```
##############################################
############ — A Modem Communication Program — ############
##############################################

### Define rules ###
ECHO ON
ABORT '\nBUSY\r'
ABORT '\nERROR\r'
ABORT '\nNO ANSWER\r'
ABORT '\nNO CARRIER\r'
ABORT '\nNO DIALTONE\r'
ABORT '\nRINGING\r\n\r\nRINGING\r'

### Check if the modem answers, give it 40s respond time ###
'AT'
TIMEOUT 40

### Echo commands to the computer ###
OK ATE1
TIMEOUT 40

### Read all SMS and then delete them ###
#1# Put the modem in textmode
OK AT+CMGF=1
#2# List all SMS in memory
OK AT+CMGL="ALL"
#3# Delete all SMS in memory
OK AT+CMGD=1,4
```

```
### Establish a GPRS connection with the Telenor gateway ###
#1# The APN server
OK 'AT+CGDCONT=1,"IP","internet.telenor.se","0.0.0.0",0,0'
TIMEOUT 40
#2# QoS required
OK AT+CGQREQ=1,0,0,0,0,0
TIMEOUT 40
#3# QoS minimum
OK AT+CGQMIN=1,0,0,0,0,0
TIMEOUT 40
#4# No authentication
OK AT#GAUTH=0
TIMEOUT 40
#5# Dial up
OK ATD*99#
TIMEOUT 40
#6# Wait 40s for a "CONNECT" answer
CONNECT ""
SAY "\nConnected.\r"

### Info ###
# The given commands with answers from the modem can be found in:
# /etc/ppp/connect-errors
# This file is read when the program look for new settings received by SMS
##########

##############################################
########## — End of the modem communication program — ##########
##############################################
```