FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Firefighter of the Future

João Manuel Oliveira Rodrigues

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Supervisor: Gil Manuel Magalhães de Andrade Gonçalves (Professor)

Co-supervisor: João Correia (Engineer)

23 July 2012

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES **2011/2012**

A Dissertação intitulada

"Firefighter of the Future"

foi aprovada em provas realizadas em 17-07-2012

o júri

Presidente Professora Doutora Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Maria Teresa

Professor Doutor José Manuel Tavares Vieira Cabral
Professor Auxiliar Departamento de Eletrónica Industrial da Universidade do Minho

Mestre Gil Andrade Gonçalves
Assistente Convidado do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

João Manuel Oliveira Rodrigues

Autor - João Manuel Oliveira Rodrigues

Faculdade de Engenharia da Universidade do Porto

Resumo

As operações de busca e salvamento efetuadas pelos bombeiros decorrem num ambiente caótico onde a tomada de decisões é um processo crítico que envolve vidas humanas. De modo a facilitar esse processo por parte dos comandantes de cada equipa, surge a necessidade de criar um sistema capaz de monitorizar o estado de cada bombeiro interveniente na operação.

O principal objetivo deste projeto passa por estudar uma alternativa que possa ser usada para apoiar a tomada de decisões dos comandantes. Para tal definiram-se três objetivos: desenvolver um sistema capaz de adquirir os dados vitais e a posição dos bombeiros, integrar esse sistema com uma plataforma web, analisar os resultados e tirar conclusões quanto à sua adequação.

O sistema referido divide-se então em dois subsistemas. O primeiro subsistema é denominado sistema de aquisição e é implementado em cada bombeiro sendo responsável por fazer a aquisição dos dados vitais e da posição de cada um deles. O segundo subsistema, chamado sistema de visualização, tem como principal função reunir os dados obtidos e apresentá-los de forma a que possam ser facilmente interpretados por parte do comandante.

O sistema permite adquirir o ECG de cada bombeiro através de uma placa de aquisição de sinais que comunica através de Bluetooth. A posição de cada bombeiro é adquirida utilizando um receptor GPS.

Os dados adquiridos são enviados para um servidor usando um dos dois métodos disponíveis: através da Internet usando WiFi ou através de uma rede local usando Zigbee. Quando é utilizado o primeiro método, os dados são colocados diretamente na base de dados através do serviço web da PlugSense ficando estes imediatamente disponíveis. Para possibilitar a utilização do segundo método é criada uma rede local dinâmica que permite o reencaminhamento dos dados para o camião. Utiliza-se o *Collection Tree Protocol* para criar uma árvore dinâmica que permite o envio de dados para a raiz, neste caso o camião.

No camião é possível visualizar os dados através da aplicação web da PlugSense. Os dados são atualizados automaticamente aquando a receção de novos valores. É assim possível monitorizar em tempo real a condição física e a posição de toda a equipa.

Após alguns testes, os resultados obtidos mostraram que este sistema pode ser implementado e funciona como esperado. Os dados foram adquiridos e enviados com sucesso para a base de dados, utilizando os métodos descritos anteriormente, e podem ser imediatamente observados com a PlugSense. No entanto, é necessário efetuar testes adicionais num cenário dinâmico para tirar conclusões relativamente ao desempenho da rede local criada com o *Collection Tree Protocol*.

Abstract

The search and rescue operations performed by firefighters occur in a chaotic environment where decision making is a critical process that involves human lives. In order to facilitate this process by the commanders of each team, there is the need to create a system to monitor the status of each firefighter intervening in the operation.

The main objective of this project is to study an alternative that can be used to support decision-making of commanders. For this purpose three objectives were defined: to develop a system capable of measure firefighters vital signals and position, to integrate this system with a web platform, analyze results and conclude on the system adequacy.

This system is divided into two subsystems. The first subsystem is called acquisition system and it is implemented in each firefighter and it is responsible for making the acquisition of vital signals and the position of each firefighter. The second subsystem, called visualization system, has as main function to gather all data and present it so that it can be easily interpreted by the commander.

The system allows to acquire ECG data of each firefighter using a biosignals acquisition board that communicates via Bluetooth. The position of each fire is acquired using a GPS receiver.

The acquired data is sent to a server using one of two methods: via Internet, using WiFi or through a local network, using Zigbee. When the first method is used, data is placed directly in the database through the web service of PlugSense becoming immediately available. To enable the use of the second method it is needed to create a dynamic local area network to enable data to be forward to the firetruck. The Collection Tree Protocol is used to create a dynamic tree that allows sending data to the root, in this case the firetruck.

On the firetruck data can be visualized using the web application of PlugSense. The data is updated automatically upon receipt of new values. It is then possible to monitor in real time the physical condition and position of the entire team.

After some tests, the achieved results proved that this system can be implemented and works as expected. Data was acquired and successfully sent to the database, using the methods described earlier, and it can be immediately observed with the PlugSense. However, it is needed to make additional tests in a dynamic scenario to conclude on the performance of the local network created using the Collection Tree Protocol.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ADC	Analog to Digital Converter
BAB	Biosignals Acquisition Board
CR	Carriage Return
CRC	Cyclic Redundancy Check
CTP	Collection Tree Protocol
DHCP	Dynamic Host Configuration Protocol
ECG	Electrocardiogram
ETX	Expected Transmissions
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Global System for Mobile Communications
I2C	Inter-Integrated Circuit
IP	Internet Protocol
JNI	Java Native Interface
LF	Line Feed
NMEA	National Marine Electronics Association
OS	Operative System
PDA	Personal Digital Assistant
PMB	Personal Monitoring Board
RFCOMM	Radio Frequency Communication
SPI	Serial Peripheral Interface
SSH	Secure Shell
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WMN	Wireless Mesh Network
WSN	Wireless Sensor Networks

Chapter 1

Introduction

1.1 Motivation

Fire rescue is one of the most important public safety activities. After a fire department gets an alarm call, a fire rescue team will be sent to the field. Normally, the team consists of a small number of vehicles and, the most important role, a set of firefighters grouped as squads associated with one of the vehicles. During the process of fire rescue, the commander is in charge of the whole operation, including monitoring the fire field and making real-time decisions on firefighter assignment.

This mode of operation has some gaps. First, the commander cannot have a clear view neither of the status of firefighters nor of the accurate situation on the field which makes the decision-taking difficult for him. Second, a firefighter does not know the danger of the situation around him which places him in danger too. Finally, the fire department located far away cannot get real time information about the operation, which is important in big cities having multiple fires at the same time[16].

To overcome the previous gaps, arises the necessity to develop a system capable of acquiring vital signals and position of each firefighter. This information should be obtained in real time and be available for the commander to help him taking decisions.

This project was developed at Inovamais facilities, in a partnership with Faculty of Engineering of University of Porto. Inovamais is a consultant company that develops innovative projects, promoting the practice of innovation to their customers [17]. The project is presented as a dissertation of Master in Electrical and Computers Engineering. This document presents all stages of the project including a previous study, the development, results and conclusions.

1.2 Description

In this project it is intended to develop a system based on wireless sensor networks to support commanders on their decisions. After taking a first look to the problem, it is possible to divide the system into two subsystems:

- The first subsystem will be on each firefighter. This subsystem will be responsible for obtaining data relative to a firefighter.
- The second subsystem will be located in a firetruck. This subsystem will be responsible for receiving and displaying data to the commander.

The subsystem on each firefighter will have several sensors to obtain vital data and position. These sensors will be connected to a personal monitoring board (PMB) so data can be processed and sent to a base station located in the other subsystem (in the firetruck). The PMB will have the capability to use several wireless communication technologies such as Bluetooth, WiFi, Zigbee and GSM.

In order to transfer data from one PMB to the subsystem located in the firetruck, a valid wireless communication must be set. Since the firefighters are in constant movement, a dynamic network must be created to prevent data loss. Data will be transmitted to the base station using PMBs to forward messages.

If a PMB can be connected to the Internet using a WiFi network, data will be sent directly to the base station instead of using the method described in the previous paragraph.

1.3 Goals

The main goal of this dissertation is to study an alternative that can be used to support the decision making process by the commanders of an operation. To accomplished this goal, the following objectives were defined:

- Develop a system capable of measure firefighters vital signals and position;
- Integrate the system with a web platform that can make the information available to commanders;
- Analyze the results and conclude on the adequacy of such a system.

1.4 Methodology

The methodology used during was organized according with the methods of Systems Engineering defined in: IEEE Std 1220-2005 IEEE Standard for Application and Management of the Systems Engineering Process. So, this work was developed using the following orientation:

1. Requirements analyses;
2. Functional analyses;
3. Synthesis;
4. Verification.

1.5 Organization of the document

This document is divided into six chapters:

1. Introduction - The present chapter briefly presents the project. It contains the motivation, a short description and the objectives.
2. State of Art - In this chapter a study on the available technologies and related projects is presented.
3. Problem Definition - In this chapter is made an overview of the project. It is described a possible scenario to demonstrate how this system can be useful. The requirements of the project as well as the functional analyses will be described.
4. Conception and Implementation - In this chapter the development of the whole system will be explained. First, the implementation of all functionalities on the PMP will be described. Then, will be described how data is acquired and transmitted. Finally, will be presented how data can be visualized. The resume of all process will be detailed in a diagram at the end of this chapter.
5. Integration and Tests - In this chapter, the integration of all the system will be described. Some tests will be presented to demonstrate that all implemented functionalities work correctly.
6. Conclusions - In this final chapter will be presented the conclusions on the project as well as future work that can be developed.

Beyond these paragraphs, this document has also two appendixes:

A. PlugSense - This appendix describes the architecture and functionalities of PlugSense.

B. Biosignals Aquisition Board - This appendix describes the architecture of the BAB used in this project.

Chapter 2

State of art

Before starting to think about any kind of solutions or implementations it is first needed to study the available technologies that can be used in this project. In this chapter, a study is performed in order to get some knowledge about the theme and available solutions.

First it is made a study of the current techniques used by firefighters to overcome their obstacles. Then a study of wireless sensor networks is performed, as well as possible localization techniques and communication technologies. Finally, two related project will be described.

2.1 Firefighters Current techniques

Firefighters had developed several orientation techniques in low visibility environments. What is pretended here, is to describe some of those techniques [3]. They tend to be simple and effective, and the used equipment is low technology but very reliable.

Following a hose is a simple but effective method to find the exit of a dark place or with a lot of smoke. When there are not any hoses, ropes are used to connect a firefighter to a defined point outside the danger zone. This ropes are called lifelines. The other side of the rope may be attached to one element of another team and, if a new team arrives, this line can be used to create new branches in other directions. This lifelines can also contain knots to give firefighters information relative to their location and distance to the exit.

Another used technique is the placement of a lantern at an entry of each division to locate the exit and to inform others that the current place is already being observed. Teams returning from a searching mission can help other by drawing an outline of that place.

Using sounds to indicate the position, thermal cameras to detect victims and radio communications are other techniques that can be used by firefighters.

Although these techniques are simple and can become more effective with practice, they can sometimes fail. A rope can be cut or burned, lanterns can be buried and thermal cameras cannot be used with high temperatures.

2.2 Wireless Sensor Networks

2.2.1 Overview

Wireless sensor networks (WSN) are acquiring a very important role due to technology evolution of micro-electro-mechanical systems which turned the development of intelligent sensors easier. These sensors are small, with limited processing capability and computing resources, but are cheap when compared with traditional sensors. They are capable of measure and obtain information of a certain environment and, based on a local decision process, can transmit data to the user [18].

Intelligent sensors are low power devices equipped with one or more sensors, a processor, memory, a power source, a radio and an actuator. WSN can contain several kinds of sensors and monitor data like temperature, humidity, movement, luminosity, pressure, among others. They can be used in military, environmental, commercial applications and others. An example of this sensor nodes, also known as motes, can be observed in figure 2.1.

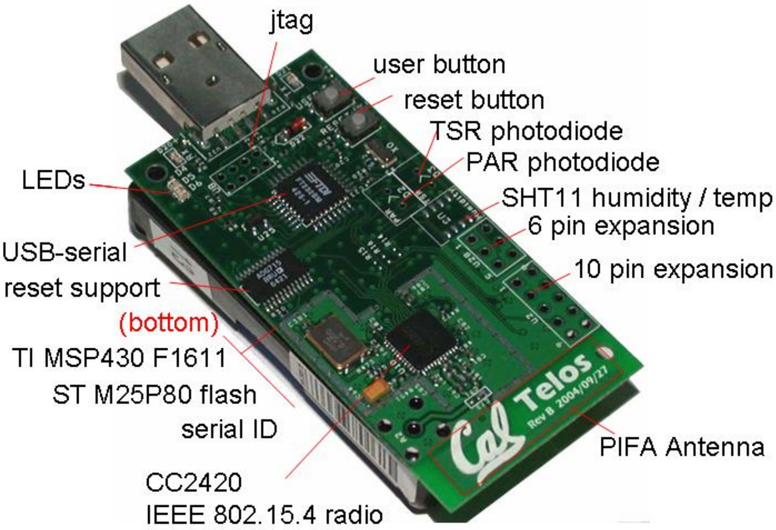


Figure 2.1: An example of a mote (sensor node) [8]

There are two types of WSN: structured and unstructured. In the first ones, some nodes of sensors are deployed on fixed and previously determined locations. This type of WSN uses less sensor nodes and the maintenance costs are lower. In the second type of WSN, sensor nodes are randomly deployed in a field. Using this topology is harder to detect failures and manage connectivity because there is a greater number of nodes. Structured networks have advantages when compared with the other ones- Due to the strategical positioning of nodes, the network can cover all desired area while with random deployment some areas can be missed. [1].

2.2.2 Key issues

To allow applications with wireless sensors, using sensor technology, there is a set of tasks that can be classified in three groups, as shown in figure 2.2.

The first group is the system. Each node is an individual system. In order to support different software applications in a sensor system, it is necessary to develop new platforms, operative systems and storage.

The second group is the protocol communication that allows communications between applications and sensors. It also establishes the communication between other nodes.

The last group contains the services developed to improve applications, network efficiency and system performance.

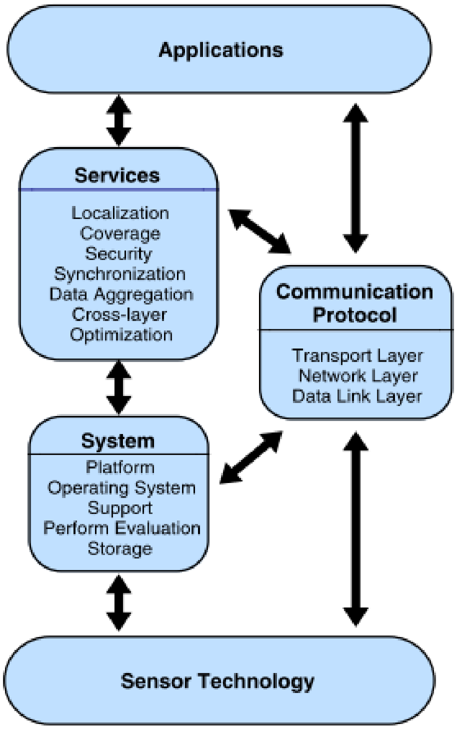


Figure 2.2: Issues in a WSN [1]

From network management point of view it is important that sensor nodes have the capability to organize themselves and to maintain the network efficiently. As sensors are limited in terms of power supply and storage, new communication protocols need to be developed to meet this requirements [1].

2.2.3 Mobile Networks [1]

There are several types of sensor networks: terrestrial, underground, underwater, multimedia and mobile. For this particular case, only mobile networks will be detailed.

Mobile WSN consists in a group of sensor nodes that can move and interact with the physical environment. The great difference is they can reposition themselves and reorganize the network. Nodes can be changed from their original position to obtain more information and transmit it from note to node. Another important issue is data transmission. While in fixed networks the transmission path is always the same, in mobile ones that path is dynamic. Challenges of this network type include deployment, localization, self-organization, navigation and control, coverage, energy, maintenance, and data process.

2.3 Location Principles

There are several localization principles that can be used. Some of those principles are presented bellow.

2.3.1 Global Positioning System [2]

The Global Positioning System (GPS) is a great technological success story. It was developed by the Department of Defense to provide precise estimates of position, velocity, and time.

GPS is a constellation of satellites transmitting radio signals to users. There are about 24 functional satellites. GPS receivers can receive these signals and, using at least three satellites, compute them to estimate the position. GPS signal in space will provide a "worst case" pseudorange accuracy of 7.8 meters at a 95% confidence level.

2.3.2 Infrastructure based [3]

Many localization systems, including GPS, use measures of distance and angle to determine the coordinates of a target. Typically, the target emits signals (infrared, ultrasounds, among others) which serve as identification for a system installed on a building. The sensor measures signal distance and direction and, using a triangulation, can estimate the position of a target.

A different method called fingerprinting uses existing WiFi spots to locate a device. The visible access points and respective signal strength are different along the building, so a client can use it to estimate a location.

These methods have a good level of precision but they can fail with the change of environmental conditions.

2.3.3 Using WSN [3]

There are already developed algorithms that allows the calculation of relative positioning of sensors. This way, each sensor contributes to the calculation of its location instead of using a central device. If some sensors are used as anchors with a known position, it is possible to obtained the absolute position of the remaining nodes.

This method does not need any kind of calibration, exception made to the anchor nodes. Sensors are placed and, due to wireless communication, there is no need of any infrastructure. The failure of a sensor does not provoke the network to fail, but moving an anchor to a different position can cause wrong position measures. However, this would not be a realistic solution for search and rescue missions in houses or buildings but in airports or subterranean parks.

2.3.4 Ad hoc Relative Positioning [3]

To determine the direction of a target is crucial to many techniques of basic navigation, including following a sound or a light. Using physical rotation, a transmitter or receiver, or using several transmitters and receivers, it is possible to obtain angular measures between devices.

This method has the advantage of being simple and does not need any complex processing.

2.4 Wireless Technologies

Wireless communication is the transfer of data between devices without using a physical connection. This communications can be performed using radio frequency signals, microwaves or infrared. There are several technologies used to transfer data and the most relevant for this project are the following:

- Bluetooth - The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth Specification defines a uniform structure for a wide range of devices to connect and communicate with each other [19].
- WiFi - Is the popular name given to the set of IEEE 802.11 standards. This standard is responsible for the implementation of a wireless local area network.
- GSM - It is a communication used by mobile phones all over the world. This technology supports short text messages, data networking and voice communication. The evolution of this technology allowed the introduction of others like GPRS (General Packet Radio Service) wich allows the transfer of data packets.
- Zigbee - It builds upon the IEEE 802.15.4 standard which defines the physical and MAC layers for low cost, low rate personal area networks [20]. This technology is the most used on WSN.

The table 2.3 shows the comparison between some wireless technologies.

2.5 Related Work

2.5.1 The E-SPONDER Vision [4]

The E-SPONDER project aims to the development of an infrastructure that provides communication and decision support to the rescue units involved using the most recent technologies. The system architecture can be observed in figure 2.4 where the key elements of this system are presented. The project of first responder units has the objective to select the best mobile hardware system to respond to user needs. Their system should have an interface with several modules (3D visualization and communication with the mobile center). This application should consider important issues like sensors, communications, interior and exterior navigation, among others.

	ZigBee	**802.11 (Wi-Fi)**	**Bluetooth**	**UWB (Ultra Wide Band)**	**Wireless USB**	**IR Wireless**
Data Rate	*20, 40, and 250 Kbits/s*	11 & 54 Mbits/sec	1 Mbits/s	100-500 Mbits/s	62.5 Kbits/s	20-40 Kbits/s 115 Kbits/s 4 & 16 Mbits/s
Range	*10-100 meters*	50-100 meters	10 meters	<10 meters	10 meters	<10 meters (line of sight)
Networking Topology	*Ad-hoc, peer to peer, star, or mesh*	Point to hub	Ad-hoc, very small networks	Point to point	Point to point	Point to point
Operating Frequency	*868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)*	2.4 and 5 GHz	2.4 GHz	3.1-10.6 GHz	2.4 GHz	800-900 nm
Complexity (Device and application impact)	*Low*	High	High	Medium	Low	Low
Power Consumption (Battery option and life)	*Very low (low power is a design goal)*	High	Medium	Low	Low	Low
Security	*128 AES plus application layer security*		64 and 128 bit encyption			
Other Information	*Devices can join an existing network in under 30ms*	Device connection requires 3-5 seconds	Device connection requires up to 10 seconds			
Typical Applications	*Industrial control and monitoring, sensor networks, building automation, home control and automation,*	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets	Streaming video, home entertainment applications	PC peripheral connections	Remote controls, PC, PDA, phone, laptop links

Figure 2.3: Comparison between some wireless technologies [9]

The mobile emergency center should work as a bridge between the operation center and the first responder units. It will have to deal with issues related with communications and data exchange.

Finally, the operation center, located at the headquarters, will be responsible for gather all data from the field transmitted by the mobile emergency unit. The issues related to the operation center are similar to the mobile unit with addition of a geographical information system and the E-SPONDER portal, as well as the system interfaces.

It is important to realize that this system is equivalent to a Wireless Mesh Network(WMN) [21] since that all nodes are characterized by their mobility and, directly or indirectly, communicate between them.

Other important characteristic of this system is the autonomous configuration of the system used by first responder units. It is extremely important that a network is created without human intervention. There is the need to use network protocols capable of fulfill this requirements.

2.5.2 MobileMap [5]

MobileMap is an application designed to execute in mobile devices, such as cellphones and PDAs, as a complement to the already existing radio communications. This application uses maps previously created and loaded to the mobile device containing geographical information of the field

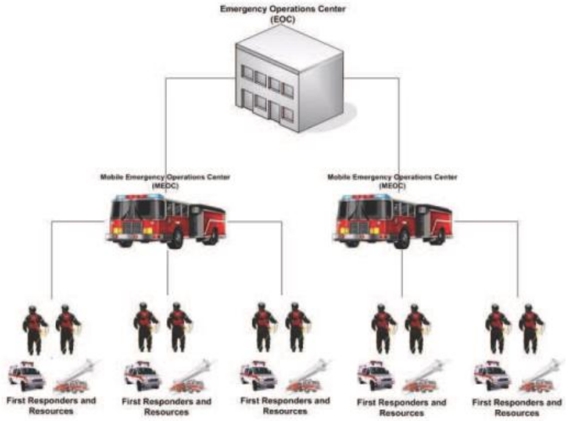


Figure 2.4: E-SPONDER Architecture[4]

and interacts with the command center through messages sent via WiFi or GSM.

This system implements the following functionalities:

- Navigation service - allows a firefighter to navigate through the map manually or automatically(if the device has the capability to use a GPS receiver);
- Destination service - allows user to introduce destination (using latitude and longitude) or the desired address. After defining the location, the application shows the direction of the destination and the path to follow to arrive there;
- Distance calculation - Calculates the distance between points defined by user.
- Information management - Allows the user to visualize information relative to emergency situations, useful spots like police stations and hospital, and information exchange with other firefighters or the command center;
- Localization - Shows the location of users and near firetrucks.

This application uses a service oriented architecture where the command center and each mobile device is a potential producer/consumer service. The information is shared with all devices and each user uses the one that is most useful.

Chapter 3

Problem definition

In this chapter, a possible scenario will be described in order to understand in what kind of situations this system can be useful. Then, will be presented a discussion of requirements and the architecture of the system as well as a functional analysis.

3.1 Scenario description

Next to an apartments building there is a small forest with a great density of trees. Suddenly, due to a cigarette that was thrown from a car passing by, a fire starts. When people living in the building notice the fire, they call the local Fire Station to ask for help because the fire has already spread over the forest putting apartments in danger.

A truck full of firefighters leaves the stations and quickly arrives to the forest. Each element has attached to his outfit two small devices: a BAB (Biosignals Acquisition Board) and a PMB (Personal Monitoring Board). The first one is responsible for measuring vital signals of a firefighter and the second one manages all information related to him. In the truck there is a computer that has Internet access through an Internet dongle.

When the team arrives, the commander orders the firefighters to place in strategical positions to fight the arising fire while he stays in the truck commanding the operation and watching it all through a computer. Albert and John go next to the apartments to prevent that fire reaches the building, while the others go further inside the forest to stop the spread of the fire. Figure 3.1 shows the disposition of the team.

Periodically, the PMB attached to each firefighter sends the information relative to him to the computer located in the truck. The PMBs of Albert and John can find an available network with Internet access provided by one of the apartments in the building, so it is used to directly send information to the computer in the truck.

Unfortunately, the other firefighters do not have Internet access, but with the strategical positioning of Albert and John a wireless network is created and data can be sent indirectly to the truck by retransmitting the information until it arrives to the truck.

Some firefighters are surrounded by trees and cannot be located using GPS, but using the local network it is possible to estimate their relative position. Firefighters having a valid GPS signal are used as references to determine the absolute position of others.

During the whole operation, the commanders stays in the truck watching the development of all the action while it gives orders to all with his radio communicator. With connection to the Internet, he can see where is each firefighter and how are his vital signs.

The wind then gets stronger and the smoke starts to spread between the field leaving the firefighters with low visibility. Jack, due to the place's high temperature, has a drop in blood pressure and faints. Due to the low visibility, none of his teammates notices this occurrence. As his pulse rate drops, an alarm fires on the computer in the truck and the commander immediately notices it, analyses the position of the team on the field and orders Steve, the closest to Jack, to go and rescue him. Steve is in a very critical situation: his visibility is low and he cannot find Jack. The commander guides him through the radio while watching both men's position on the computer screen. Steve successfully arrives to his teammate before the fire do, and brings him to the truck to receive treatment. Jack receives local assistance and he wakes a little dizzy due to smoke inhaled, but still alive. The quick intervention of the team has saved a life.

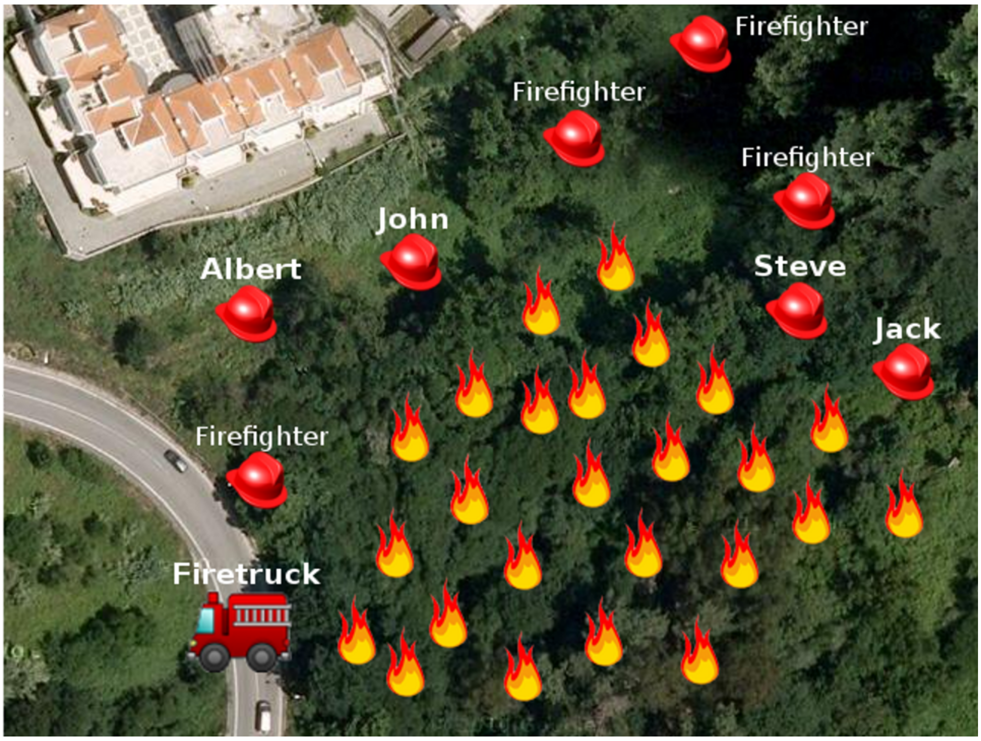


Figure 3.1: Scenario example

As the fire fighting goes on, the water resources are getting low and the tank needs to be refilled. The commander checks the computer for water supplies near him and starts planning the refilling of the truck's tank. Fortunately, a few moments later the fire is extinguished and there is no need to refill the tank. The team then returns to the Fire Station with their duty accomplished.

3.2 Requirements

In order for make the previous scenario possible, the following requirements were specified. The requirements are divided in two groups which are divided in two subgroups:

1. Functional Requirements
    1.1. Acquisition System;
    1.2. Visualization System.
2. Non-functional Requirements
    2.1. Acquisition System;
    2.2. Visualization System.

3.2.1 Functional Requirements

3.2.1.1 Acquisition System

Table 3.1: Acquisition System - Functional Requirements

Code	Description
FR_ACQ_1.	The system should acquire the firefighters' vital signals
FR_ACQ_2.	The system should acquire the firefighters' position
FR_ACQ_3.	The system should transmit data using wireless technologies
FR_ACQ_4.	The system should automatically identify available wireless communication to use
FR_ACQ_5.	The system should recover from a failure

3.2.1.2 Visualization System

Table 3.2: Visualization System - Functional Requirements

Code	Description
FR_Vis_1.	The system should display the location of all firefighters
FR_Vis_2.	The system should display the vital signals of all firefighters
FR_Vis_3.	The system should have Internet connection
FR_Vis_4.	The system should use the same wireless technologies as the acquisition system to receive data
FR_Vis_5.	The system should notify the user of irregular situations
FR_Vis_6.	The system should allow the addition of new firefighters
FR_Vis_7.	The system should allow the configuration of a team
FR_Vis_8.	The system should allow the management of sensors

3.2.2 Non-functional Requirements

3.2.2.1 Acquisition System

Table 3.3: Acquisition System - Non-Functional Requirements

Code	Description
NFR_ACQ_1.	The system should be easy to install in a firefighter
NFR_ACQ_2.	The system should not limit firefighters movements
NFR_ACQ_3.	The system should be resistant to water and shocks
NFR_ACQ_4.	The system should have a long battery life
NFR_ACQ_5.	The system should have low power consumption
NFR_ACQ_6.	The system should not need user intervention once initiated
NFR_ACQ_7.	The system should allow the addition of new sensors

3.2.2.2 Visualization System

Table 3.4: Visualization System - Non-Functional Requirements

Code	Description
NFR_Vis_1.	The system should be intuitive and easy to use
NFR_Vis_2.	The system should have an organized interface to be easy to observe data
NFR_Vis_3.	The system should allow the customization of the user interface
NFR_Vis_4.	The system should allow the visualization of useful information like water supplies, hospitals, fire departments, police stations, among others.

3.3 Constraints

Besides the previous specified requirements, a few constraints were also imposed to the system development:

- The use of an already developed board to obtain the vital signals of a person. This board has a few sensors implemented and communicates over bluetooth. The architecture of this board is described in Appendix B;

- The use of a platform to store and visualize the obtained data called PlugSense. This platform can obtained data from several sensors (over serial communication, bluetooth or using a web service), save the data and display it. This system is described in Appendix A;

- The system's development must be based on an already acquired board - pico-SAM9G45. The functionalities of this board will be described on the next chapter.

3.4 Architecture of the System

With the requirements specified, it is now possible to start thinking about the system's organization.

From the previous scenario description and from the requirements analysis, the system can be divided into three parts according to the functionality: data acquisition, data transmission and data visualization.The first two functions are made by the PMB while the last one will be made by the computer located in the truck.

Focusing in the data acquisition part, the PMB has to have capability for acquiring two types of data: vital signs and positioning.

The vital signals are acquired by the BAB which was already developed in a previous project. This board has a bluetooth interface to send and receive commands so, the PMB can obtain this signals using a bluetooth dongle.

To get the firefighters position, several methods can be used as described in the state of art section. For exterior localization, a GPS receiver proves to be a very accessible and effective solution, but for interior localization that solution is no longer possible. In that case, an alternative solution, based on the wireless network, has to be studied.

The transmission of data, as specified on the requirements, has to be wireless and there has to be more than one way to transfer data. To ensure that data can always be transmitted to a base station, there should be a method capable of working in any place, that method is using a local network created by the firefighters on the field using PMBs as retransmission nodes.

The other alternative is to send data directly to the base station using an available wireless technology like WiFi or GSM.

When data is received on the base station, it is necessary to display all information in a user-friendly interface. This interface should allow a user to look at the screen and quickly understand the status of all firefighters.

3.4.1 Functional Analysis

Figure 3.2 shows an overview of the system's architecture. It is possible to observe that there are three physical components: BAB, PMB and a local computer containing the interface. The BAB is ready to be used and it does not need any kind of additional development. In order to obtain measures taken by it, it is needed to add a bluetooth dongle to the PMB. In the following sections it will be described the functionalities of the system's components except for the BAB which only function was already described.

3.4.1.1 Personal Monitoring Board

The main modules of the PMB can be found in figure 3.3. Each block have one or more associated functionalities. Those functionalities can be found in table 3.5. Each functionality as an associated code, a description and the modules involved.

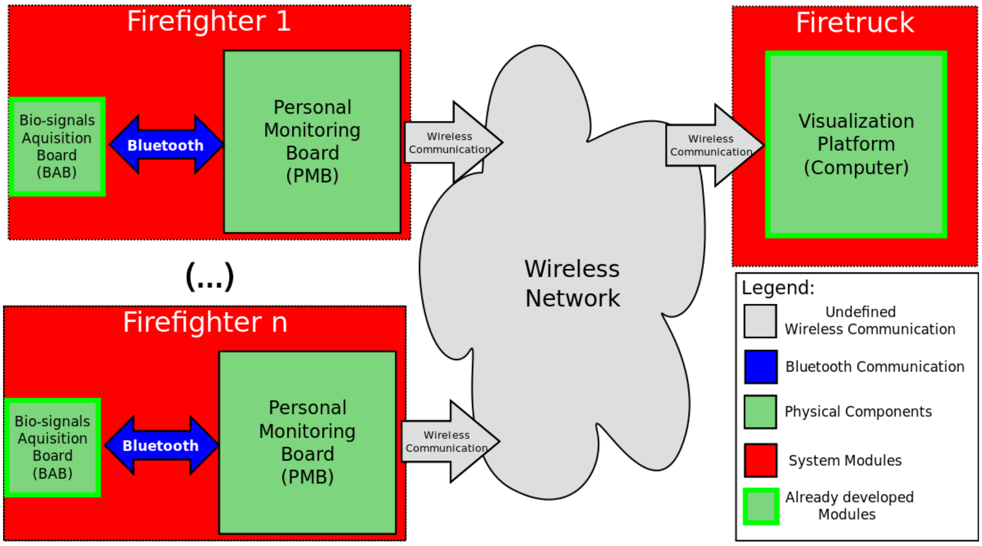


Figure 3.2: System's high level architecture

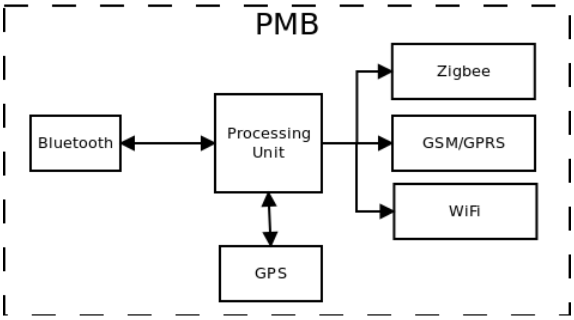


Figure 3.3: Block diagram of a PMB.

Table 3.5: PMB - Functionalities specification

Code	Description	Modules
Func_PMB_1	Get vital signals from the BAB	Bluetooth
Func_PMB_2	Obtain geographical position	GPS
Func_PMB_3	Obtain relative position	Zigbee
Func_PMB_4	Have an unique ID	Processing Unit
Func_PMB_5	Process received data to avoid errors	Processing Unit
Func_PMB_6	Forward messages received over the local network	Zigbee, Processing Unit
Func_PMB_7	Maintain the organization of the network	Zigbee
Func_PMB_8	Send messages directly to the firetruck	GSM/GPRS, WiFi
Func_PMB_9	Identify all available communication alternatives	Processor Unit

3.4.1.2 Visualization Platform (Computer)

The main modules of the visualization platform can be found in figure 3.4. The right side of the image represents a Fire Station just to demonstrate where and how data is stored. Each block have one or more associated functionalities. Those functionalities can be found in table 3.6. Each functionality as an associated code, a description and the modules involved. This modules are only the ones of the visualization platform (Fire Station modules are not included).

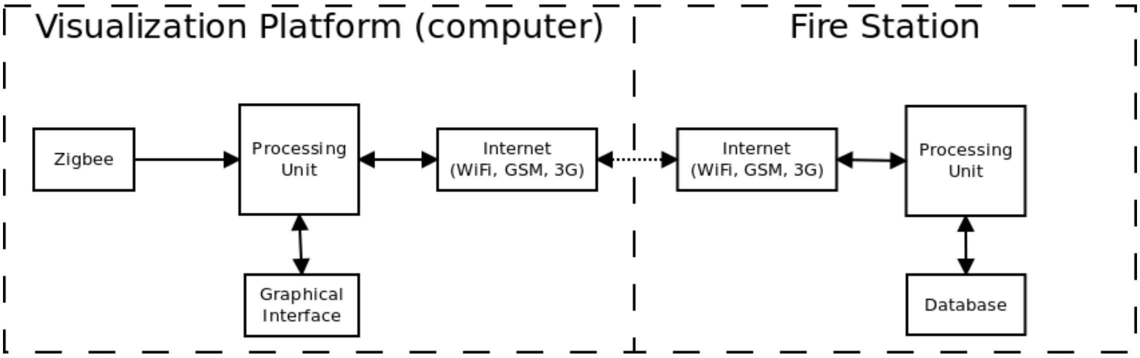


Figure 3.4: Block diagram of the visualization platform (and the Fire Station).

Table 3.6: Visualization Platform (Computer) - Functionalities specification

Code	Description	Modules
Func_Vis_1	Have Internet access (using dongles)	Internet
Func_Vis_2	Show firefighters position	Graphical Interface, Internet
Func_Vis_3	Display alarms	Graphical Interface, Processing Unit
Func_Vis_4	Display firefighters vital signals	Graphical Interface, Internet
Func_Vis_5	Show location of useful information like water supplies, hospitals, fire departments, police stations, among others	Graphical Interface, Processing Unit, Internet

Chapter 4

Conception and Implementation

After defining all functionalities of the system, it is now time to start implementing all systems functionalities. In this chapter, is described how the whole system is implemented. That implementation is divided into four steps: implementation of the required devices on the board, implementation of data acquisition which includes vital signals and position, signal transmission and finally the integration of the visualization platform.

4.1 System Specification

Based on the defined requirements and constraints, an analysis of the required components to use on the system needed to be performed.

It is already known that the board must have a bluetooth connection to communicate with BAB. It is also needed to obtain the position that can be done using a GPS receiver or using the local network to obtain relative position. Even if the second option is used, it is still needed to have a reference point, so a GPS receiver has to be implemented in at least one firefighter or in a fixed anchor to have one reference position.

One method to send data to the base station could be using the PlugSense web service. To use this, a valid Internet connection must be available. The use of a WiFi device proves to be an accessible solution since there are no additional costs if a network is found. To use GSM service, the board needs to be upgraded to obtain the necessary hardware since it only has the SIM slot, so this technology was not implemented.

If an Internet connection is not available, an alternative solution is required. The use of a Zigbee communication is the main solution to create a local network in order to solve this issue. The main solutions to create the network were using a Zigbee module or a mote device. The choice was to use motes because they have a dedicated operative system with many already implemented functionalities and they can contain a large variety of integrated sensors to obtain additional environment data that can be useful to the commander such as temperature, amount of oxygen in the air, among others.

Having studied all possible solutions the architecture can be updated. In figure 4.1 a more specific architecture of the subsystem in the firefighter.

The programming language used was java. This decision was mainly taken due to fact that TinyOS programming has more detailed support for this language. Also, since Android is a popular operative system for mobile devices and its applications are programmed in this language, if later this OS is installed on the PMB to match further requirements, this application can be ported as well.

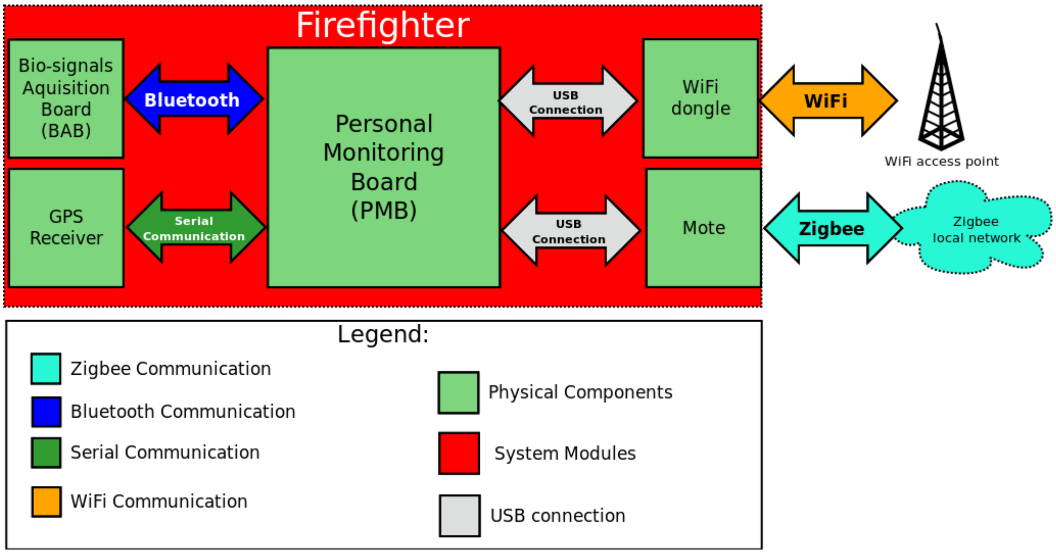


Figure 4.1: Detailed architecture of the subsystem present in each firefighter.

4.2 Personal Monitoring Board Configuration

The board used in this project, has been already chosen previously for another related project and it is a pico-SAM9G45 manufactured by Mini-Box.com.

4.2.1 The board: pico-SAM9G45

Pico-SAM9G45 is a highly extensible low cost, low power, pico-ITX compatible board. Its main characteristics are the following:

- Processor: ARM9, 400Mhz, ARM926EJ-S, 32/32K;
- Memory: 256MB DDR2;
- USB Ports: 4 USB 2.0 ports 480Mbps, shared;
- SD card reader;
- SIM card slot;

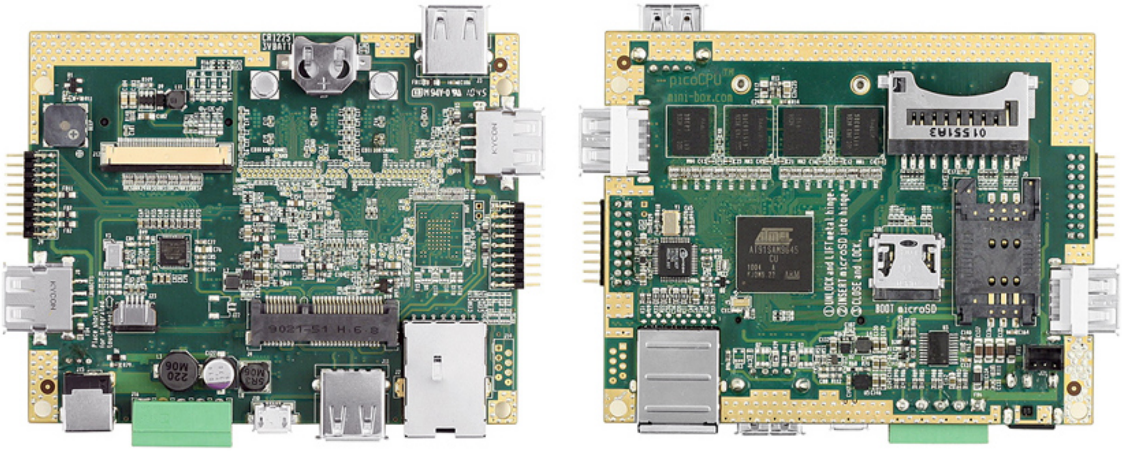


Figure 4.2: pico-SAM9G45 board (front and back view) [10]

- Bootable microSD;
- Expansion header for SPI, I2C, RS232;
- Linux / Android boot options.

With those characteristics, it is possible to verify that the requirements specified can be successfully met: the USB Ports can be used to implement new communication methods using USB dongles[1] like bluetooth or WiFi, the SD card reader allows to store data in an SD for further use, the SIM card slot can be used to send data over GSM and the expansion header gives the possibility to add any kind of device that communicates trough serial port, for example a GPS receiver. As the board runs an OS like Linux or Android, it is possible to include software that was previously developed without having to make significant changes.

Besides that, this boards possess an official Wiki page [10] to help its users to configure it and to get started using its functionalities.

4.2.2 Board configuration

Before starting to develop any kind of program, it was needed to test all board functionalities to be used. According to the Wiki page, to start using the board a microSD card must be present containing the OS files. The choice was to use the Linux distribution called Ångström. The familiarization with Linux was greater when comparing with Android which had a huge importance on this decision. Another important issue to decide to use Linux was the fact that the graphical interface on the board was not going to be used in this project where Android is mostly used.

After copying the OS files to the microSD card and installing it on the board, a connection with it needed be established. This connection was done through an Ethernet cable connecting the

---

[1] A USB Dongle is a device designed to fit in a USB port that adds new functionalities which were not included when the system was manufactured.

board to a computer. In order to get access to the board, a valid IP must be assigned to the board so, a DHCP Server[2] was configured on the computer side.

After the IP has been assigned to the board, it was possible to access it using SSH[3]. It was time to start making tests to the board.

4.2.2.1 Bluetooth

The first test is to make sure the available USB bluetooth dongle worked on the board. The dongle was connected to the board and started the search for near devices and for bluetooth services associated to them. Since the board communicates using RFCOMM[4] the main goal was to connect to a device using this protocol. After defining the PIN the connection was successfully established and it was proved that the bluetooth device worked correctly and could be used to communicate with the BAB.

4.2.2.2 WiFi

The test procedure of the USB WiFi dongle was the same as the one described for the bluetooth dongle. After connecting the dongle, it was still unable to scan for available network. Fortunately, after finding and updating the firmware of this type of dongle, it was possible to find available networks. There was a successful attempt to connect to the only known visible network. After connecting to it, a ping was sent to www.google.com with a successful response indicating that the WiFi dongle worked correctly and could be used to access to Internet.

4.2.2.3 Serial Communication

The test made to serial port was the simplest one, a jumper was used to connect RX0 and TX0 terminals of the board. Random characters were written over serial port *ttyS0* and the same characters were received back. After this procedure the proper functioning of serial port was verified.

4.2.2.4 Mote (Zigbee)

This was the configuration that presented most problems. Initially, the mote was not recognized by the PMB because drivers for the FTDI chip[5] were not present on the installed kernel. This was one of the difficulties encountered during the project.

With the help provided by the pico-SAM9G45 wiki, the kernel was recompiled with the drivers needed. However, the PMB still could not use the motes because the JNI[6] file to interact with the motes could not be found. It was needed to compile the required files for the architecture of the

---

[2] A DHCP server assigns automatically an IP to a client.

[3] SSH is a network protocol that allows a connection with another computer in the network and execute remote commands.

[4] RFCOMM is a transport protocol that simulates a serial port communication over bluetooth.

[5] FTDI chip is a component that allows the use of an USB input/output as a normal serial port

[6] Java Native Interface is a programing framework that enables the integration of java code with code in other languages. This way it is possible to call native libraries and applications.

board. After searching deeply for the native files in the TinyOS repository, they were found and compiled to obtain a new JNI file to interact with the processor integrated on the PMB. It was then finally possible to interact with the mote.

4.3 Data Acquisition

As it was mentioned several times on the previous chapter, the signal acquisition is made by the BAB board and the connection to it is made over bluetooth. The protocol used to communicate with the BAB is explained in the next subsection.

4.3.1 The Protocol

Table 4.1 shows an example of a request message for pulse rate data.

Table 4.1: Protocol used by the BAB - example message

02	47	45	7D	03	7E

The data messages used by this protocol have the following format:

- The first byte is the message start byte and it is always the hexadecimal value 02;
- The second byte is the command. It can be character ’G’, ’S’ or ’E’ (represented by the hexadecimal values 47, 53, 45 respectively) which represent the functions Get, Set and Error respectively;
- The third byte is the type of data to request or the type of data containing in the message. It can be the character ’E’ for ECG data, ’O’ for oximetry, ’B’ for blood pressure, ’P’ for pulse rate, ’H’ for all (includes ECG data of one second) and finally ’A’ to get all data except ECG (the characters representation in hexadecimal values are 45, 4F, 42, 50, 48 and 41 respectively). The last two types cannot be used in the Set function.
- The fourth byte is the amount of data (number of bytes) contained in the message.
- From the fifth byte data values are included.
- The last two bytes are the closing byte 03 followed by the CRC[7] byte.

In this project, the only implemented sensor on the BAB is the ECG sensor. As the pulse rate can be obtained from it, the implementation of the previous protocol on the board only responses correctly to three requests:

- Get ECG - the board responds with 1500 bytes of ECG data;
- Get Pulse rate - the board responds with one byte indicating the pulse rate;
- Get One second ECG - responds with the values of ECG of one second.

---

[7] Cyclic Redundancy Check is a technique to verify errors in transmitted data.

4.3.2 Data Processing

After knowing the protocol used by the BAB, the program to acquire the data was developed. First, a bluetooth connection with the BAB is established and then the process to obtain data begins as specified on figure 4.3.

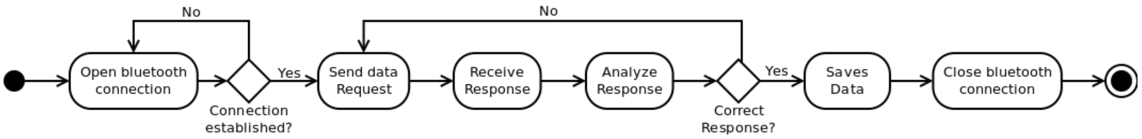


Figure 4.3: Vital signals acquisition process

As it can be seen on the picture, the PMB first attempts to communicate with the BAB. After the connection is established, the PMB sends a request to receive data. This data can be the pulse rate or the ECG. After receiving the response, an analysis is performed to check if the message is according to protocol and verifying that the CRC byte matches the one calculated locally. After this process the communication is closed.

4.4 Positioning

To determine the geographical position of the firefighters a GPS receiver was introduced in the board but, before that a receiver needed to be chosen.

4.4.1 GPS Receiver Selection

After searching for GPS receivers, the following four devices came up as possible solutions:

1. **66 Channel UP-501**

   Channels: 22 Tracking, 66 Acquisition;

   Power: 75mW at 3.0V;

   Protocol: NMEA (GGA, RMC, GSV, GSA);

   Acquisition Time: Unknown (Acquisition sensitivity: -148 dBm);

   Position Accuracy: Unknown

   Built-in battery: Yes;

   Antenna: Internal.

2. **66 Channel LS20031**

   Channels: 22 Tracking, 66 Acquisition;

   Power: 41mA at 3.3V;

Protocol: NMEA (GGA, GLL, GSA, GSV, RMC, VTG);

Acquisition Time: Hot start < 2s, Cold start 35s;

Position Accuracy 3m;

Built-in battery: Yes;

Antenna: Internal.

3. **20 Channel EM-406A**

   Channels: 20 channel all-in-view tracking;

   Power: 70mA at 4.5-6.5V;

   Protocol: NMEA (GGA, GLL, GSA, GSV, RMC, VTG) and SiRF binary protocol;

   Acquisition Time: Hot start 1s, Warm start 38s, Cold start 42 s;

   Position Accuracy: 5m;

   Built-in battery: Yes;

4. **20 Channel EM-408**

   Channels: 20 channel all-in-view tracking;

   Power: 75mA at 3.3V;

   Protocol: NMEA (GGA, GLL, GSA, GSV, RMC, VTG);

   Time to first fix: Hot start < 2s, Cold start 35s;

   Position Accuracy: 5m;

   Built-in battery: Yes;

   Antenna: Internal or external.

All the previous receivers could be implemented on the board and their characteristics are similar, but one had to be chosen. The choice was to use the receiver 4 because it already came with a cable and the integration with the PMB was more accessible. Despite it consumes a little more power than the others, an external antenna could be added to increase the performance of the device. However, due to time constraints, to ensure that the device arrived on time to familiarize with it, an alternative choice was made and instead of acquiring the receiver 4 witch would take more time to arrive, the device 3 was the one obtained. Its characteristics are similar with the other one and it also has a led indicating if the position is fixed or not which is a great help while developing the system.

All of the presented GPS receivers communicate using NMEA messages. The protocol is briefly explained on the next subsection.

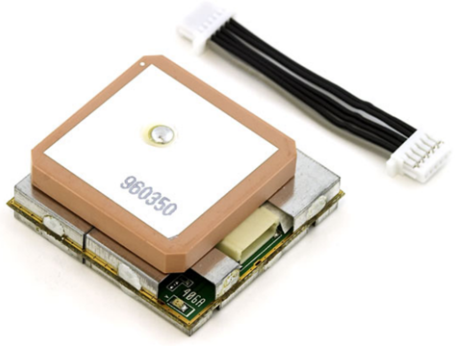


Figure 4.4: EM-406 GPS Receiver [11]

4.4.2 The NMEA 0183 Protocol [6]

4.4.2.1 What is the NMEA 0183 Standard?

The National Marine Electronics Association (NMEA) is a non-profit association of manufacturers,distributors, dealers, educational institutions, and others interested in peripheral marine electronics occupations. The NMEA 0183 standard defines an electrical interface and data protocol for communications between marine instrumentation.

4.4.2.2 Electrical Interface

NMEA 0183 devices are designated as either talkers or listeners (with some devices being both), employing an asynchronous serial interface with the following parameters:

Baud rate:	4800
Number of data bits:	8 (bit 7 is 0)
Stop bits:	1 (or more)
Parity:	none
Handshake:	none

4.4.2.3 General Sentence Format

All data is transmitted as sentences. Only printable ASCII characters are allowed, plus CR (carriage return) and LF (line feed). Each sentence starts with a '$' sign and ends with <CR><LF>.There are three basic kinds of sentences: talker sentences, proprietary sentences and query sentences.

- **Talker sentences** - The general format for a talker sentence is:

  $ttsss,d1,d2,....<CR><LF>

  The first two letters following the '$' are the talker identifier. The next three characters (sss) are the sentence identifier, followed by a number of data fields separated by commas, followed by an optional checksum, and terminated by carriage return/line feed. The data fields are uniquely defined for each sentence type. A sentence may contain up to 80 characters plus ’$’ and CR/LF. If data for a field is not available, the field is omitted, but the delimiting commas are still sent, with no space between them. The checksum field consists of a ’*’ and two hex digits representing the exclusive OR of all characters between, but not including the ’$’ and ’*’.

- **Proprietary Sentences** - The standard allows individual manufacturers to define proprietary sentence formats. These sentences start with "$P", then a 3 letter manufacturer ID, followed by whatever data the manufacturer wishes, following the general format of the standard sentences.

- **Query sentences** - A query sentence is a means for a listener to request a particular sentence from a talker. The general format is:

  $ttllQ,sss,<CR><LF>

  The first two characters of the address field are the talker identifier of the requester and the next two characters are the talker identifier of the device being queried. The fifth character is always a ’Q’ defining the message as a query. The next field (sss) contains the three letter mnemonic of the sentence being requested.

4.4.3 Data Processing

After understanding the NMEA protocol, it was needed to analyze the input and output messages supported by the chosen GPS receiver. The list of messages are specified below. A detailed description can be found on the GPS receiver datasheet[15].

4.4.3.1 Output messages

- GGA-Global Positioning System Fixed Data;
- GLL-Geographic Position-Latitude/Longitude;
- GSA-GNSS DOP and Active Satellites;
- GSV-GNSS Satellites in View;
- RMC-Recommended Minimum Specific GNSS Data;

From the previous messages, the one that has the most important information for this application is the GGA message. Its description is detailed on table 4.2 extracted from the device datasheet. From those values, the relevant fields for our application are:

- Latitude;
- N/S indicator;

- Longitude;
- E/W indicator;
- Position fix indicator - to know when data is valid;
- MSL Altitude[8].

Table 4.2: GGA message description [14]

Example Message:
$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,,,0000*18

Name	**Example**	**Units**	**Description**
MessageID	$GPGGA		GGA protocol header
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		Value Description 0 Fix not available or invalid 1 GPS SPS Mode, fix valid 2 Differential GPS, SPS Mode, fix valid 3 GPS PPS Mode, fix valid
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR><LF>			End of message termination

4.4.3.2 Input messages

Bellow, it is a list of all input messages that can be sent to the GPS receiver. All messages has the format showed in table 4.3. The <ID> field on the table is the message identifier and it should be replaced by the ID of the list below. The message format must be according to the protocol described earlier. This list only presents the input NMEA messages that GPS receiver can receive.

- Set Serial Port - ID:100 - Set PORT A parameters and protocol.

---

[8] Mean Sea Level - altitude compared with the average sea level

Table 4.3: NMEA messages - input format [14]

Start Sequence	Payload	Checksum	End Sequence
$PSRF<ID>	Data	*CKSUM	<CR><LF>

- Navigation Initialization - ID:101 - Parameters required for start.
- Set DGPS Port - ID:102 - Set PORT B parameters for DGPS input.
- Query/Rate Control - ID:103 - Query standard NMEA message and/or set output rate.
- LLA Navigation Initialization - ID:104 - Parameters required to start using Lat/Lon/Alt.
- Development Data On/Off - ID:105 - Switch Development Data Messages On/Off.
- Select Datum[9] - ID:106 - Selection of datum to be used for coordinate Transformations

Such as for the output messages, not all of these messages are used. The serial port parameters were left in their default values as specified in 4.4.2.2. In case of system failure, it is still possible to start using the GPS receiver because default values are used, so there is no need to change them.

As the user cannot introduce his coordinates, it is not possible to start using a known location. It will always start using the location stored on its memory. Datum selection is also the default one (World Geodetic System 1984) and, like serial port parameters, it is not changed.

Basically, the only functions used are the navigation initialization to start using the device and the query/rate control to manage message reception. The initialization message used was the LLA Navigation Initialization because, when making tests to the receiver, it was more accessible to obtain the latitude and longitude using a web application like Google Maps then the X,Y,Z coordinates. However, both messages can be used since the inputed coordinates are not used.

The message format of the LLA Navigation Initialization and the Query/rate control messages are detailed on table 4.4 and 4.5 respectively.

4.4.3.3 Communication Description

Before obtaining the position, it was first needed to configure the GPS receiver. In order to obtain data, the GPS receiver must be initialized so it can start searching for the position.

The first step was to send the initialization command using LLA Navigation Initialization with the following configuration:

---

[9] Datum is a method to define the shape and size of the planet Earth

Table 4.4: LLA message description [15]

$PSRF104,<Lat>,<Lon>,<Alt>,<ClkOffset>,<TimeOfWeek>,<WeekNo>,
<ChannelCount>,<ResetCfg>*CKSUM<CR><LF>

<Lat>	Latitude position, assumed positive north of equator and negative south of equator (float, possibly signed)	
<Lon>	Longitude position, it is assumed positive east of Greenwich and negative west of Greenwich (float, possibly signed)	
<Alt>	Altitude position (float, possibly signed)	
<ClkOffset>	Clock Offset of the receiver in Hz, use 0 for last saved value if available. If this is unavailable, a default value of 75000 for GSP1, 95000 for GSP1/LX will be used. (INT32)	
<TimeOfWeek>	GPS Time Of Week (UINT32)	
<WeekNo>	GPS Week Number (UINT16)	
<ChannelCount>	Number of channels to use. 1-12 (UBYTE)	
<ResetCfg>	Bit mask (UBYTE)	01=Data Valid warm/hot starts=1 02=clear ephemeris warm start=1 04=clear memory. Cold start=1

<Lat>	0 (Don't care)
<Lon>	0 (Don't care)
<Alt>	0 (Don't care)
<ClkOffset>	0 (Don't care)
<TimeOfWeek>	0 (Don't care)
<WeekNo>	0 (Don't care)
<ChannelCount>	12
<ResetCfg>	1

This message tells the receiver to start searching for position using the last values stored in memory. The <ResetCfg> field indicates the receiver to perform a warm start and the <ChannelCount> field tells device to use all 12 available channels.

Next, to decrease the traffic on the serial port, all periodic messages are disabled using the function Query/Rate Control with the following fields:

<msg>	From 0 to 5, depending on message to disable
<mode>	0 - Set rate
<rate>	0 - Off
<cksumEnable>	1 - Enable CheckSum

The message in this format disables the sending of periodic messages of all types. It is just needed to send message individually changing only the <msg> field. The GPS receiver setup process is described on figure 4.5.

Since the only message relevant for this application is the GGA message, when the value is required it is needed to request the value to the GPS receiver. The message to request for the GGA message has the following parameters:

Table 4.5: Query/control message description [15]

$PSRF103,<msg>,<mode>,<rate>,<cksumEnable>*CKSUM<CR><LF>

<msg>	0=GGA, 1=GLL, 2=GSA, 3=GSV, 4=RMC, 5=VTG
<mode>	0=SetRate, 1=Query
<rate>	Output every <rate>seconds, off=0, max=255
<cksumEnable>	0=disable Checksum, 1=Enable checksum for specified message

<msg>	0 - GGA message
<mode>	1 - Query GGA message
<rate>	0 (Don't care) - The rate is not needed
<cksumEnable>	1 - Enable checksum

Then, the device responds with a GGA message and the relevant fields can be obtained. The process is shown on figure 4.6

Serial port on the board had some problems and sometimes it started printing unknown characters and could not accept any input commands. To solve this issue the serial port was closed and reopened, starting to work normally again.

4.5 Data Acquisition Process

The figure 4.7 shows how the communications are preformed between all elements in the system in order to acquire data. The GPS receiver is initialized, then data inquire start, first for ECG and pulse rate to the BAB and then for position to the GPS receiver.

After acquiring all data, it is needed to transmit it to a base station. The acquisition and transmission of data is performed periodically and when it finishes data transmission is initiated. The next section explains how data can be transmitted to get it to the visualization platform.

4.6 Data Transmission

Transferring the data to the base station can be performed using one of the following methods:

- Through the PlugSense web service, when a wireless network is available;
- Through the wireless network created by the motes, when the first method cannot be used.

To use the first method, the PMB must be connected to a wireless network and place a XML file using an operation called "put_String_WiseML". This operations accepts a XML file with the format specified bellow:

<Message> <sensor id="A" tipo="B" dados="C|C|(...)|C" stamp="dd-mm-yyyy HH:MM:SS" /> </Message>

The parameters A, B and C are related to each sensor while the "stamp" is the acquisition time stamp with the specified format. For example, if a PMB has an id 10 and is trying to send a pulse

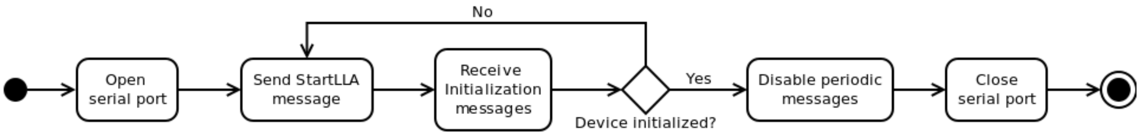


Figure 4.5: GPS receiver - Initialization process

rate with the value 88 acquired in 20-06-2012 at 15:24:12, the message to send has the following format:

<Message> <sensor id="10" tipo="PulseRate" dados="88" stamp="20-06-2012 15:24:12" /> </Message>

When the sensor data has more than one value the values must be separated with the character '|'. For example, the same PMB gets from the GPS receiver the values 41,177; -8,685; 0 relative to the latitude, longitude and altitude respectively at the same time as the pulse rate data, the message to send has the following format:

<Message> <sensor id="10" tipo="Posk" dados="41,177|-8,685|0" stamp="20-06-2012 15:24:12" /> </Message>

To send ECG data the format is the same as the ones specified except for the B parameter that should be "Ecg".

If the PMB does not have Internet access, data has to be sent using the motes. The following subsection explains how the motes were configured.

4.6.1 Motes Configuration

After studying the functioning of TinyOS[10] the motes were programed to send the data to a base station. Each mote, except the one in the base station, works as a retransmission device. If the mote receives a message from the PMB, sends it to the wireless network, if it receives a messages from another mote, it just forwards the message to another mote. This process continues until the message arrives to the base station.

The path to the base station is defined using the collection tree protocol provided by TinyOS.

4.6.1.1 Collection Tree Protocol [22]

Collection Tree Protocol (CTP) is a routing protocol for wireless sensor network. The nodes automatically calculate the minimum path (number of hops) to the closest root (destination node). A small number of roots can be defined. CTP is address free because all messages are sent to a root.

CTP uses expected transmissions (ETX) as its routing gradient. ETX of root is zero and the ETX of a node is ETX of its parent plus the ETX of its link to its parent. When a node pretends to send data to a root, it should choose the root with the lowest ETX value.

---

[10] TinyOS is an open source operative system designed specially for Wireless Sensor Network applications.

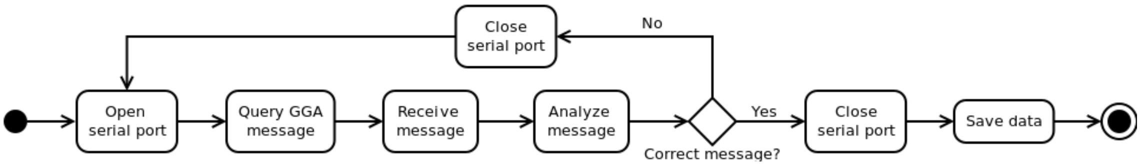


Figure 4.6: Position acquisition

A problem in a CTP is the routing loops. If a node creates a new route having a significantly higher ETX than the old one, and if the new route includes a node which was a descendant, a loop occurs.

CTP manages routing loops using two methods. Every packet has a node's current gradient value. If CTP receives a data frame with a gradient value lower than its own, then this shows that there is some error in the tree. CTP tries to solve it by broadcasting a beacon frame with the hope that the node which sent the data frame will hear and adjust its routes. CTP's second mechanism is to not consider routes with an ETX higher than a reasonable constant. The value of this constant is implementation dependent.

Another problem that can occur is packet duplication. If a message is successfully but the acknowledge was not received successfully, a retransmission will be made and the packet will be received twice. This problem may have serious consequences on the network traffic since it will exponentially increase the number of messages.

4.6.1.2 Messages

The collection interface used to send messages allows a total data size of 102 bytes. Since there are three types of data (ECG, pulse rate and position), the same amount of messages were defined to transmit using motes. These messages are characterized on table 4.6.

Table 4.6: Messages on the local network

Message	AM_TYPE (hexa)	Total data size	Fields(size)
Pulse rate	EB	3 bytes	MoteID(2 bytes) PulseRate(1 byte)
Position	EF	14 bytes	MoteID(2 bytes) Latitude(4 bytes) Longitude(4 bytes) Altitude(4 bytes)
ECG	80	98 bytes	MoteID(2 bytes) MessageSize(1 byte) ECGdata (95 bytes)

AM_TYPE is the type of the message and it works as a code to identify the message. MoteID field in all message is the ID of the PMB sending the data. This ID is defined when programing the PMB and it identifies a firefighter. The MessageSize field is needed because the ECG data

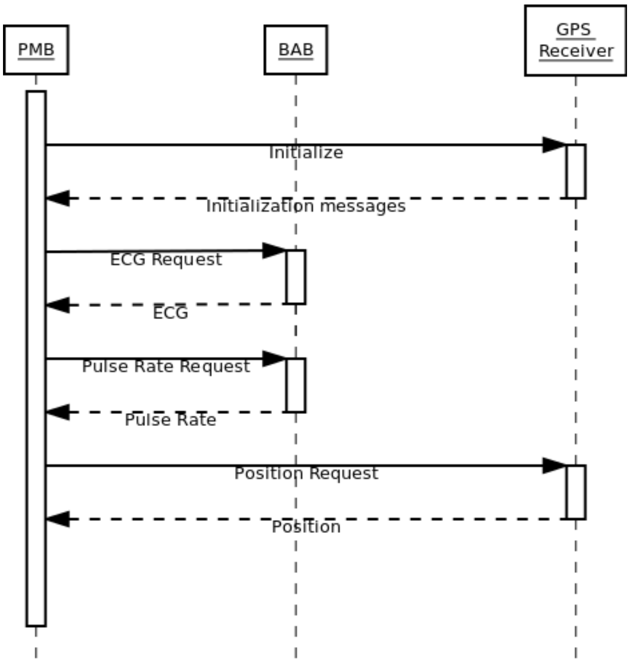


Figure 4.7: Acquisition Process

received from the BAB is greater than the maximum supported by mote messages. In order to send all data, the original ECG data had to be divided into several messages and the last message may not contain all 95 data bytes, so this field indicates the amount of bytes of ECG containing valid data. The remaining fields contain data relative to each message previously obtained.

4.6.2 Data Sending

The data sending process is shown in figure 4.8. The PMB was configured to automatically connect to a known WiFi network, if available. When data is obtained, the PMB tries to send data directly through the PlugSense web service. If this attempt fails, data is sent to the mote and it converts a serial port message to a radio message to send to the wireless network.

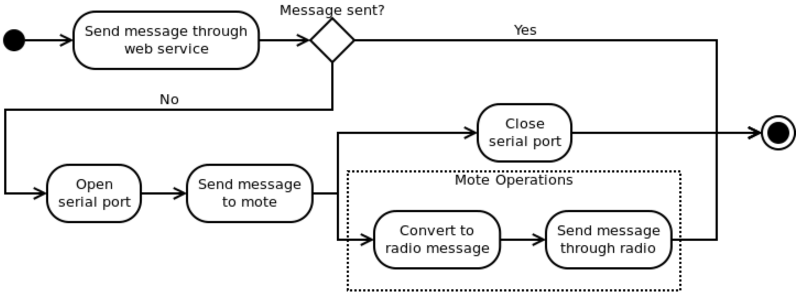


Figure 4.8: Message sending process

The mote installed on the base station perform the reverse operation. After receiving a radio message, converts it to a serial message and sends it through serial port. The receiving process is detailed on the next section.

4.7 Visualization Platform

PlugSense is platform to monitor sensors in real time. A sensor may be integrated with low effort. If that type of sensor already exists in the database, it is only needed to be associated with a certain project. The sensor can place values in database using the web service or communicating directly to a gateway. If the second option is used, a parser must be created (if it does not exist) in order to extract data from the message sent by a sensor and place it on the database.The following section explains how the parser was created.

4.7.1 Data Reception

To create the parser, the format of messages arriving through serial port must be known. According to the TinyOS documentation, serial port message of an AM_TYPE message has the format defined on table 4.7. The first line contains the message format and the second line contains a pulse rate message example sent by the mote with an id of 10 and the message has a pulse rate value of 88.

Table 4.7: Serial message

S	Prot.	Pack.	Dest.	Src.	Len.	GroupID	HandlerID	Payload	CRC	E
7E	45	00	FF FF	00 00	03	00	EB	00 0A 58	FB 52	7E

S and E values are the start and end byte of the message which is always the same (7E in hexadecimal). Next byte identifies the protocol. For the serial protocol it is 45, in hexadecimal. The following byte identifies the message type. 00 indicates it is AM_TYPE. The next two pair of bytes are the destination and the source mote. The destination, in this case, is not important because the message arrives by serial port (FF FF is the broadcast address) the source address is always 0 because the id of the mote placed in the base station is always 0. The Len. byte represents the size, in bytes, of payload field. As it was written before, the pulse rate message has a data size of 3 bytes: 2 for the moteID and 1 for the pulse rate. The GroupID is the AM group and the HandlerID identifies the message. The GroupID has always the value 00 and the HandlerID identifies the AM_TYPE that identifies the message. As referred earlier, AM_TYPE for pulse rate messages is EB in hexadecimal. The payload field is the field containing the data. After identifying the message, it is known that those values represents the moteID 10 (first two bytes) and a pulse rate value of 88 (last byte). The CRC field contains the CRC value using two bytes.

After knowing the message format, the parser was created in order to identify and extract valid data from the previous messages. The receiving process can be observed on figure 4.9.

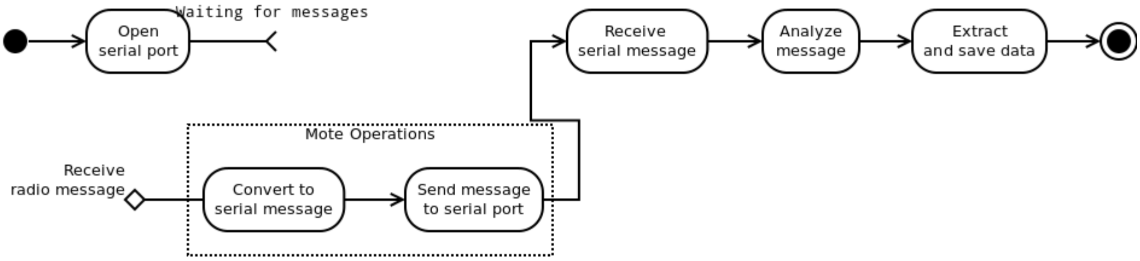
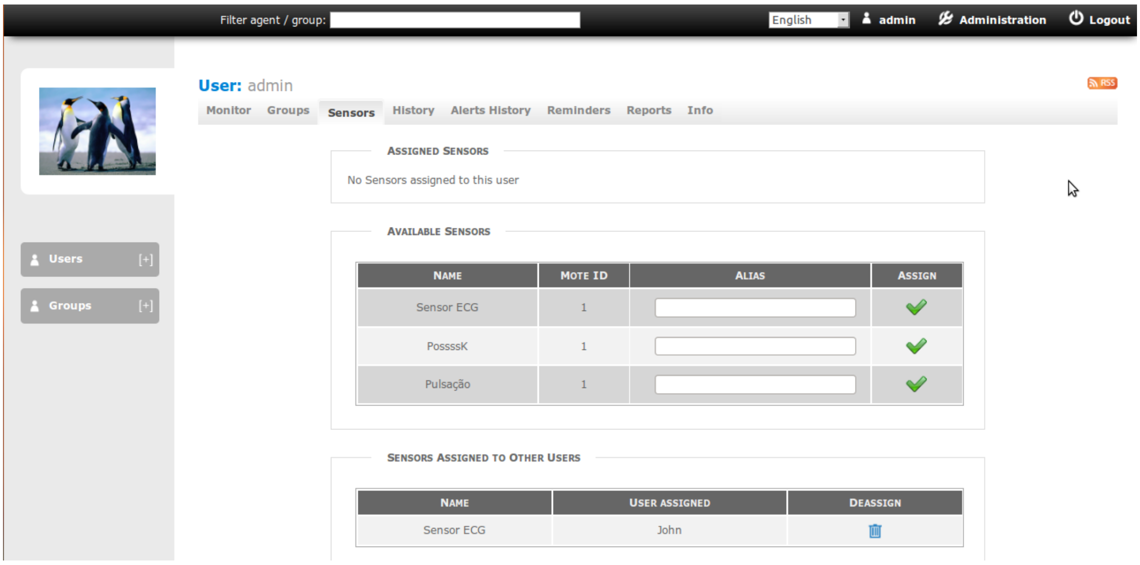


Figure 4.9: Message receiving process

4.7.2 Data Visualization

After getting all data from PMB, it is necessary to display that data in a graphical interface. PlugSense automatically does that. The only task needed, is to associate each sensor to a firefighter. This operation only need to be done once.



ASSIGNED SENSORS

No Sensors assigned to this user

AVAILABLE SENSORS

NAME	MOTE ID	ALIAS	ASSIGN
Sensor ECG	1		✓
PossssK	1		✓
Pulsação	1		✓

SENSORS ASSIGNED TO OTHER USERS

NAME	USER ASSIGNED	DEASSIGN
Sensor ECG	John	🗑

Figure 4.10: List of available sensors

When data from a new sensor is placed on the database, the user should go to the "Sensors" tab on the interface provided by PlugSense and sensors will be there unassigned. When one is assigned, data will be automatically visible in a plot or, in case of position, a map indicating firefighter position.

4.8 Process overview

After specifying all parts of the system, the whole process is now presented. Figure 4.11 presents the integration of all parts in this chapter in order to have a full system overview.

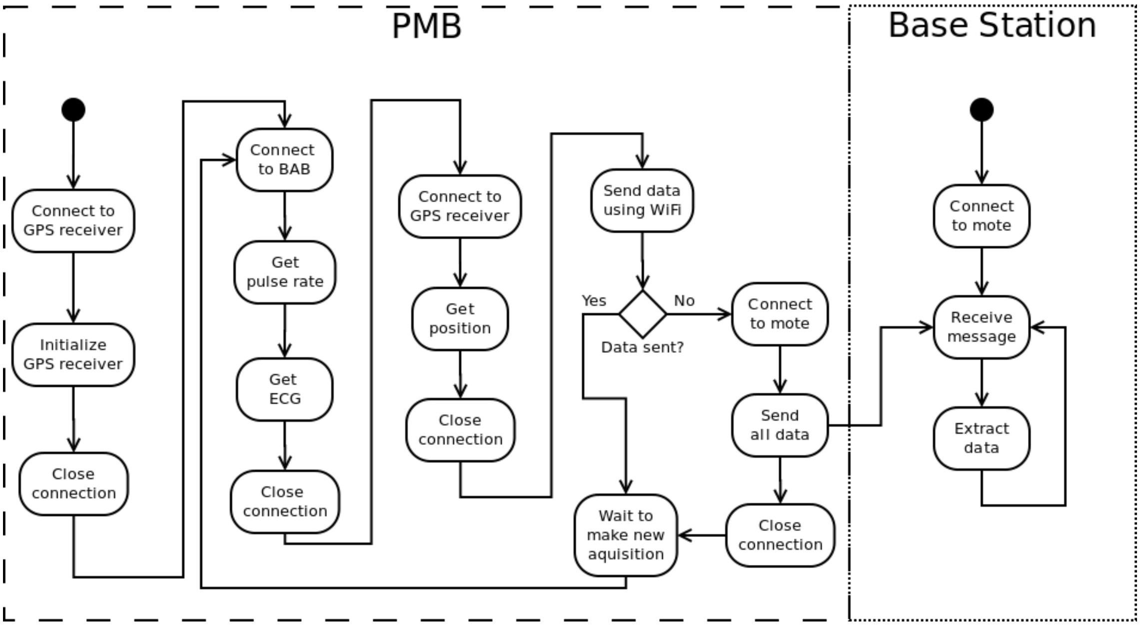


Figure 4.11: Process overview

Chapter 5

Integrations and Tests

This chapter describes a series of tests performed to verify the correct working of this system. All the described functionalities of the system were tested as presented here.

5.1 Personal Monitoring Board

The PMB has now the capability to use the following technologies: WiFi, bluetooth, GPS and mote (Zigbee) communication. Figure 5.1 shows the final appearance of the PMB.

PMB was ready to be used and to start making the final tests.

5.2 Data Acquisition and Transmission

When running the developed application on the PMB the process specified on figure 4.11 begins. Figure 5.2 shows a console output of the initializing process. An individual message is sent to disable all periodic messages and finally the start command is given. The response indicates that the GPS was successfully initiated.

Next, the connection to the BAB is made and the values of ECG and pulse rate are obtained as shown in figure 5.3.

Now, the position must be obtained. Figure 5.4 shows a successfully response of the GPS receiver. Several attempts were made (open and close serial port) and finally valid data was obtained.

To identify the error of this acquisition, the latitude and longitude were manually placed on Google Maps. Figure 5.5 shows the distance between the real position and the position obtained from the receiver. The green arrow on the left is the position obtained from GPS receiver while the green marker on the right is the real position. It can be seen that the error(based on distance measure tool from Google) is approximately 4,5 meters.

The sending process then begins. First, figure 5.6 shows the results when there is an available Internet connection and figure 5.7 when there is not.

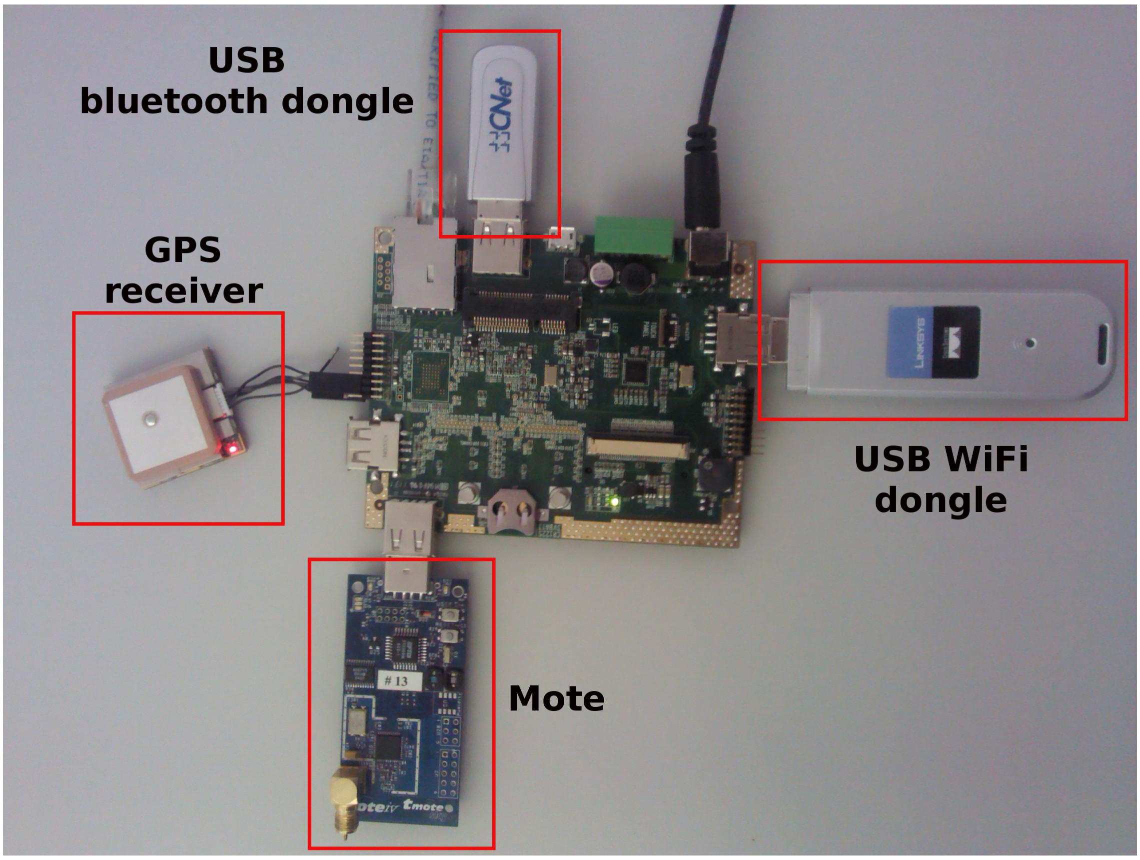


Figure 5.1: PMB appearance

When data is sent using the mote, it has to be uploaded to database. Figure 5.8 shows the program (Universal Gateway from PlugSense) running on the base station to extract data arriving through mote.

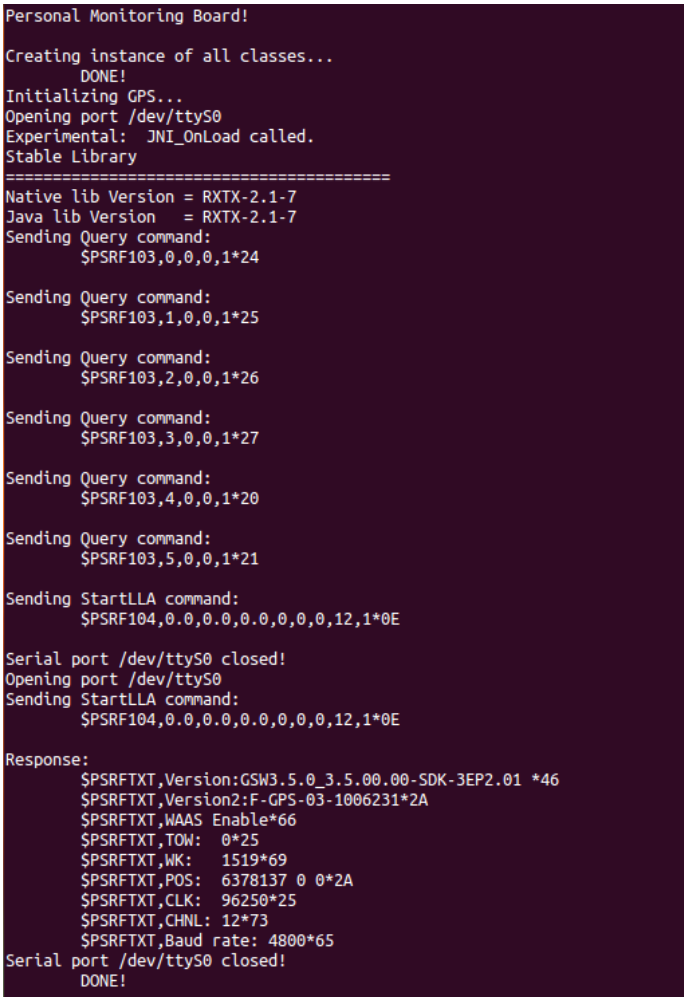
On the picture it is possible to observe that pulse rate and position are only transmitted using a single message but ECG is transmitted using three messages due to its size.

It is verified that both data acquisition and transmission can be successfully performed. Data can be observed using PlugSense as shown in next section.

5.3 Data Visualization - PlugSense

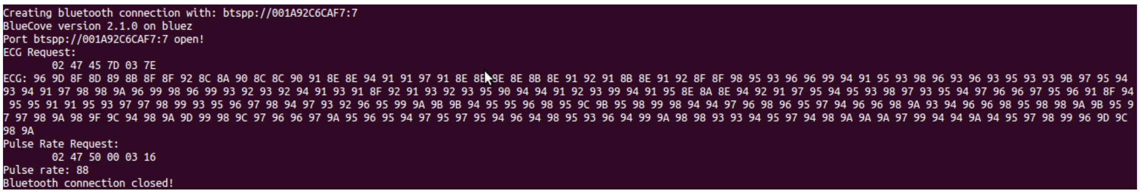
When data from a sensor is placed on database, a sensor value is immediately updated without user interface. If a new sensor is added, that sensor appears unassigned waiting to be assigned to a user. Figure 5.9 shows a list of available sensors. Sensors with Mote ID 1 are provided by the PMB. The others were created manually only for tests.

After creating two more users, Albert and John, the sensors were assigned. Sensors with mote ID 1 were assigned to user admin, those with 10 were assigned to John and 15 to Albert. Sensor 0 was left unassigned.



```
Personal Monitoring Board!

Creating instance of all classes...
        DONE!
Initializing GPS...
Opening port /dev/ttyS0
Experimental:  JNI_OnLoad called.
Stable Library
=========================================
Native lib Version = RXTX-2.1-7
Java lib Version   = RXTX-2.1-7
Sending Query command:
        $PSRF103,0,0,0,1*24

Sending Query command:
        $PSRF103,1,0,0,1*25

Sending Query command:
        $PSRF103,2,0,0,1*26

Sending Query command:
        $PSRF103,3,0,0,1*27

Sending Query command:
        $PSRF103,4,0,0,1*20

Sending Query command:
        $PSRF103,5,0,0,1*21

Sending StartLLA command:
        $PSRF104,0.0,0.0,0.0,0,0,0,12,1*0E

Serial port /dev/ttyS0 closed!
Opening port /dev/ttyS0
Sending StartLLA command:
        $PSRF104,0.0,0.0,0.0,0,0,0,12,1*0E

Response:
        $PSRFTXT,Version:GSW3.5.0_3.5.00.00-SDK-3EP2.01 *46
        $PSRFTXT,Version2:F-GPS-03-1006231*2A
        $PSRFTXT,WAAS Enable*66
        $PSRFTXT,TOW:  0*25
        $PSRFTXT,WK:   1519*69
        $PSRFTXT,POS:  6378137 0 0*2A
        $PSRFTXT,CLK:  96250*25
        $PSRFTXT,CHNL: 12*73
        $PSRFTXT,Baud rate: 4800*65
Serial port /dev/ttyS0 closed!
        DONE!
```

Figure 5.2: Initialization process

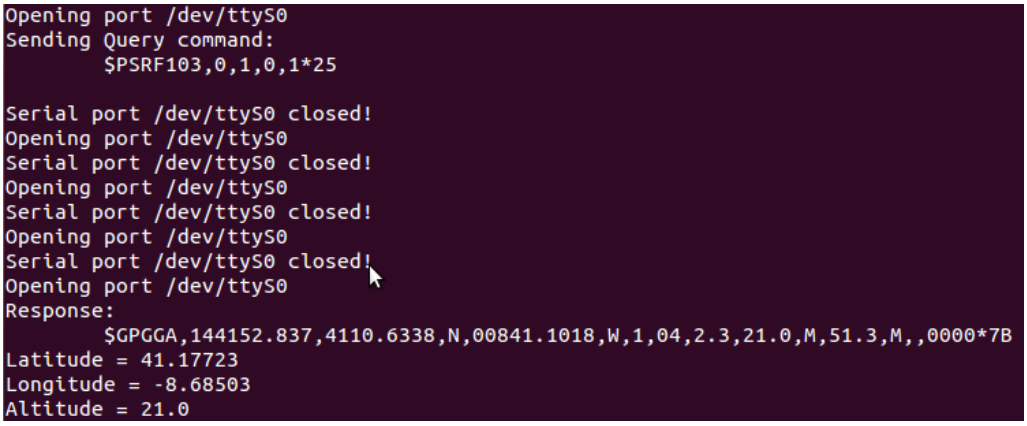


```
Creating bluetooth connection with: btspp://001A92C6CAF7:7
BlueCove version 2.1.0 on bluez
Port btspp://001A92C6CAF7:7 open!
ECG Request:
        02 47 45 7D 03 7E
ECG: 96 9D 8F 8D 89 8B 8F 8F 92 8C 8A 90 8C 8C 90 91 8E 8E 94 91 91 97 91 8E 8E 8E 8E 8B 8E 91 92 91 8B 8E 91 92 8F 8F 98 95 93 96 96 99 94 91 95 93 98 96 93 96 93 95 93 93 9B 97 95 94
93 94 91 97 98 98 9A 96 99 98 96 99 93 92 93 92 94 91 93 91 8F 92 91 93 92 93 95 90 94 94 91 92 93 99 94 91 95 8E 8A 8E 94 92 91 97 95 94 95 93 98 97 93 95 94 97 96 96 97 95 96 91 8F 94
 95 95 91 91 95 93 97 97 98 99 93 95 96 97 98 94 97 93 92 96 95 99 9A 9B 9B 94 95 95 96 98 95 9C 9B 95 98 99 98 94 94 97 96 98 96 95 97 94 96 96 98 9A 93 94 96 96 98 95 98 98 9A 9B 95 9
7 97 98 9A 98 9F 9C 94 98 9A 9D 99 98 9C 97 96 96 97 9A 95 96 95 94 97 95 97 95 94 96 94 98 95 93 96 94 99 9A 98 98 93 93 94 95 97 94 98 9A 9A 9A 97 99 94 94 9A 94 95 97 98 99 96 9D 9C
98 9A
Pulse Rate Request:
        02 47 50 00 03 16
Pulse rate: 88
Bluetooth connection closed!
```

Figure 5.3: BAB acquisition

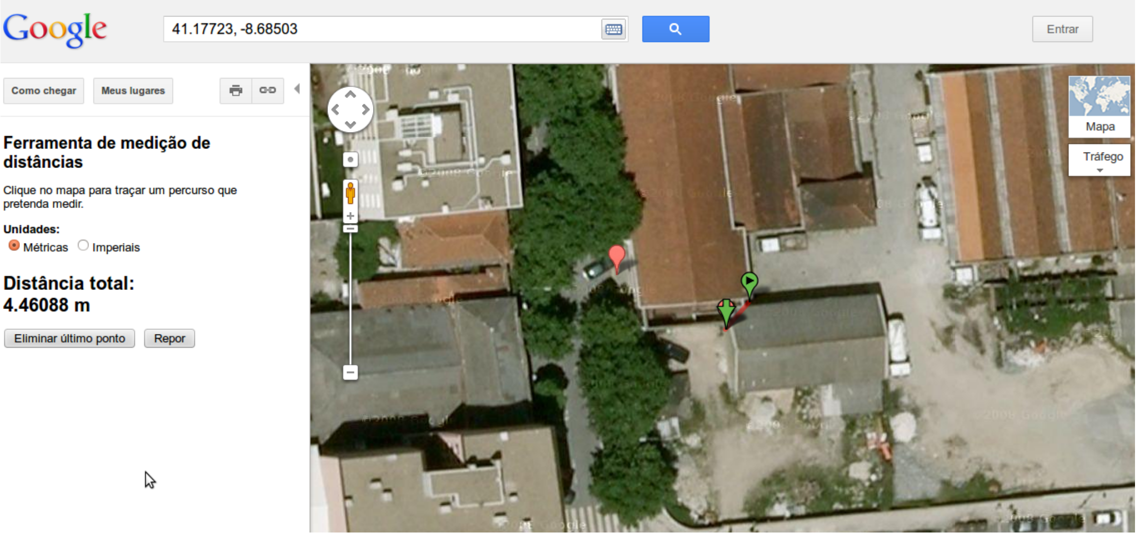
PlugSense allows to create various groups of users. The group Operation 1 was created and the three users referred on the previous paragraph were assigned to it as it can be seen in picture 5.10.

The "Monitor" tab shows the vital data of the user selected. In figure 5.11 it is possible to view data of user admin. This data is updated automatically. On top of same figure it is possible to see an alarm that was fired when a very low pulse rate value of John was received. That value was sent manually to verify if this particular functionality was working correctly.



```
Opening port /dev/ttyS0
Sending Query command:
        $PSRF103,0,1,0,1*25

Serial port /dev/ttyS0 closed!
Opening port /dev/ttyS0
Serial port /dev/ttyS0 closed!
Opening port /dev/ttyS0
Serial port /dev/ttyS0 closed!
Opening port /dev/ttyS0
Serial port /dev/ttyS0 closed!
Opening port /dev/ttyS0
Response:
        $GPGGA,144152.837,4110.6338,N,00841.1018,W,1,04,2.3,21.0,M,51.3,M,,0000*7B
Latitude = 41.17723
Longitude = -8.68503
Altitude = 21.0
```

Figure 5.4: A successfully position response by GPS receiver.



41.17723, -8.68503

Ferramenta de medição de distâncias

Clique no mapa para traçar um percurso que pretenda medir.

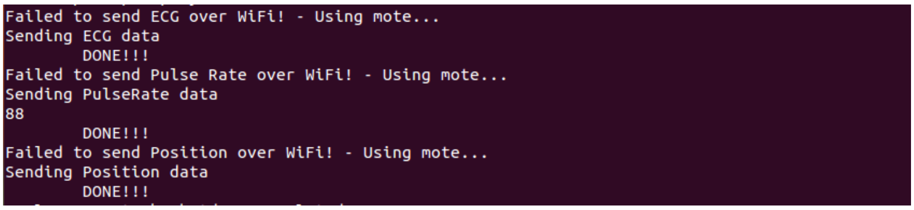
Unidades:
● Métricas ○ Imperiais

Distância total:
4.46088 m

Figure 5.5: Error obtained using Google Maps [12]

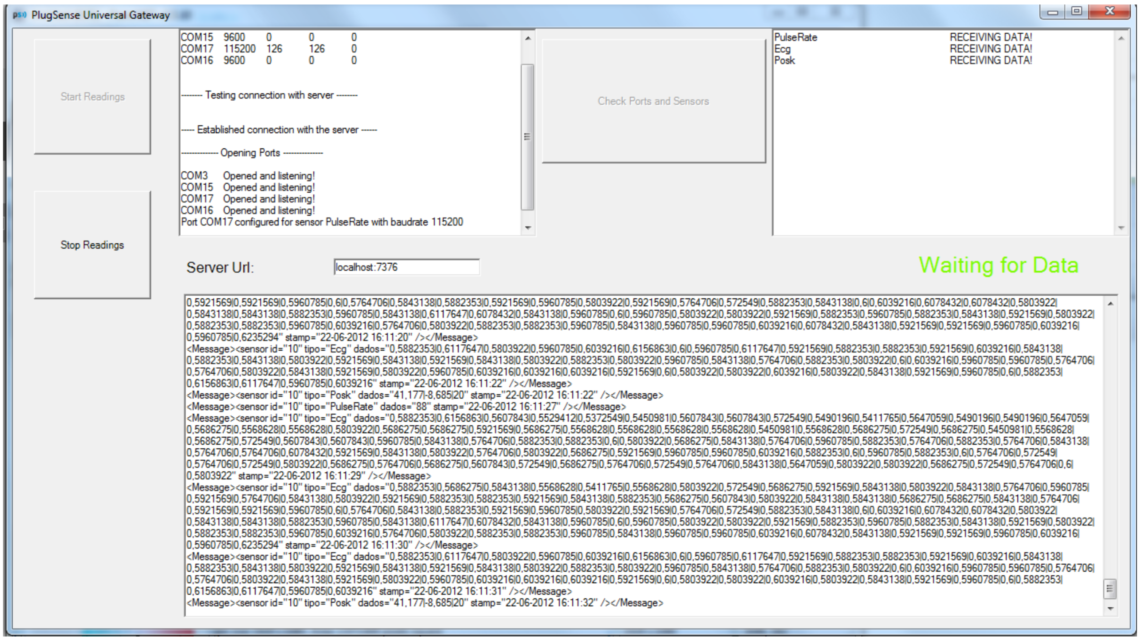
```
ECG sent using WiFi
Pulse Rate sent using WiFi
Position sent using WiFi
```

Figure 5.6: A console output extract corresponding to a successfully sent message over WiFi



```
Failed to send ECG over WiFi! - Using mote...
Sending ECG data
        DONE!!!
Failed to send Pulse Rate over WiFi! - Using mote...
Sending PulseRate data
88
        DONE!!!
Failed to send Position over WiFi! - Using mote...
Sending Position data
        DONE!!!
```

Figure 5.7: A console output extract corresponding to a successfully sent message to the mote



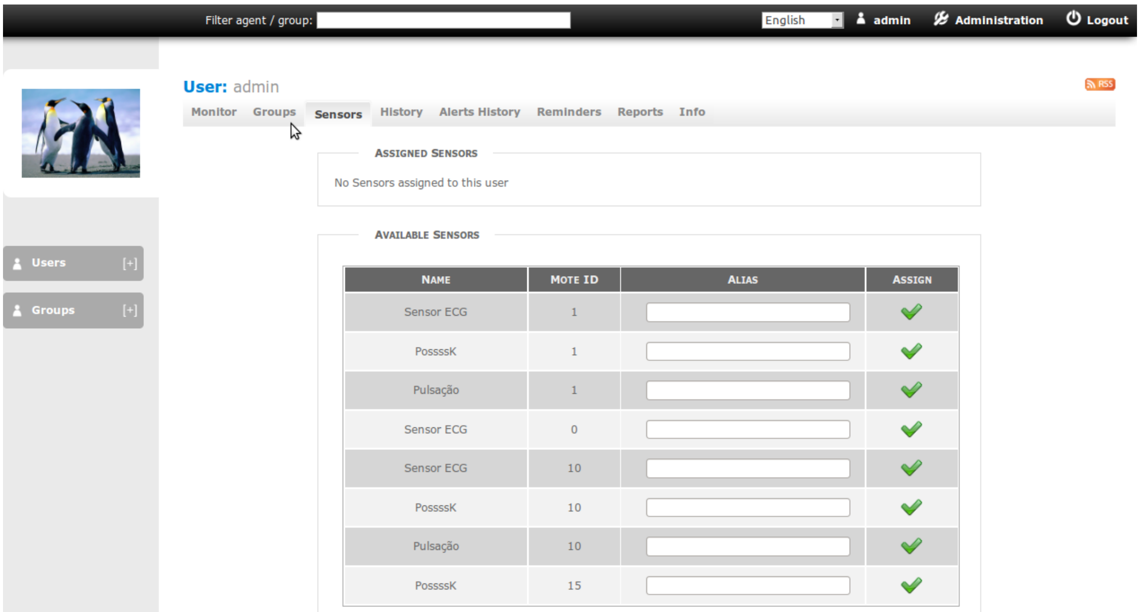
```
COM15   9600     0       0       0
COM17   115200   126     126     0
COM16   9600     0       0       0


-------- Testing connection with server --------


----- Established connection with the server ------

------------- Opening Ports --------------

COM3     Opened and listening!
COM15    Opened and listening!
COM17    Opened and listening!
COM16    Opened and listening!
Port COM17 configured for sensor PulseRate with baudrate 115200
```

Server Url: localhost:7376

```
PulseRate                                RECEIVING DATA!
Ecg                                      RECEIVING DATA!
Posk                                     RECEIVING DATA!
```

```
0,5921569|0,5921569|0,5960785|0,6|0,5764706|0,5843138|0,5882353|0,5921569|0,5960785|0,5803922|0,5921569|0,5764706|0,572549|0,5882353|0,5843138|0,6|0,6039216|0,6078432|0,6078432|0,5803922|
0,5843138|0,5843138|0,5882353|0,5960785|0,5843138|0,6117647|0,6078432|0,5843138|0,5960785|0,6|0,5960785|0,5803922|0,5803922|0,5921569|0,5882353|0,5960785|0,5882353|0,5843138|0,5921569|0,5803922|
0,5882353|0,5882353|0,5960785|0,6039216|0,5764706|0,5803922|0,5882353|0,5882353|0,5960785|0,5843138|0,5960785|0,5960785|0,6039216|0,6078432|0,5843138|0,5921569|0,5921569|0,5960785|0,6039216|
0,5960785|0,6235294" stamp="22-06-2012 16:11:20" /></Message>
<Message><sensor id="10" tipo="Ecg" dados="0,5882353|0,6117647|0,5803922|0,5960785|0,6039216|0,6156863|0,6|0,5960785|0,6117647|0,5921569|0,5882353|0,5882353|0,5921569|0,6039216|0,5843138|
0,5882353|0,5843138|0,5803922|0,5921569|0,5843138|0,5921569|0,5843138|0,5803922|0,5882353|0,5803922|0,5960785|0,5843138|0,5764706|0,5882353|0,5803922|0,6|0,6039216|0,5960785|0,5960785|0,5764706|
0,5764706|0,5803922|0,5843138|0,5921569|0,5803922|0,5960785|0,6039216|0,6039216|0,6039216|0,5921569|0,6|0,5803922|0,5803922|0,6039216|0,5803922|0,5843138|0,5921569|0,5960785|0,6|0,5882353|
0,6156863|0,6117647|0,5960785|0,6039216" stamp="22-06-2012 16:11:22" /></Message>
<Message><sensor id="10" tipo="Posk" dados="41,177|-8,685|20" stamp="22-06-2012 16:11:22" /></Message>
<Message><sensor id="10" tipo="PulseRate" dados="88" stamp="22-06-2012 16:11:27" /></Message>
<Message><sensor id="10" tipo="Ecg" dados="0,5882353|0,6156863|0,5607843|0,5529412|0,5372549|0,5450981|0,5607843|0,5607843|0,572549|0,5490196|0,5411765|0,5647059|0,5490196|0,5490196|0,5647059|
0,5686275|0,5568628|0,5568628|0,5803922|0,5686275|0,5686275|0,5921569|0,5686275|0,5568628|0,5568628|0,5568628|0,5568628|0,5450981|0,5568628|0,5686275|0,572549|0,5686275|0,5450981|0,5568628|
0,5686275|0,572549|0,5607843|0,5607843|0,5960785|0,5843138|0,5764706|0,5882353|0,5882353|0,6|0,5803922|0,5686275|0,5843138|0,5764706|0,5960785|0,5882353|0,5764706|0,5882353|0,5764706|0,5843138|
0,5764706|0,5764706|0,6078432|0,5921569|0,5843138|0,5803922|0,5764706|0,5803922|0,5686275|0,5921569|0,5960785|0,5960785|0,6039216|0,5882353|0,6|0,5960785|0,5882353|0,6|0,5764706|0,572549|
0,5764706|0,572549|0,5803922|0,5686275|0,5764706|0,5686275|0,5607843|0,572549|0,5686275|0,5764706|0,572549|0,5764706|0,5843138|0,5647059|0,5803922|0,5803922|0,5686275|0,572549|0,5764706|0,6|
0,5803922" stamp="22-06-2012 16:11:29" /></Message>
<Message><sensor id="10" tipo="Ecg" dados="0,5882353|0,5686275|0,5843138|0,5568628|0,5411765|0,5568628|0,5803922|0,572549|0,5686275|0,5921569|0,5843138|0,5803922|0,5843138|0,5764706|0,5960785|
0,5921569|0,5764706|0,5843138|0,5803922|0,5921569|0,5882353|0,5882353|0,5921569|0,5843138|0,5882353|0,5686275|0,5607843|0,5803922|0,5843138|0,5843138|0,5686275|0,5686275|0,5843138|0,5764706|
0,5921569|0,5921569|0,5960785|0,6|0,5764706|0,5843138|0,5882353|0,5921569|0,5960785|0,5803922|0,5921569|0,5764706|0,572549|0,5882353|0,5843138|0,6|0,6039216|0,6078432|0,6078432|0,5803922|
0,5843138|0,5843138|0,5882353|0,5960785|0,5843138|0,6117647|0,6078432|0,5843138|0,5960785|0,6|0,5960785|0,5803922|0,5803922|0,5921569|0,5882353|0,5960785|0,5882353|0,5843138|0,5921569|0,5803922|
0,5882353|0,5882353|0,5960785|0,6039216|0,5764706|0,5803922|0,5882353|0,5882353|0,5960785|0,5843138|0,5960785|0,5960785|0,6039216|0,6078432|0,5843138|0,5921569|0,5921569|0,5960785|0,6039216|
0,5960785|0,6235294" stamp="22-06-2012 16:11:30" /></Message>
<Message><sensor id="10" tipo="Ecg" dados="0,5882353|0,6117647|0,5803922|0,5960785|0,6039216|0,6156863|0,6|0,5960785|0,6117647|0,5921569|0,5882353|0,5882353|0,5921569|0,6039216|0,5843138|
0,5882353|0,5843138|0,5803922|0,5921569|0,5843138|0,5921569|0,5843138|0,5803922|0,5882353|0,5803922|0,5960785|0,5843138|0,5764706|0,5882353|0,5803922|0,6|0,6039216|0,5960785|0,5960785|0,5764706|
0,5764706|0,5803922|0,5843138|0,5921569|0,5803922|0,5960785|0,6039216|0,6039216|0,6039216|0,5921569|0,6|0,5803922|0,5803922|0,6039216|0,5803922|0,5843138|0,5921569|0,5960785|0,6|0,5882353|
0,6156863|0,6117647|0,5960785|0,6039216" stamp="22-06-2012 16:11:31" /></Message>
<Message><sensor id="10" tipo="Posk" dados="41,177|-8,685|20" stamp="22-06-2012 16:11:32" /></Message>
```

Figure 5.8: PlugSense Universal Gateway running on the base station.



User: admin

ASSIGNED SENSORS

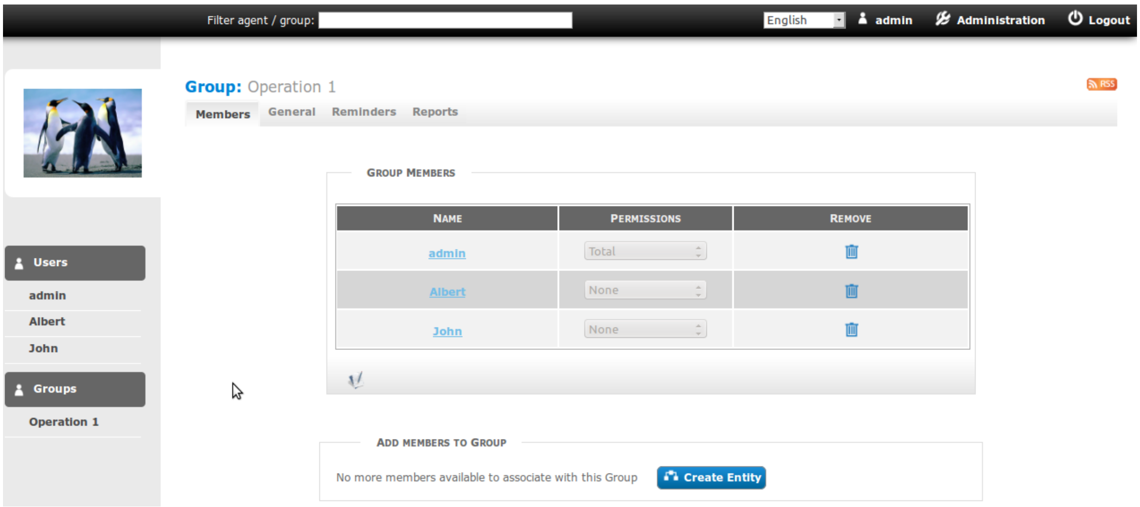
No Sensors assigned to this user

AVAILABLE SENSORS

NAME	MOTE ID	ALIAS	ASSIGN
Sensor ECG	1		✓
PossssK	1		✓
Pulsação	1		✓
Sensor ECG	0		✓
Sensor ECG	10		✓
PossssK	10		✓
Pulsação	10		✓
PossssK	15		✓

Figure 5.9: PlugSense interface - Unassigned sensors. The commander can access this tab and assign sensors to a firefighter.

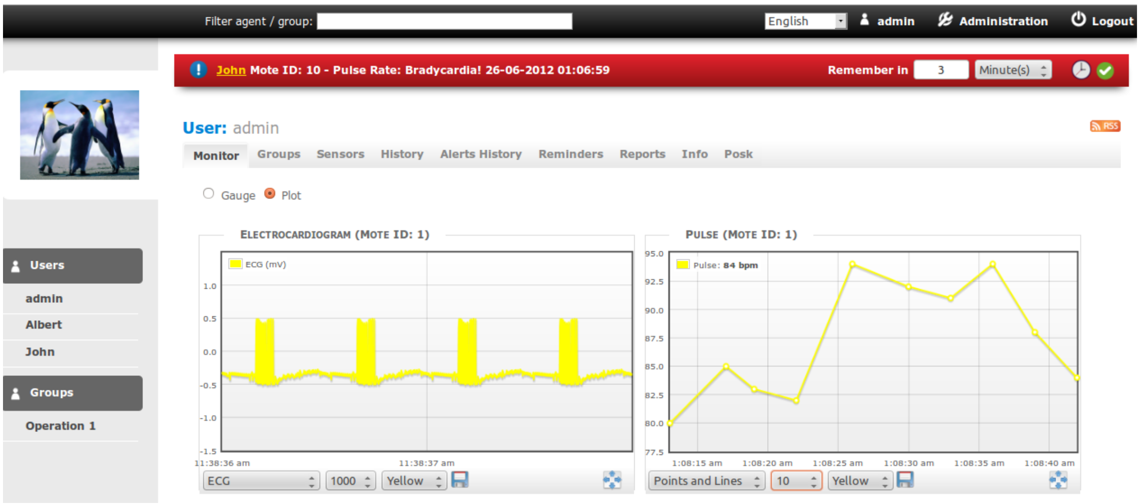
To view the current position of a firefighter the tab "Posk" must be selected. In this tab it is possible to view not only the current position of the user selected but also all users belonging to a group, in this case the group Operation 1. Figure 5.12 shows an example of the described situation. Once again, the positions of the other firefighters were set manually for test purposes and do not correspond to real locations. This tab also allows to follow a certain user each time he changes his



NAME	PERMISSIONS	REMOVE
admin	Total	🗑
Albert	None	🗑
John	None	🗑

No more members available to associate with this Group

Figure 5.10: PlugSense interface - Users belonging to group Operation 1. The commander has the possibility to add more users and/or groups and assign firefighters to groups.



John Mote ID: 10 - Pulse Rate: Bradycardia! 26-06-2012 01:06:59

Remember in: 3 Minute(s)

○ Gauge ● Plot

Pulse: 84 bpm

ECG 1000 Yellow

Points and Lines 10 Yellow

Figure 5.11: PlugSense interface - Monitor tab with ECG and Pulse Rate. An alarm is shown on top of page. The commander is viewing admin data but the alarm related to John shows up on top of the page.

position by checking the box "Follow User" right on top of the location window.

5.4 Conclusions

The previous tests show that data can be obtained and sent to the base station successfully. However, the effectiveness of the routing protocol could not be tested intensively and cannot be guaranteed that messages can be transmitted in a highly dynamic scenario. This issue was only tested by moving one mote at a time just to ensure that the network would be automatically reconfigured.

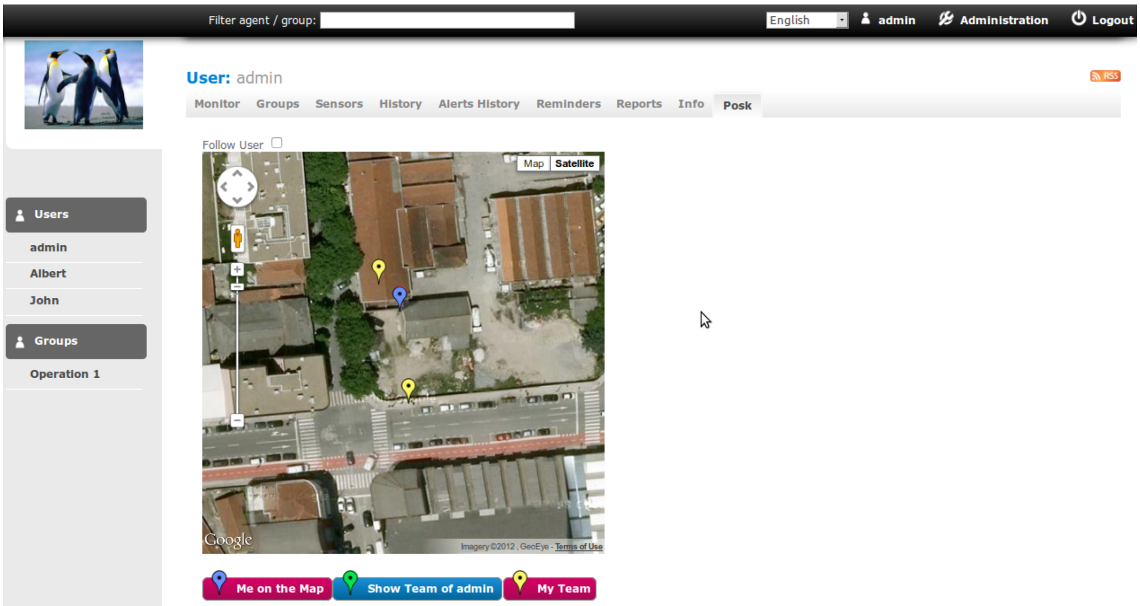


Figure 5.12: PlugSense interface - Posk tab with showing the position of all team members.

This tests demonstrate that this system can be very useful tool to support commanders on their decision since they have a total overview of the field and the position of firefighters as well as their physical condition. PlugSense interface is simple and easy to use as demonstrated.

Chapter 6

Conclusions

With the recent advances in technologies, it is now possible to introduce any kind of sensor in a network with a relative low cost. The great evolution of wireless sensor network has a huge importance in monitoring systems. This project proved that a great variety of sensors can be integrated in a wireless network.

The main goal of this project was to create a reliable wireless network capable of monitoring a whole team of firefighters to help them on their operations.

The vital signals were successfully obtained using a reliable bluetooth connection. That fact allows firefighters to keep their mobility since there is no need to make any kind of wired connection that could disturb the firefighter. At this point, BAB only measures the ECG and the pulse rate which are very important parameters to know the vital status of a firefighter. However, additional sensors can be easily added to the PMB.

The positioning of the firefighters was also obtained successfully. The integration of a GPS receiver was a complicated task, not due to working mode of the device but mainly due to the serial port instability. When trying to communicate with a device in order to understand its behavior, the serial port was constantly printing unknown characters until it was reinitialized. This fact forced to create a program with a greater processing capacity in order to solve this issue. This system only allows to obtained position when a firefighter is in a place with a good sky visibility. An alternative solution must be studied when the GPS is impossible to use since that was not implemented in this project.

Another great obstacle of this project was the use of motes. TinyOS had to be study from the very basics. Nevertheless, the implementation of the collection tree protocol was successfully achieved and even if firefighters changed their disposition, it was still able to correctly delivered the messages to the root, after automatically rebuilding the tree. However, the protocol needs to be tested in a dynamic environment to evaluate its real performance. The other great obstacle related with motes was the integration of them with the board. It was needed to recompile the kernel and some libraries so that motes can work correctly, which took a lot of time.

PlugSense proved to be a very useful tool for data visualization, since it was possible to view data in real time and easily add a new sensor to an application. It also provides the capability to view data everywhere needing only an Internet connection and a web browser.

The tests performed to the system proved the correct functioning of all process and it could be shown that this system could be an important tool to firefighters commanders.

The purposed goals were successfully achieved and the system was build successfully.

6.1 Future Work and Improvements

With the network created and the communication established there are still a number of improvements that can be done. The PMB has the capability to integrate more sensors, using a bluetooth connection or even directly connected to it, to measure not only the physical condition of a firefighter, but also environmental parameters to help decision taking.

The functionalities of the PMB can also be explored in order to introduce new ones on the system. For example: the buzzer integrated on the board can be used to sound an alarm each time a firefighter needs help in order to alert the remaining team or even for a leader to guide a team using its sound when the visibility is low. Another possible functionality to be introduced is the support for GSM networks since the board already has a SIM socket. A touchscreen can also be added to allow firefighters to share information between them and the commander, such as maps, directions, orders (advance, retrieve, warnings) etc.

Another improvement could be the use of motes for more than just creating a network. There are motes with all kind of sensors included that can be used to monitor environmental data. They can also be used to obtained relative position of the elements do determine the position when GPS can not be used.

The base of this system can be used for other applications. The PMB can be assigned to each patient in a hospital so that doctors can visualize the state of their patient at their office or even at home since the PlugSense can be accessed from everywhere. PlugSense can also be used to sound an alarm when a patient enters in a critical state alerting the doctor.

Another application of this system could be to monitor elderly people living alone. Their family can see if everything is fine using the PlugSense web application.

Appendix A

PlugSense[7]

A.1 Description

PlugSense is a product designed to measure, monitor and control wireless sensor networks. It has a set of services to allow data visualization anytime and anywhere. It also provides several interfaces on order to receive data from all kind of sensors and store them in database. This document present how does this product works.

A.2 Architecture

PlugSense architecture is shown in the diagram A.1.

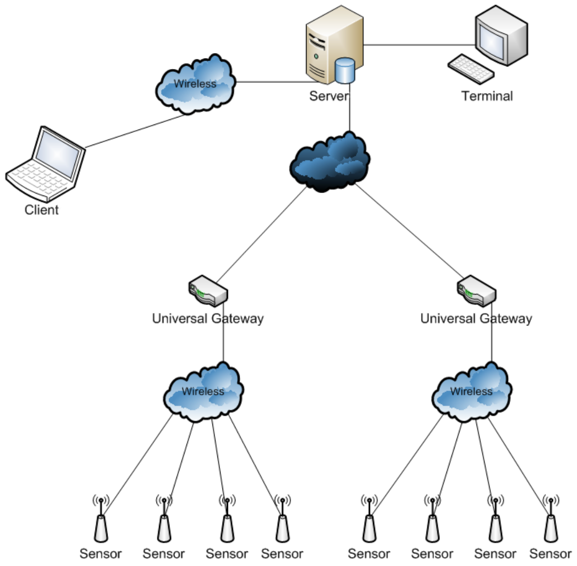


Figure A.1: PlugSense architecture

The server is the machine responsible for the application working, it stores all data received through the Universal Gateway and from the clients. The web server is responsible for processing request from clients, presenting the results through a set of web pages.

The client is the computer in which the user can visualize and interact with the application running on server using a browser with Javascript, CSS and XHTML compatibility.

The Universal Gateway is a device that bridges sensors to server.

A.3 Sensor Data Reception

Each sensor has an associated model with the following fields:

- ID - Key identifier on database;
- Data - Data from sensor;
- ReadTime - Time and date of reading.

These fields correspond to a sensor measure. Data field can be divided into other fields depending on data complexity. For example, a GPS sensor has, instead of Data, Latitude, Longitude and Altitude.

There is also a controller capable of transforming data into a plot.

A.4 Universal Gateway

Universal Gateway is a data reception system for wireless sensors. This system can be composed by one or more sensors to which are connected wireless receptors like Bluetooth, RF, Zigbee, among others. The objective of this system is to bridge sensors to the main server. Data are received and converted into a standard XML format file.

A.5 PlugSense Server

PlugSense Server is a web service that contains several methods to input data on database. PlugSense web service can be used to send data to the main server, manage projects, activate or deactivate the reception of data through a mote and manage alerts.

A.6 Functional Diagram

Figure A.2 shows the functioning of the whole PlugSense system.

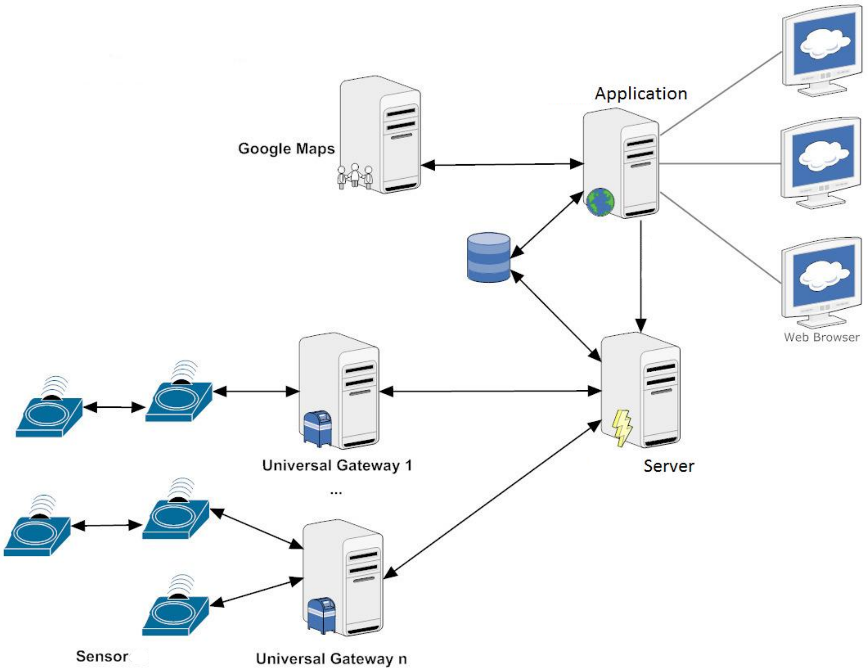


Figure A.2: PlugSense Functional diagram

Appendix B

Biosignals Acquisition Board

B.1 Description

The Biosignals Acquisition Board, is a small board that can measure ECG and pulse rate, and communicate using a bluetooth connection.

B.2 Architecture

Figure B.1 shows the architecture of the BAB.

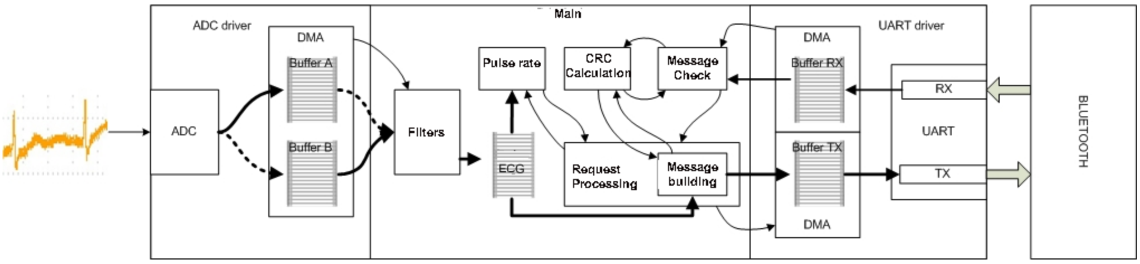


Figure B.1: BAB Architecture [13]

On the figure there are three parts ADC driver, Main and UART driver. ADC and UART drivers are independent of the main program.

ADC makes the sample and conversion of the ECG data and stores data into one of the buffers. When a buffer is full, it activates a flag and starts filling the other buffer with data. The main program notices the flag and starts processing data (filtering). When UART drivers receives a request over bluetooth it saves the request and activate another flag to notify the main program that a request arrived. The main program reads the request buffer and starts to fill the response buffer and activates the sending process. The UART driver is responsible to transmit data over bluetooth using the RFCOMM service.

References

[1] Mukherjee B. Ghosal D. Yick, J. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[2] GPS.gov. Official u.s. government information about the global positioning system (gps) and related topics. Internet: `http://www.gps.gov/`, May 2012.

[3] Gellersen H. Fischer, C. Location and navigation support for emergency responders: A survey. *IEEE Pervasive Computing*, 9(1):38–47, 2010.

[4] Garbi An. Calarco G. Casoni M. Paganelli A. Morera R. Chen C.-M. Wódczak M. Vassiliadis, D. Wireless networks at the service of effective first response work: The e-sponder vision. pages 210–214, 2010.

[5] Ochoa S.F. Pinoo J.A. Herskovic V. Neyem A. Monares, A. Mobilemap: A collaborative application to support emergency situations in urban areas. pages 432–437, 2009.

[6] K. Betke. The nmea 0183 protocol. *National Marine Electronics Association (NMEA), New Bern, USA*, 2000.

[7] FreedomGrow. Plugsense - architecture report. Internet: `http://www.freedomgrow.pt/`, June 2012.

[8] Jonathan Hui David E. Culler. Eecs-194 sp08. Internet: `http://inst.eecs.berkeley.edu/~cs194-5/sp08/lab1/`, June 2012.

[9] Future Electronics Egypt Ltd. (Arduino Egypt). Comparison of wireless technologies.

[10] MiniBox.com. Mini-box.com pico wiki. Internet: `http://arm.mini-box.com`, April 2012.

[11] SparkFun Electronics. Sparkfun electronics. Internet: `http://www.sparkfun.com/`, April 2012.

[12] Google. Google maps. Internet: `http://maps.google.com/`, June 2012.

[13] João Pedro Sousa Oliveira. *Desenvolvimento e integração de sensores numa plataforma para sistemas de monitorização pessoais*. 2012.

[14] Inc. SiRF Technology. Nmea reference manual, 2005.

[15] GlobalSat. Em-406a datasheet.

[16] Shi W. Watkins O. Sha, K. Using wireless sensor networks for fire rescue applications: Requirements and challenges. pages 239–244, 2006.

[17] INOVA+. Serviços de consultoria e gestão. Internet: `http://www.inovamais.eu`, June 2012.

[18] Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[19] Inc. Bluetooth SIG. Official bluetooth technology website. Internet: `http://www.bluetooth.com/Pages/Bluetooth-Home.aspx`, April 2012.

[20] Pillai P. Chook V.W.C. Chessa S. Gotta A. Hu Y.F. Baronti, P. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655–1695, 2007.

[21] Wang X. Wang W. Akyildiz, I.F. Wireless mesh networks: A survey. *Computer Networks*, 47(4):445–487, 2005.

[22] TinyOS. Tinyos. Internet: `http://www.tinyos.net/`, May 2012.