UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Bluetooth/Zigbee Bridge for Mobile Phone based Wireless Sensor Network Gateways

TITULACION: Sistemas de Telecomunicación

TUTORES: Dr. Alex Gluhak [1]

Dr. Javier del Pino [2]

AUTOR: Loren Alejandra Moreno Cabrera

FECHA: Julio 2011

[1] University of Surrey (UK)

[2] Universidad de Las Palmas de Gran Canaria (Spain)

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The following project has been carried out in the Department of Centre for Communication Systems Research at the University of Surrey, thanks to the agreement between University of Surrey (UK) and Universidad de Las Palmas de Gran Canaria (Spain) inside the Socrates/Erasmus Program.

Also has special acknowledgment to the University of Las Palmas de Gran Canaria and Gobierno de Canarias because they funded the period of time in England and were responsible for managing the grant.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Abstract

The Wireless Sensor Network has been expanding over recent years, now growing in different applications, both commercial (automation, industry, medicine) and military.

WSNs are composed of independent sensors nodes, and distributed evenly, able to interact with their environment, which means they can get information depending on their use or application. Usually the function has focused on monitoring physical conditions or the environment. Nodes of sensors networks have no predetermined structure; mesh networks are generally used, allowing access through different routes, giving alternative routes between origin and destination in case of obstruction. Imagine a network of sensor nodes interacting with their environment to get data from each sensor node. To obtain this information from the sensors to the 'real world', they need to incorporate a gateway that allows or provides connectivity between the network of wireless sensors and the host application through an interface radio properly. This device will get the data available at each sensor independently. In general, these technologies are based on the Zigbee standard (IEEE 802.15.4), and are optimized to transfer data at lower speeds with low power consumption and adding performance.

Today virtually all mobile phone handsets incorporate Bluetooth technology for data transfer, as the most common hands free link. So, the idea of this project is to design a gateway able to provide access to users from the mobile terminal through Bluetooth technology to the network of sensors and incorporating the radio-based technology Zigbee needs to communicate with nodes in the sensor network; being able to independently obtain data from each of them and provide such data through Bluetooth to a mobile phone terminal.

The design will also be provided to develop the necessary drivers to work correctly with the network of sensors. This device finally incorporates Bluetooth technology, so it has developed as well as the drivers that enable the Bluetooth module to connect to any device and transfer data, in turn incorporating the radio interface based on Zigbee. For us the platform used is Sunspot, based on Java programming.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Acknowledgements

First and foremost, I would like to thanks to my boyfriend David for giving me the unconditional support, love and be patient during all these months. I really appreciate it.

Then I offer my sincerest gratitude to my supervisor, Dr Alex Gluhak, who has teaching me since I came to this University, has giving me new challenges to developing in each step of this project, becoming it an experience in my life, gave me another point of view of telecommunications and what's happens behind of them, and opened my mind in new perspective about technologies as well as prepare me to work in the real world.

Dr Bill Headley, thanks for help me to develop the prototype.

My Spanish supervisor, Dr Javier del Pino, who has also supported me, and for believe in me and giving me the necessary tools to start and finish my project.

I thank my parents, Carmen & Alfredo, for supporting me throughout all my studies at University, also my sister- Loren R, who were there via facebook to encourage me at every moment :-).

During my six months in Guildford, I would like to thanks all the people I met there for the good time I had: Dr Gokce thanks for your words, Kamran for your unconditional help with my English, Laura and Irene for being patient in the hard days of work (including weekends), supporting me and encouraging me to continue- Irene, also thanks for the last days in the lab, you know what I mean ;-P. Most important with all of you and I safe with me is each moment we spend together.

Finally, a special thanks to all the people that supported me from Spain: my aunts, cousin, grandmother - Florentina Pérez, and all people that love me. As well as, all of them that I have in Venezuela!

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Table of Contents

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

List of Figures

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

List of Tables

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Chapter 1

1. Introduction

After more than a decade of research, Wireless Sensor Networks (WSNs) are slowly entering to the consumer market and experiencing commercial deployment in various application domains. Energy monitoring in home environments, building automation, healthcare, fitness, environmental sensing, etc. are examples to these application domains.

Many of the WSNs are built on radio and networking technology following the Zigbee standard (in particular the IEEE 802.15.4 standard at MAC and Physical layer), which is optimised for energy efficient communication at low data rates. Data from these sensor networks is typically accessed via gateway devices, which implement a network card with the appropriate radio interface.

While such interface cards are readily available for PC based gateway devices, there are no commercial implementations that are suitable for different mobile phone based gateway devices.

The main idea of this project is to provide ubiquitous access to sensor data acquired from the surrounding environment using WSN for mobile phone users. Nearly all of today's mobile phones provide a Bluetooth communication interface for connecting peripheral devices such as headsets and for short-range data exchange with other peer devices and or personal computers. Exploiting this fact, the purpose of the project is to design and develop a Bluetooth to Zigbee Bridge. In terms of software applications will be developed corresponding driver support that allows applications on a mobile phone to query access data from a Zigbee.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Currently there are many possible connections that could be used in mobile phones to provide interconnection with a WSN. Several standards which could become viable alternatives to connect on WSNs in terms of their cost, simplicity and performance ratio are presented below:

1.1 Deciding the right Technology

1.1.2 Wireless Communication Technology

Thanks to advances provided in recent years and incessant study of wireless communication. Today wireless communication technology has become one of the leading communications standard in the field of telecommunications. This would not be possible without WSNs.

The main reason of these facts is that the communication is performed using Radio Frequency (RF-including microwaves and Millimetre waves) since they do not need sight between the transmitter and receiver, and provide links omnidirectional.

In order to achieve efficient communication, different standards have been developed considering the needs of the network in different scenarios in terms of design and quality. These standards use different protocols depending on the application requirements. [1][2]

Thus, their contributions to the WSNs are increasingly important. In mobile phones, wireless networks are supported by standard GSM connection, GPRS, 3G data connections, Bluetooth, Wireless Local Area Network (WLAN), etc.

In fact wireless networks are many times limited by their radio waves. It means that in the case of mobile phone that can easily lose signal reception, it is

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

limited by the scope. Sometimes the environment such as obstacles has limitations too, as well as metallic walls repeater without, concrete that can block the signal [3].

A good reception also depends on antenna features, design, power amplification, etc.

Wireless networks can be classified into different types according to their coverage area. Ports that can be formed by a few or many nodes on a Wi-Fi, is based on the standard 802.11 family, uses the same protocols for implementation of Wireless Local Area Network (WLAN), and communication between them. These nodes in turn are made up of distribution equipment and network terminal equipment. Some points to consider regarding wireless networks on mobile phones are as follows:

- Cellular modems are expensive and delicate when it comes to implementation.

- Battery life does not last long if you are using a Wi-Fi network.

The main characteristics of a Wi-Fi such as high speed, and high bandwidth, have made it use prominent use in the interconnection of computers mainly. [4]

Technology, size and cost are the three factors for inexorable development of sensors networks.

Because of all of the above discussions, wireless network is not used in this project to connect mobile phone and WSNs.

1.1.3 USB

USB is a standard developed by a group of companies including Microsoft, Intel, etc. to establish connection between a device and a host-device. USB has been

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

becoming the best choice to connect devices. The devices don't need their own power supply because USB adds electrical supply. This standard also allows the installation of drivers for hardware, thereby allowing proper synchronization between devices. USB was conceived with the idea of being a standard channel through which two computers can connect which each other, and as mentioned above to allow the connection between external peripherals and the host controller to ensure proper operation. Example external peripherals are: keyboards, audio, printers, cameras, scanners, etc. [5]

Today, virtually all computers have a USB connection, which allows them to communicate primarily with an external device. To carry out the project both the advantages and disadvantages of each device have been analysed. At present, simplicity is very important, but what finally makes it unusable is the cables through we have to connect computers to synchronize with another device. Their length is not extremely small, but is small enough to limit our devices in terms of mobility and autonomy. This is why ultimately the possibility of using a USB device in wireless sensor networks is excluded.

1.1.4 Zigbee

Zigbee is a standard based on IEEE 802.15.4, which is mainly used in communications with low bit rate transmission and low power consumption by offering significantly lower cost. Its protocol stack contains the specifications for wireless communication, specifically for Wireless Personal Area Network (WPAN) based on the physical and Medium Access Control (MAC) defined in the standard mentioned above. Because of its simplicity, it provides secure communications, increased battery life (for low power consumption), easy integration, low cost, among other features that make it a simple and efficient network. Sunspot is created upon IEEE 802.11.4. [6]

Zigbee Alliance is responsible for certifying the Zigbee implementations and

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

developing this application. Currently, the prevailing booms in home automation applications.

1.1.5 Bluetooth

Bluetooth is standardized based on IEEE 802.15.1. Bluetooth provides a low-cost technology primarily for mobile phones. This is characterized by short-range technology and low power consumption.

Its use is closely related to its transmission range, which can be 1, 10 and 100 meters. It provides different transmission rates and at the same time it is characterized by providing Quality of Service (QoS). [1] Its main advantage is to allow communication (data transfer) between mobile phones without the need of a cable. In this way, it provides greater autonomy.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

DECIDING THE RIGHT TECHNOLOGY

Communication Technologies	Standard	**Description**
Wireless Technology (Wi-Fi)	IEEE 802.11 family	Wireless communication is growing on many telecommunications applications offering mobility, high bandwidth, high speed and flexibility. Mainly done for computer link. Require to transmit more power and his implementation is sometimes more expensive than Bluetooth.
Universal Serial Bus (USB)	Standard for peripheral devices	Is a standard that arises with the idea of replacing cables for connecting peripherals to a computer, becoming at standard method to connect of different devices with the same cable. Minimizes the advantages of Bluetooth concerned to mobility and autonomy.
Bluetooth	IEEE 802.15.1	Developed primarily for mobile phones, emerge as the replacement of cables in connection with other computers. Its scope is much shorter and less bandwidth to the Wi-Fi network, but is characterized by low power consumption. Virtually all-mobile devices are equipped with this standard.
Zigbee Standard	IEEE	This specification is developed on high-

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

	802.15.4	level communication protocol, such as the network and application layers. Devices with low rate and short-range is based on this standard. The main application is on sensors networks, home automation, etc. It is focused on interconnection of sensors and devices with battery powered, it allowing using smaller controllers; Zigbee provides the gateway for connecting with sensor networks. Also provides alternative paths and carries messages through intermediate nodes to destination.

Table 1-1. Deciding the right technology

The project foresees the development of a corresponding proof-of-concept prototype and a subsequent demonstration and evaluation of it.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

1.2 Problem addressed and Summary of the main contribution

In order to solve the existing problem, it is proposed to design a gateway adapted to a Bluetooth that can have access to sensor data from WSN. The platform that is used is Sunspot standardized on 802.15.4, which is able to interact with the surrounding environment (collecting data) or with the other nodes on WSNs.

The main purpose is to have access at data sensor on sensor network by mobile phones and creating an environment accessible from mobile devices. Mobile devices are based on Android. Basically they provide a gateway adapted to permit interconnection from end-to-end mobile phone to WSNs.

1.3 Structure of this dissertation

The project is structured as follows:

The current Chapter 1: Introduction. In Chapter 2, the background study and related work about the general concepts of WSNs and the importance of mobile extensions, overall overviews of radio technologies (Zigbee, Bluetooth), as well as a summary of Android OS and Bluetooth stack, and mobile integration are presented. In Chapter 3, System Architecture is presented a description and specification of HW/SW architecture are given. System Component Design is part of Chapter 4. Evaluations of the system components are presented in Chapter 5. And finally there will be a breakdown of the Conclusions in Chapter 6.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Chapter 2

2. Background and Related work

2.1 Wireless Sensor Networks

The need to evolve in communications has made inevitable the emergence of what is now known as Wireless Sensor Networks (hereinafter WSNs). This communication is without physical connection and consists of nodes. It can be composed of just a few to several of them depending on usage. In addition, it does not have a pre-set architecture. This does not mean that a node by itself is able to fulfil tasks. Most of the time at least additional wireless communication is required. Each node is connected to sensors/actuators, and therefore each of them is capable of receiving information from the other sources and retransmitting it at the same time where required. This communication permits physical parameters or sensing. [7][8]

The size of each sensor node is a factor that varies depending on the application functionality. In applications where nodes should be inappreciable, such as military applications, they can reach tiny sizes. For many others, such as weather stations, the size could be compared to that of a shoebox. In short, both the size and price are variants that depend on or are closely linked to the utility, complexity and need for the network.

Another consideration to be taken into account in the design is energy efficiency. In many of the applications of WSNs, nodes can be connected to a power supply directly. In cases where this is impossible, the nodes are equipped with a battery on-board or with solar cells. In both cases, the durability is long in time, so

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

that is one of the main advantages of these networks. In cases where the power supply is not a significant problem, it can be added that nodes have an acceptable size and the cost of each of these is quite low, so the reduction in the total price of a WSN is significant, especially in distributed systems where a higher number of sensor nodes is used. In addition, the software executed on a sensor node may also vary greatly from system to system, depending on the factors mentioned above, and their computing, storage, and communication resources.

The use of sensors or a sensor network extends over several applications, based on its actuating faculties and sensing, combined with its main advantages mentioned above (e.g.: low cost, ease of programming and networking, duration of sensor nodes in terms of energy efficiency). Based on this fact, some of these applications are able to determine pressure, temperature, humidity, vibration, acoustics, chemical sensors, potential event radar, visual and infrared light, and in some more specific applications can have tactile or motion sensors, even determining their own speed or location.

The main components of a basic sensor node are:

MEMORY: as regards memory, a sensor node is equipped with a Random Access Memory (RAM), whose function is to store the packets that were sent from other nodes to it, including the values for the sensor, or a Read-only Memory (ROM) or Electrically Erasable Programmable Read-Only Memory (EEPROM) whose function is to store the program code. This is because RAM memory is volatile, which means that information is lost at the moment that there is a power cut, which obviously requires the use of ROM or more typically EEPROM.

CONTROLLER: among its main functions it can obtain information from the sensors, receive data sent from other nodes and process these data, and determine the behaviour of the actuator. Ultimately, it is the Central Processing Unit (CPU) of each node.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Sensor/Actuators: this is where you interact with the external world: setting physical parameters, monitoring or obtain information by detecting of their environment.

Communication Device: wireless communication allows the exchange of data between different nodes. Each individual node is capable of bi-directional communication (sending and receiving data at the same time).

Power supply: is the energy source of a node. As mentioned above, the supplying energy can be in different ways, e.g. by solar cell.

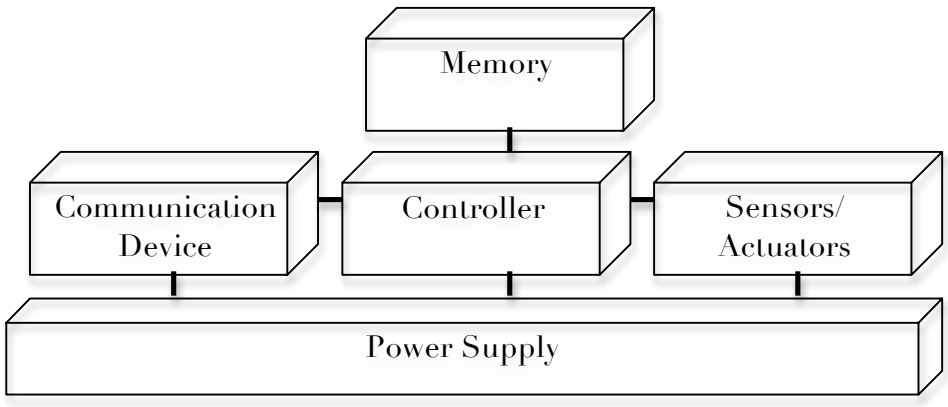


Figure 2-1. Main components of the basic sensor node [8]

Wireless sensor networks are based on radio technology, based on different standards such as Zigbee, more specifically defined by the IEEE 802.15.4 standard in media access control-MAC and physical layer.

2.1.2 Mobile ad hoc Networks and Wireless sensor Network

So far we have reviewed features of wireless sensor networks; self-configuration is one of them that we cannot omit. A self-configuring device is made to comply with a sole purpose, thus achieving a smooth and rapid communication.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The wireless sensor network allows through nodes to extend the geographical coverage area, which means that nodes can communicate with each other without direct vision; also they can transmit information from a remote location to another. If we compare it with other networks, where cables are used between points to establish a communication, or simply have to have direct communication between receiver and transmitter, sometimes we find structures that prevent direct communication and so the operation of the nodes is more efficient. [8]

MANETs (Mobile Ad hoc Networks) is a concept of the family of WSNs. Due to the applications of WSNs, their innovative new mechanisms for network communications, as well as their architecture, this new concept is introduced as an extension and improvement of WSNs.

In WSNs, data from physical interaction with the environment or from different applications specific to the WSN, what we call the data source, are given to an own network (nodes) or many times systems outside the network (the fire-fighter's PDA communication with WSN). It is here where the importance lies of mobile extensions in their communications with WSNs.

In MANETs human interaction plays an important role in bringing to communicate, due to complex and diverse applications developed by each team; unlike the WSN, this requires the use of powerful computers such as PDAs, Laptops ... that in turn, are able to establish communications with web servers or applications that require greater support for its implementation. An example of the interaction of humans with these applications is the use of these communications directed mainly to voice communication, where obviously without them any human interaction would not be possible.

In addition WSNs allow for extending the range of applications, because they have great diversity of settings of actuation; examples of this are applications that allow real time monitoring, such as alarms. In which MANETs are directed instead

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

to more conventional applications (as mentioned above: voice, web server, so on.).

In MANETs applications on the other hand, are specific applications in which there is no interaction with the environment. We can expect as far as traffic characteristics are concerned, greater stability, meaning that there is no abruptness or changes in a connexion network over time, so you can scale or somewhat predict network traffic. In WSNs, having applications that interact with the environment, characteristics of network traffic if you produce abruptness or changes over time, means that you can spend a long period of time without activity as we can change from one moment to another high traffic in a matter of minutes or seconds, because it is monitored in real time (example are alarms or sensors to monitor an event).

As for energy consumption, WSNs have more restrictions that allow extending the life of the batteries, so its replacement (recharge or replace battery) is less frequent than for the case of MANETs. This is due to the requirements in the system architecture; it is simpler for WSNs (thus decreasing consumption) than for MANETs.

Both of them, MANETs and WSNs, have self-configurability (as mentioned earlier). If we refer to requirements, both present and require different levels of Reliability and QoS. In the case of MANETs these are regulated by the application used: for applications such as voice, your requirement is low. In the case of WSN, the QoS is a concept that must go hand in hand with the energy consumption, and reliability is measured by each individual node.

MANETs use data redundancy, have greater memory capacity and their architecture in terms of network software and operating system, is more complex than in the case of WSNs. Nodes in WSNs are characterized by their simplicity and lack of resources, are further limited in energy, so the system architecture must be consistent with the constraints presented in comparison with computers (desktops) today.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

2.2 Zigbee

Zigbee is developed by “Zigbee Alliance” and standardized in IEEE 802.15.4-based wireless technology and is mainly characterized by its low cost, low-data rate, and one of its main requirement: long battery life. Zigbee standard is recognized as meeting the highest levels of network and application. Its main applications are extended in the field of home automation, remote controls, sensors, and so on. [6]

IEEE 802.15.4 defines the physical layer and compatibility with other wireless networking standards such as Bluetooth (IEEE 802.15.1) and WLAN (IEEE 802.11) [9]. The physical layer in a transmission model is in its lowest level, the role of managing the reception / transmission of packets that are in a network, connecting / disconnecting devices on the network, while also providing a medium based on basic safety skills. On the other hand, in its highest function it provides the necessary data to the upper layer [8]. It is known as a global standard because of its adaptability to different network topologies, and the fact that it works in the 2.4GHz band used worldwide, and also allows better communication development. [10]

High reliability: Because it provides "robust radio technologies" and offers a low signal to noise ratio. Depending on the network topology such as mesh networking, it provides access to different nodes, providing different routes / options for communication between them, giving independence and after being able to detect alternative routes in case there is a disability (barrier or blocking) between origin-destination points. It also recognizes if the package has been received more than once, thus avoiding duplication.

Profitability: Often technologies are associated with a high cost both in terms of post-production and pre-production, but Zigbee is characterized by its low cost. And the reason for this cost-effectiveness is that a module is basically made up of small micro-controllers, sensors and actuators that allow for a small size battery and so more easy to add on boards.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Being a competitive product on the market in terms of profitability and reliability.

SECURITY: It is based on the AES-128 bit security. This standard has been studied and designed to ensure security at the level of data packet travelling in a Zigbee network consisting of nodes. Encrypting and decrypting network packets make it inaccessible to intruders, so it is difficult to corrupt data packets in these networks. As stated in the book: "Zigbee uses AES-128bit encryption and authentication for security."

LOW-DATA RATE: Zigbee has an average of 25kilobits per second, making it especially suitable for low data transmission environments. Its wireless communication is not simultaneous, even being communication at half duplex (both directions) between devices, which means that while transmitting it is not capable of receiving and vice-versa.

2.2.1 THE ZIGBEE ALLIANCE

Zigbee Alliance was created under the banner of global standard, adding to this the characteristics described above. It was created more than a decade ago by a group of companies focused on bringing to market a standard that would fit the needs of wireless technologies and adjust their requirements, thus forming what is known as Zigbee.

Figure 2-2. Zigbee Alliance. [6]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Figure 2-3. Zigbee Alliance Members [10]

Today there are more companies that join as a promoter, participant and adopter to develop this standard, obtaining a product that meets market needs in many applications such as: building / home automation, body sensor network, healthcare, remote control, and so on.

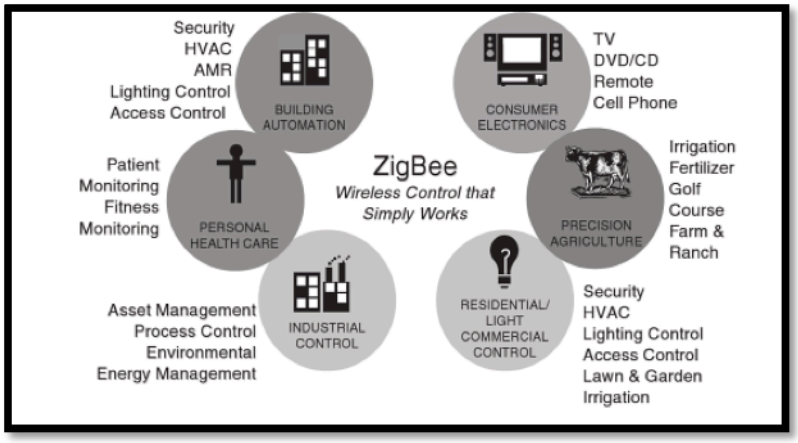


Figure 2-4. Zigbee Markets. [10]

In order to facilitate adaptability to market standard, Zigbee Alliance introduces two specifications, which are listed below:

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The Zigbee Specification

Zigbee establishes two classifications (Zigbee and Zigbee PRO) providing suitable characteristics for use in a meshed network Zigbee, and providing autonomy, versatility and ease of use. These are self-configuring devices and offer low power consumption, ideal for networks with many nodes.

The Zigbee RF4CE specification

It is characterized by its simplicity, which means that they are low cost devices. It has been developed in network applications therefore simple topologies are not very expensive to implement it. These devices offer features that do not require the ability of a node in a mesh network Zigbee. Its function is designed for two-way communication, device to device; this makes it easy to commercialize and specialize in developing simple network topologies and to facilitate their expansion.

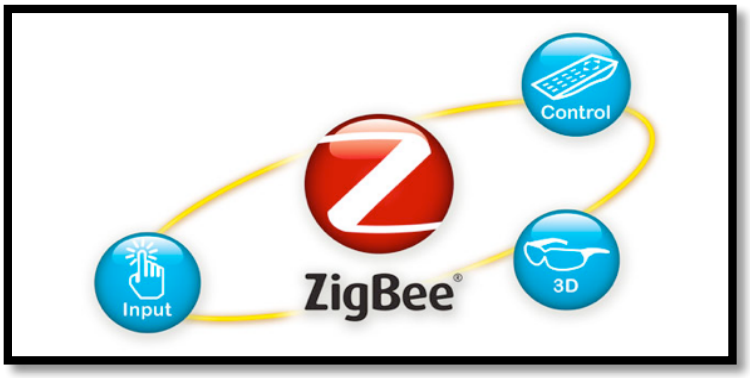


Figure 2-5. The Zigbee RF4CE Specification [6]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

2.2.2 Zigbee Architecture

Zigbee architecture does not follow the structure of the OSI Layers Model-7. Overall Zigbee is organized by Physical, MAC and Network layers. The upper layers (application, presentation, session, transport), compared to the OSI model are included in Application Support (APS) and Zigbee Device Object (ZDO) layers.

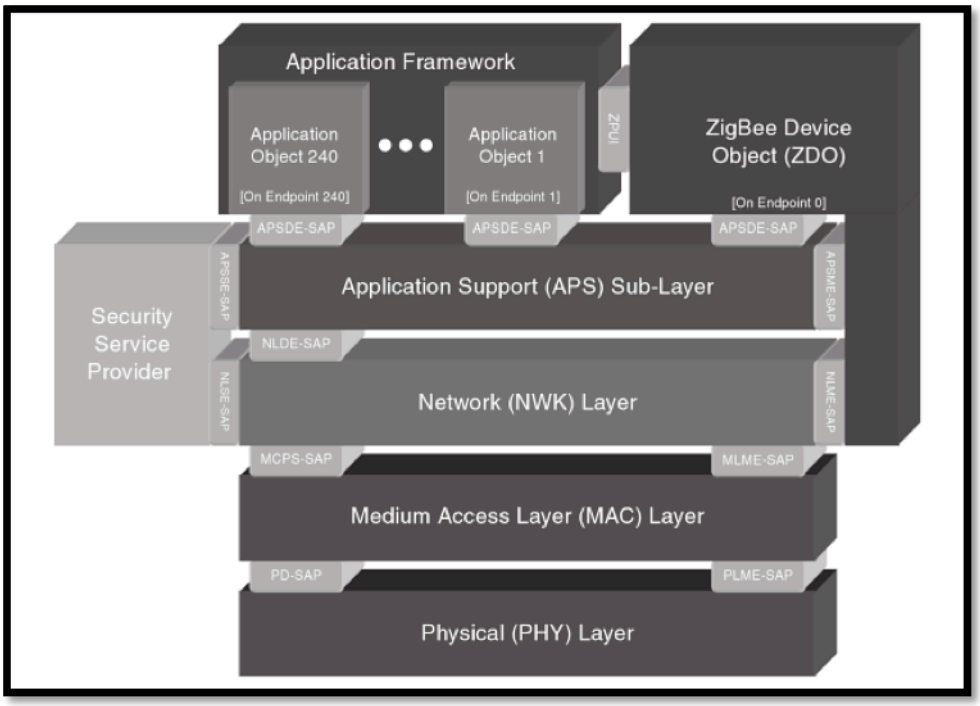


Figure 2-6. The Zigbee Architecture [10]

Between layers are Service Access Points (SAPs), which are responsible for identifying the internal operations between the layers before and after providing an Application Programming Interface (API). These fall into internal operations: management and data between layers, so finding a SAP for data and another SAP for management. For example we have the Network Layer Data Entity Service Access Point (NLDE-SAP), whereby all data communications go to and from the network layer.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The lower layers such as physical and MAC layers are denoted on the IEEE 802.15.4.

The Physical layer is in charge of the conversion to bits of packets to be sent or received over-the-air.

The MAC layer provides data network, including PAN ID (unique identifier), manages connections and networking discovery.

The Network layer, is in charge of providing the route where packets are sent / received, and is also responsible for making the broadcast in a mesh network. In addition, including security in the transmission / reception, determines alternative routes in case of finding any impediment between two points.

The APS layer, works as a filter to avoid duplication of messages. It is in charge of running the application, simplifying the logic of it. It is here where the nodes or groups are available to connect to the network, summarized in a ''local table binding'' adding this layer.

The ZDO layer manages the connexions over-the-air, which means that it provides the node status, available services and service to discover nodes in the network. This layer includes Zigbee Device Profile (ZDP).

The Application framework provides the framework where applications are executed and where the Zigbee Cluster Library is located.

2.3 Bluetooth

The idea of creating a secure communication between two devices without using cables and over short distances is what we know today as Bluetooth. Bluetooth provides a secure communication by pairing devices. This means that one device

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

has to begin a connection by sending an authorization, the specified device has to accept it, and in this way the connection is established. To transfer files between two devices connected via Bluetooth it is necessary, as discussed above, to establish the connection through the pairing of the same. The device that sends the request for parity to connect is called hereinafter the *server*, while the device which accepts the connection, called hereinafter the *client,* remains pending to receive the packets transferred from the server. [11]

Communication is established through wavelength radio transmissions.

To facilitate the search for Bluetooth devices at the user level, generally we associate a name (device name) to a numeric address. This name will allow users to easily identify our device, or if we want to connect with any other device, to know the device we want to connect to. Internally, when the device is trying to search and establish the connection a conversion is made in which the "name" or identifier, is displayed to the user as the Bluetooth address associated with it. The way to find other devices is making 'broadcast a discovery' which permits us to know the devices around us, awaiting responses from these devices with Bluetooth active, ready to connect.

There are several transport protocols that are used by Bluetooth devices. In our case we will refer to the Radio Frequency Communication (RFCOMM) protocol, the most used by these devices, with virtually the same quality of service as TCP and highly robust. The way a transport protocol specifies communication programs is through the allocation of ports.

PORT NUMBER: in general, the ports are numbers allocated to enable communication between devices, in some cases reserved for a particular purpose, allowing multiple applications simultaneously. Bluetooth transport protocol RFCOMM contains channels numbered 1-30 available for allocation.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

In searching, we have different configurations. One of them is 'traditional connection establishment', which allows communication between Client and Server, with both configured in the same port number. On the other hand, we have 'connection establishment using SDP'. SDP stands for Service Discovery Protocol, as we specify in the manual for programmers: "*uses port 1, and RFCOMM connections are multiplexed on Logical Link Control and Adaption Protocol (L2CAP) port 3*". This protocol allows configuration in various scales of performance or quality and is based on packet exchange; therefore SDP provides the list (service record), which gives the description of a service; among other specs on this list, the most relevant are the Service ID and the Service Class ID List, through which the service is identified.

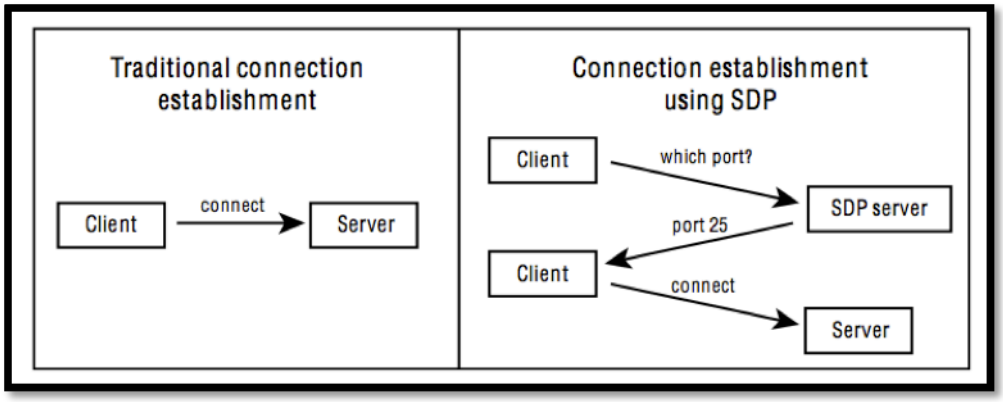


Figure 2-7. Traditional network applications use hard-coded port numbers, whereas most Bluetooth applications use an SDP Server. [11]

As stated above, some protocols, such as TCP and L2CAP, have specific ports for particular applications or special services (ports reserved). Bluetooth instead of assigning a port, offers the possibility of assigning an identifier through which the device knows the type of service offered or which kind of service to connect to.

Universally Unique Identifiers (UUID) is used in the Service ID and its function does not go beyond establishing a specific service class. It also permits us to know pre-created profiles and to describe the transport protocol. To conclude, the manual for programmers clearly relates the UUID with the SDP as

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

follows: "*Specifically, a developer chooses this UUID at design time and when the program is run, it registers a service record containing its Service ID with the SDP server for that device. A client application trying to find a specific service would query the SDP server on each device it finds to see if the device offers any services with that same UUID*".

SDP	0x0001
RFCOMM	0x0003
L2CAP	0x0100
SDP Server Service Class	0x1000
Serial Port Service Class	0x1101
Headset Service Class	0x1108

Table 2-1. Example of reserved Bluetooth UUIDs to obtain the full 128-bit UUID from a reserved UUID, shift it left by 96 bits and add the Bluetooth Base UUID. [11]

2.4 Android OS

Android as a Smartphone platform emerged from the idea of a group of companies that together formed the Open Handset Alliance (Google Inc., HTC, T-Mobile, etc....). These leaders in the telecommunications industry came together with a single purpose: to create an open platform for software development targeted at mobile devices. [12]

The software stack includes the operating system (Android OS), software development (SDK) and development tools, based on the Linux kernel and using mainly Java programming language.

Among its features it provides the following: an application framework which enables the reuse and replacement of components, simplifying their use, as well as services and systems attached as resource manager; management in notifications and

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

activities; an integrated browser based on WebKit; API of 3D graphics based on OpenGL ES 2.0 and graphics 2D; database SQLite for structured data storage; Dalvik(VM) optimized for mobile devices where the application is running; technology, hardware-dependent, such as GSM (mobile), Bluetooth, EDGE, 3G and Wi-Fi; support to media audio / video and image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), in addition to the features offered by hardware such as Camera, GPS, compass and accelerometer; it also includes everything you need in a development environment such as emulator, debugging tools, performance profilers, libraries, IDE plugin for Eclipse.

Android also provides a flexible platform that allows application development to turn into something practical (cheaper) and accessible in developing applications, thereby extending the functionality of mobile devices. Released under Apache license (free software and open source license). It promises to be a fount for those programmers who want to innovate in the development of communications and mobile applications. It also allows for downloading of all the online applications through the Android Market. [12] [13]

2.4.1 Bluetooth Stack

In the basic structure of the frameworks, the Android platform is compatible with the communications stack: Bluetooth. Bluetooth is used, as we mentioned before, for wireless communication between devices at short distances, so the platform allows the exchange of data with other Bluetooth devices. This is because it incorporates the functionality of Bluetooth in APIs of Android Bluetooth. These APIs provide wireless access features: point-to-point or multipoint, at the same time of being responsible for the connection allowing data exchange between devices. To make the connection between Bluetooth devices a series of steps are needed which encompass and characterize a Bluetooth device and allow the exchange of data. So definitely an Android Bluetooth API is able to run those steps, adding the following features: searching for Bluetooth devices, consulting the local Bluetooth adapter for

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

devices paired, setting the RFCOMM communication channel, connecting to other devices via the service discovery (using UUID), managing multiple connections and ultimately making the transfer of data to and from other devices. [12]

To carry out this process the Android Bluetooth API is built into the package android.bluetooth adding the following structure of interfaces and classes that are needed:

Interfaces	
BluetoothProfile	Is an interface that is in a Bluetooth profile specification. This specification describes the interface that enables wireless communication between Bluetooth-based devices. (Eg hands-free profile.)
BluetoothProfile.ServiceListener.	This interface allows the internal communication to perform BluetoothProfile between clients IPC that have been connected or disconnected from the service.

Table 2-2. Interfaces of Android Bluetooth API [12]

Classes	
BluetoothA2dp	"Advanced Audio Distribution Profile", sets the mode in which a device transmits to another, high-quality audio through a Bluetooth connection.
BluetoothAdapter	Local Bluetooth Adapter (Bluetooth radio). In any Bluetooth communication, it searches for devices, sees devices that have been paired, and at the same time creates from a known MAC address a BluetoothDevice and this is where you create the BluetoothServerSocket to

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

	listen to other devices.
BluetoothClass	Refers, in a Bluetooth device, to its characteristics and properties that define it as well as their capabilities and services. They do not define, most of the time, reliably Bluetooth profiles or compatibility with different services, so it is advisable to have a simple reference in use. Other Classes: BluetoothClass.Device; BluetoothClass.Device.Major; BluetoothClass.Service.
BluetoothDevice	Used to make a connection to a remote device through a BluetoothSocket, getting from it its name, MAC address, device class, and service class, among other features that may include the remote device and representing BluetoothDevice.
BluetoothHeadset	Used for Bluetooth connections between mobile phones and headsets, providing the necessary support and profiles used (both headset and handsfree).
BluetoothServerSocket	Represents a socket on the server that listens for incoming requests, analogous to the behavior of TCP ServerSocket. Upon acceptance of a petition to establish the connection, BluetoothServerSocket provide the BluetoothSocket currently connected. To make possible the connection between Android devices, one must open a ServerSocket with this class.
BluetoothSocket	It is here that the exchange of data between Bluetooth devices via the InputStream / OutputStream is allowed. Its behaviour is analogous to a TCP socket.

Table 2-3. Classes of Android Bluetooth API [12]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

2.5 Related work for mobile integration

Advances in technology and communication have allowed direct investigations into the field of wireless sensor networks (WSNs), now emerging in applications not only military but also commercial.

As mentioned in this chapter, WSNs are developed in different fields of application and are increasingly the advances in these technologies, in turn allowing the incorporation of other technologies with methods of short distance such as Bluetooth devices.

The first projects developed with sensors, catalogue their beginnings over a decade ago.

Initially the sensors were conditioned to obtain data such as vibration, pressure, sound, light, and temperature. Although that its first applications were intended for military applications (such as intrusion detection, chemicals...) gradually rolled out in conventional and commercial applications in different fields, such as: efficient use of electricity, environments that require a high level of security (e.g.: airports, government buildings...), environmental sensors (temperature, humidity...) and industrial (vehicle tracking, fleet management...) in the automotive industry in the field medicine (heart rate, blood pressure...), automation (home/intelligent buildings...), among others.

Since, as we have seen, is a technology that is evolving exponentially, and at high speeds, it was inevitable incorporation of other technologies that working in tandem with the WSNs (such as mobile integration, before this: incorporating a personal computers PC on WSNs), so as to achieve the data exchange, or disseminate effectively the information obtained by these sensors, to an end user, and thus work with the data obtained, achieving a breakthrough in adding interaction between the environment and human being.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

As for wireless sensor networks, the mobile integration has been undoubtedly one of the major advances providing end users with ubiquitous access to data from the sensor in a WSN.

The incorporation of Bluetooth technology facilitates the adaptability of integration mobile in WSN. Exploiting the fact that most mobile phones have Bluetooth, and wireless sensor networks is a fact that every day thrive in different application areas, arises the need to initiate remote access to these networks, incorporating as a solution remote access, mobile phones.

As emerging technology, arise different projects, which correspond to the incorporation of remote access via the mobile, incorporating some improvements to existing technologies, and others to be a focus on technological innovation.

Today, has been developed into different aspects, one of them the mobile integration on wireless sensor networks, broadly we named some of work that include mobile integration and enabling the proliferation of the same on WSNs:

“Mobile Enabled Large Scale Wireless Sensor Networks” [14] incorporates mobile wireless sensor networks, offering a growing improvement in large-scale networks. Introduces a new network architecture for wireless sensor networks "multi-radio enabled mobile wireless sensor and network" (MEMOSEN), adding mobile devices to wireless sensor networks, thereby obtaining a type of hybrid wireless sensor network in which mobile devices act as a sensor.

“A Heterogeneous Sensor Network Architecture for Highly Mobile Users” [15] combines "Personal Area Networks" (PAN) with the WSN technology with, thus carrying sensor data in a network body "Body Area Network" (BAN) to the Internet. Presents results for low-scale, meaning less number of sensor nodes, focused on addressing the following of people with great mobility in hazardous work environments, such as firefighters and police force.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The project is aimed at providing coverage in rural areas, so the mobile phone can include greater coverage area, and where the information recorded by a sensor BAN can be retrieved remotely and using the Bluetooth module to communicate through the mobile terminal to the "backbone"-Internet. Equipping the sensor nodes with a dual-radio technology as the gateway: Bluetooth module for connecting to the Internet via mobile phone and radio broadcast to route data from node efficiently. In short, they propose a network that is structured on one hand: the BAN sensor communication with the mobile device (with the latter offering greater coverage) and through the Bluetooth module to connect with Internet (Bluetooth both in the BAN sensor as well as on mobile devices), on the other hand a headquarters or command centre located anywhere worldwide, with the point of interconnection between them: Internet.

Bluetooth arises as compatible medium between the wireless sensor networks and mobile terminals, as a standard mobile terminal lacks the radio technologies used in WSNs, impeding to act as a gateway for direct access. This is where we try to explore adding a Gateway that incorporates Bluetooth, capable of communicating at one end with the WSN (without adding Bluetooth module for each sensor) and the other with mobile terminals (taking advantage that mobile terminals are equipped with Bluetooth).

“Open Wireless Sensor Network Telemetry Platform for Mobile Phones” [16] developed to create an open-source system, which creates a link by which the data are collected by the user via a mobile terminal, through a gateway, and sent for further processing. These data are overwritten on every request made by the user of the WSN. The technology used between the WSN and the mobile is Bluetooth, this time, has created a Bluetooth Gateway for connecting planned sensor networks with mobile terminals at low speed. The mobile terminals are a resource to enable connection of WSN (low energy) to the Internet.

Since the WSN dissipate as much battery power in the transmission of data,

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

incorporating the functionality of the mobile phone, this research is a subset of terminals (phones) that allow download, transfer and save WSN data. Being on the other hand, function of the Gateway obtain data from the WSN, thus achieving a one-way communication and widespread.

Various sensors spaced in a particular area, collect data that are stored by the Gateway, and this in turn, when requested by the user, transfer this data to mobile terminal.

In our case, the interaction can connect to a given node of the network of WSN and get the data the user wants through the Gateway. Also provides Bluetooth connectivity in faster mode, obtaining data from the WSN without using the Internet, and changing the approach to create a subset for downloading data from a WSN, for other that allows users to access extended with mobile, any node in the WSN.

The incompatibility of protocols between mobile technologies and WSNs has created a barrier in providing direct access to wireless sensor networks.

We have described in this section, several ways in which mobile technology has been investigated and integrated in the WSNs, remote access to wireless networks is not a new topic and increasingly becomes more significant because of the importance and latent evolution in different applications. To our case, we propose the following approach: to adapt a Bluetooth module on wireless sensor platform, as the platform Sunspot (by Sun Microsystems), allowing connectivity to mobile devices via the interface Bluetooth on both ends (WSN-mobile terminal) and ensure the exchange with other nodes in a WSN by communicating via radio technology pertinent: Zigbee.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Chapter 3

3. System Architecture

3.1 Requirements

The installation of a network of sensors and functionality are separate activities, need not be related to the distribution of nodes with the functionality they will inherit.

Depending on the functionality we want to give the sensor network can adapt to different topologies, for our case, the distribution of the nodes will follow mesh network topology.

As mentioned, a mesh network of nodes, involves the distribution of nodes in an area of interest, in which each node communicates with each other, directing information to a device, which we call 'Gateway'. Subsequently, the latter will send relevant information to end-users.

The installation of many nodes on a network, ensure greater coverage area, which may extend it, and on the other hand, these added nodes are able to supplant the failed nodes around it.

Requirements for both WSN and for each sensor node:

Characteristics of WSN:

- ✓ Lifetime
- ✓ Network Coverage
- ✓ Security

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

- ✓ Cost
- ✓ Easy Installation
- ✓ Response Time
- ✓ Accuracy and measuring frequency

Characteristics of the Sensor Node:

- ✓ Energy expenditure
- ✓ Robustness
- ✓ Communication skills
- ✓ Security
- ✓ Flexibility
- ✓ Size and cost features of each node
- ✓ Easy Sync

In the WSN, the topology in the network structure mostly used, is a mesh network topology. The mesh topology networks are those in which the nodes are evenly distributed, allowing the transmission between the closest nodes. This distribution extends the coverage area, as mentioned above. Another feature in mesh networks is that although the nodes have the same characteristics and qualities, can be assigned to one additional functions, being in this case the network led by it and in turn be easily supplanted by any other node.

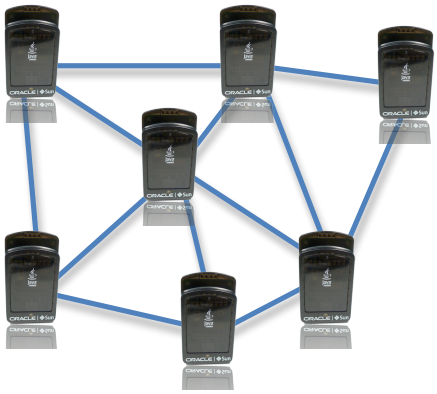


Figure 3-1. Mesh Topology in WSN.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

This type of network has a greater robustness, at the time of failure on individual nodes or links, providing alternative routes between origin and destination node, usually indicate the same characteristics, which facilitates communication between them.

The main function of the MAC layer is give media access to try to transmit data, control or manage the packets from other nodes (unicast), also has the function to configure the nodes in broadcast, multicast mode, in this way, applications that depend on this behaviour, comply.

Another major reason to use the MAC protocols, is the importance of reducing energy consumption in the network, the approach to achieve this is by entering a sleep time to the devices while they are running, thus achieved reduce energy consumption, an important requirement in WSN.

The Physical layer, located below the MAC layer, its main function is to transmit and receive data packets from the physical channel. Makes the changes neccesary in the sequence of data to and from the physical channel.

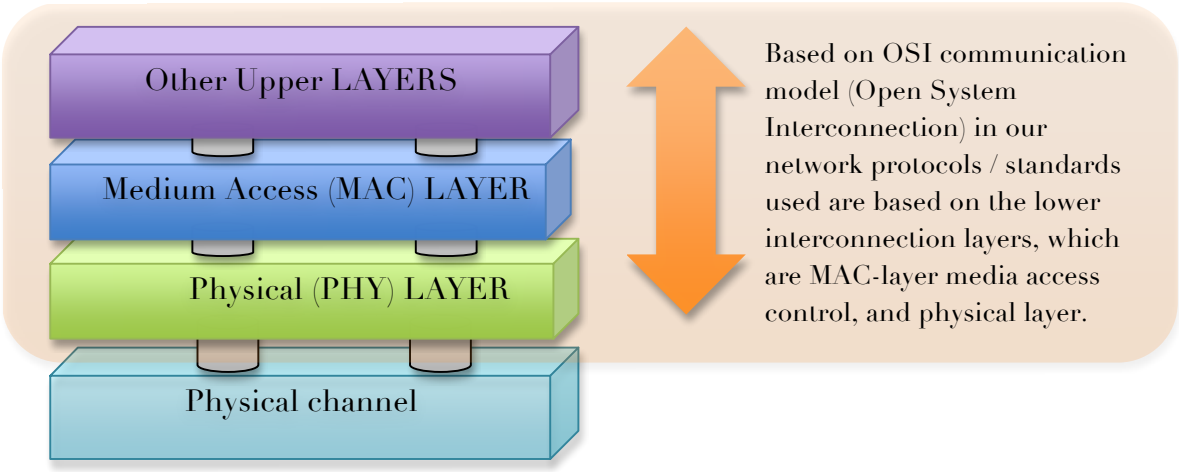


Figure 3-2. The Lower Layers in OSI communication model

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

3.2 Hardware Architecture

The following shows the main characteristics of the devices, which will be made by the network, taking into account a number of requirements to achieve efficient use of it.

Sensor nodes are based on radio technology and Zigbee network following the IEEE 802.15.4 standard, which is optimized for efficient communication of energy at low speeds.

To access the data from the sensor network has designed a Gateway device, which implements a network card with Bluetooth radio interface, to provide access to end users, the network of sensors.

Sensor Node: Sun SPOT. The SunSPOT consists of two plates: Processor (eSPOT) and Sensor board (eDemo), each developing different tasks. [17]

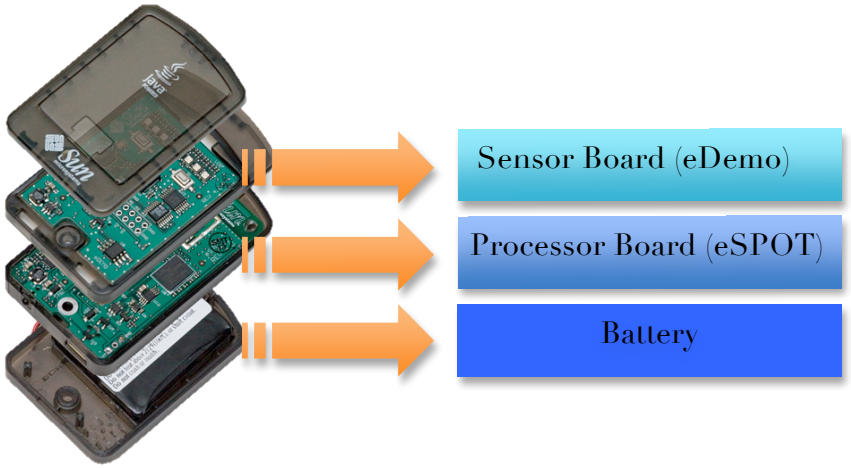


Figure 3-3. SunSPOT components. [17]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

SENSOR BOARD:

Characteristics of the Sensor Board [17]:

- Use the temperature sensor on main processor board.
- 4 digital GPIO pins: D0-D3 (no D4).
- 4 analogic in lines: A0-A3 (no A4+A5). Sampling takes longer as done via ATmega microcontroller.
- Tri-color light sensor and Tricolor LEDs controlled directly by main processor board.
- MMA7455L Accelerometer replaces LIS3L02AQ accelerometer.
- I2C from sensor board output available on AVR SCL & SDA pins.
- Lo-fi audio speaker lets the SPOT now make sounds.
- IR receiver (on SW1 input line) & IR transmitter (on SW1 input line).
- By default sensor board ATmega microcontroller stays awake when main processor board is powered down for deep sleep, and can generate interrupts to wake SPOT up on pin changes, switch presses, etc.
- Serial line from main processor board is available on RX/TX pins.
- I2C from main processor board is available on SCL & SDA pins.

PROCESSOR BOARD:

The following are the characteristics that the Processor board, which will be adapted in which the Bluetooth module. [18]

- Incorporates the ARM920TTM ARM® Thumb® Processor. The AT91RM9200 is a complete system-on-chip built around the ARM920T ARM Thumb processor. It incorporates a rich set of system and application peripherals and standard interfaces in order to provide a single-chip solution for a wide range of compute-intensive applications that require maximum functionality at minimum power consumption at lowest cost.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

- Low Power.
- Additional Embedded Memories (16K Bytes of SRAM and 128K Bytes of ROM)
    - Internal Memory Mapping
        - Internal RAM
            - The AT91RM9200 integrates a high-speed, 16-Kbyte internal SRAM.
        - Internal ROM
            - The AT91RM9200 integrates a 128-Kbyte Internal ROM.
- External Bus Interface (EBI)
- Ethernet MAC 10/100 Base-T
- USB 2.0 Full Speed
    - The AT91RM9200 integrates a USB Host Port Open Host Controller Interface (OHCI).
- Multimedia Card Interface (MCI)
- Synchronous Serial Controllers (SSC)
- Universal Synchronous / Asynchronous Receiver / Transmitters (USART)
    - Programmable Baud Rate Generator
- Master/Slave Serial Peripheral Interface (SPI). Access to the radio (CC2420) on a second SPI bus.
- Timer/Counters (TC)
- Debug Communication Channel (DCC) support
- Power Supplies.
- 3.7V rechargeable 770 mAh lithium-ion battery
- 65 uA deep sleep mode
- 2.4 GHz IEEE 802.15.4 radio with integrated antenna

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

BLUETOOTH MODULE:

Bluetooth is a wireless communication link, operating in the unlicensed ISM band at 2.4 GHz using a frequency hopping transceiver. It allows real-time data communications between Bluetooth Hosts. The link protocol is based on time slots. [19]

Baud rate speeds	2400-115200bps
Class 1 Bluetooth Radio Modem	Distance: 330' (100m) Output transmitter: 12dBm -80dBm typical receive sensitivity
Frequency	2402 ~ 2480MHz
Modulation	FHSS/GFSK 79 channels at 1MHz intervals
Encryption	128 bit (secure communications)
Configuration	UART local Over-The-Air RF configuration (like Wi-Fi, 802.11g, and Zigbee)
Others	Error correction for guaranteed packet delivery. Auto-discovery/pairing requires no software configuration (instant cable replacement). Auto-connect master, IO pin (DTR) and character based trigger mode. Low power consumption: 25mA avg. Operating Voltage: 3.3V-6V

Table 3-1. Bluetooth: Technical Characteristics [20]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

3.3 Software Architecture

Bluetooth protocol Stack

The Bluetooth protocol stack allow implement communication between devices, this protocol stack, as shown in the figure below, consists generally of higher level protocols and protocols at low levels.

Under the application layer, we find the JSR-082 APIs, those APIs contains higher-level protocols as low-levels protocols. These APIs have been made to the software Bluetooth developers, who working in a computer environment based on the Java programming language. [21]

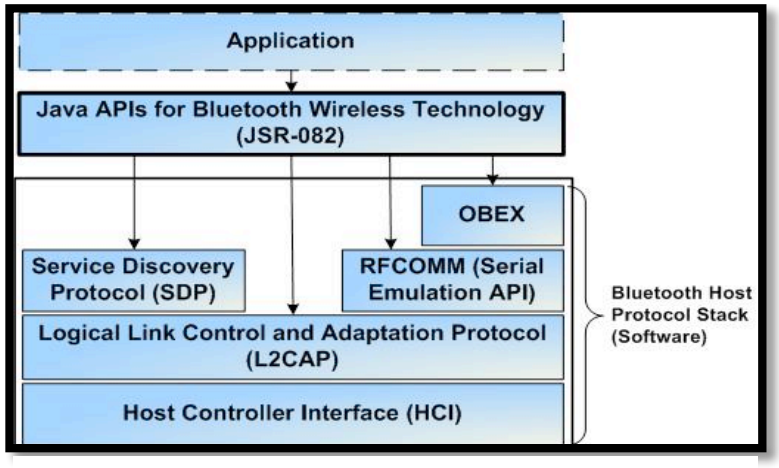


Figure 3-4. Bluetooth Protocol Stack. [21]

The higher-level protocols, refer to the APIs for Bluetooth wireless technology, include:

SERIAL EMULATION INPUT / OUTPUT, this emulation is done through series of *RFCOMM*-Radio Frequency Communications to create a Serial Data Stream.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

OBEX, object exchange, it occurs on RFCOMM protocol.

SERVICE DISCOVERY PROTOCOL (SDP) allows knowing the services supported by the remote device, in addition to the parameters associated with it.

The low-level protocols provide packet segmentation, reassembly of packets, and quality of service and multiplexing of logical connections that use different protocols in the higher level. Provided by the *L2CAP* layer, Logical Link Control & Adaptation Protocol.

HCI- HOST CONTROLLER INTERFACE, in this layer there are different standards, using different hardware interface, this allows to exchange data with minimal adaptation to facilitate access between the controller firmware (host stack) and driver software (Bluetooth communications module IC).

SUN SPOT

Sun SPOT (Sun Small Programmable Object Technology) is a Sun Microsystems' Project in order to give programmers confidence and support to start new applications based in wireless communication (Wireless sensor network built as mentioned earlier in IEEE 802.15.4 standard) and thus get another vision to create programs that interact with each other and with the environment as well in the same time their users can done and apply any new methods of programmable.

JAVA.

The program to development and work with Sunspot is Netbeans and language is based in Java.

Java developed by Sun Microsystems it's the programming language easier based in his slogan: “Write once, run everywhere”. In fact, each application can rely

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

primarily on existing classes that are part of language itself as components, database access, threads, so on; or created by users, which frankly makes it more feasible to handle by anyone, result of having been designed more recently and created by one unique team has made more easy to work on it. They define it as "with its versatility, efficiency, portability, dynamic, multitask and security make it the ideal technology for network computing". [22]

Program Supported on Sunspot: Java.

SUN SPOT SOFTWARE STACK:

The Software Stack of each node based on Sun SPOT platform, contains the next layers.

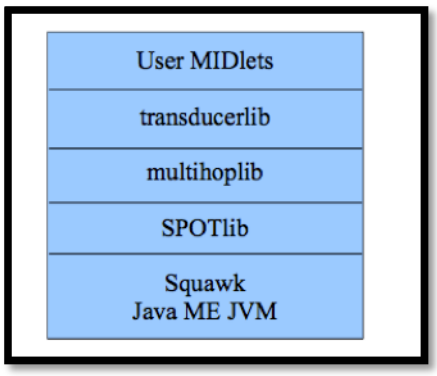


Figure 3-5. SPOT software stack. [23]

In the top, User MIDlets, this layer is where the user write the SPOT application (extends the Java ME MIDlet class), in the opposite, on lower layers, Squawk is the where the application runs, is not a operating system, and is the Java Virtual Machine. [23]

Between them, different SPOT libraries, like as SPOTlib provides us the SPOT access, basics I/O, including the MAC access radio (low-layer protocol).

The multihoplib provides the high-level of radio protocols implicated as

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Radiogram and Radiostream, liaison between devices to provide routing of packets between distant points.

To access the SPOT hardware board, as is the case, access to the eDemo sensor board and its main functions, such as accelerometer, LEDs, switches, digital / analogy, etc., we have the transducerlib library, who provides support and access.

SunSPOT software. [17]

- IPv6 support to radio protocols.
- API to give support writing device drivers.
- Sensor API to provide generic interfaces to common sensors and actuators for SPOTs.
- Basic Task class to specify some code to be run starting at a specified time and then repeated periodically.
- BootloaderListenerService.
- IDeepSleepListener for registering/notifying listeners when SPOT wakes from deep sleep.
- IStandby Interface: used by accelerometer & tricolor light sensor.
- ADJDS311TriColorLightSensor class.
- TriColorLEDArray class.
- WatchdogTimer class.
- Four serial lines are now available:
    - One from the sensor board: edemoserial://
    - Three from the main processor board:
        - Serial://usb & serial:// which both go to RX1/TX1
        - Serial://usart & serial://usart0 which both go to RX0/TX0
        - Serial://usart1 which goes to RX2/TX2
- FiqInterruptDaemon to support multiple listeners for different events & added a method to disable button presses from causing a reboot.
- IToneGenerator API.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

- eDemo board speaker controlled by a Tone Generator.
- All MIDlets now run in child Isolates. The SPOT can be configured to run several different MIDlets at startup.
- 'spot.restart.mode' property to control what happens when all MIDlets finish. Recognized values are: restart (reboot SPOT = current default), off (SPOT goes into deep sleep) or continue (SPOT goes into shallow sleep & continues to listen for OTA radio commands).
- Gateway Server to run multiple host services to/from SPOTs.
- Solarium can now deploy to SPOTs using old SDKs (provided OTA & configuration page format).
- Shared base-station support to radio view.

Sunspot runs on Squawk Java Virtual Machine. The Programmer's Manual of Sun SPOT- in his release v6.0 -yellow- specifies all that we need to understand before to start to use Sunspot and of course how to start to use it. Then a brief breakdown of the parts that I have focused on the manual:

SUNSPOT TOOLS: SUN SPOT MANAGER AND SOLARIUM.

The Sunspot Manager Tool allows us to query information or modify parameters for each of them, such as: start-up configuration (which configuration is running on Sunspot, state of OTA command server also enable/disable it and set up a Base station); features (MAC address and downloaded version as well as update and install new SDKs versions); configuration properties (specific information as battery model and version, so on); current libraries; console (displays the command line output of this tool, every command executed successful or wrong is shown in this tab); solarium (launches a tool to start Spot emulator that lets you test/load an application software and interact with each Sunspot) and docs tab (browse and view currently documentary and Spot Tutorial).

Moreover Solarium Tool, started by solarium on Spot Manager, it's a useful

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

tool where you can carry out different functions with each SPOTs remotely. This functions are: Discovering and displaying SPOTs, Interacting with SPOTs (test and load software on Spot), Managing a Network of Spots (to know which application should be run and deploy it on each of them) and Emulating SPOTs (we can make virtual Spots and run applications on it).

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Chapter 4

4. System Component Design

4.1 Analysis and study of system components

This project has been made on a open source platform, that allow us develop a program to work with the environment, this means that both the code developed by this project as the ideas can be developed by any other person wishing to extend the application as well, existing codes have been reused from it to achieve our goal. The idea is to provide a different view in the field of WSNs, with this approach is intended to give to collaborate on the development of technologies. Providing necessary to continue its development.

This reduces the cost of developing an application, affording a tool for the development of WSNs.

Once evaluated on the devices involved, the characteristics, hardware and software, this chapter must adduce economic valuation and feasibility of the project. (Possibility to make it)

The Sun SPOT platform provides the necessary to develop your own application, the Netbeans environment used is based on Java programming.

To develop the application using the Bluetooth module was used Eclipse workbench, which like Netbeans, is based on the Java programming language, both have a similar environment and way of working is pretty similar.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

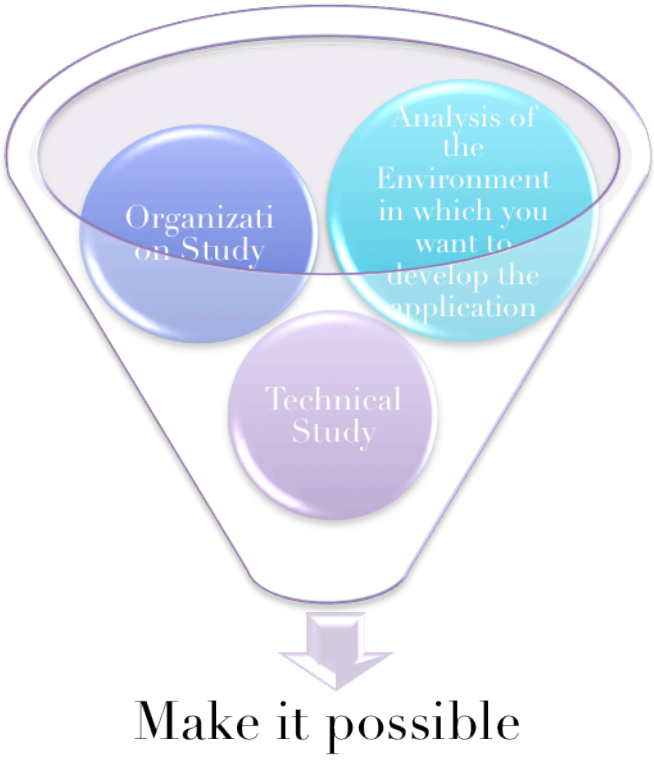
For the design of the board where is adapted the Bluetooth module, we used the EAGLE program, which allows the design of printed circuits, the cost of the printed board and the material used is quite cheap.

All programs involved both the development of the application and design, are free to download, so do not add any cost to our implementation.

Then a breakdown in general prices of both the platform used as a Bluetooth module that has adapted to it, which is part of economic plan. These components have greater repercussion on the final price for the development of the application.

Sun SPOT Java Development Kit [27]	US$399.00 / Pack
Bluetooth Modem BlueSMiRF [20]	$64.95

However, the viability of the project is not only measured in economic terms, other relevant aspects of their study and those listed below:



© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Analysis of the environment, WSNs are currently increasingly embracing in the consumer market today, as noted throughout the project development, being a 'new' technologies, creates different guidelines for implementation, this is where we place our idea, to submit a new application.

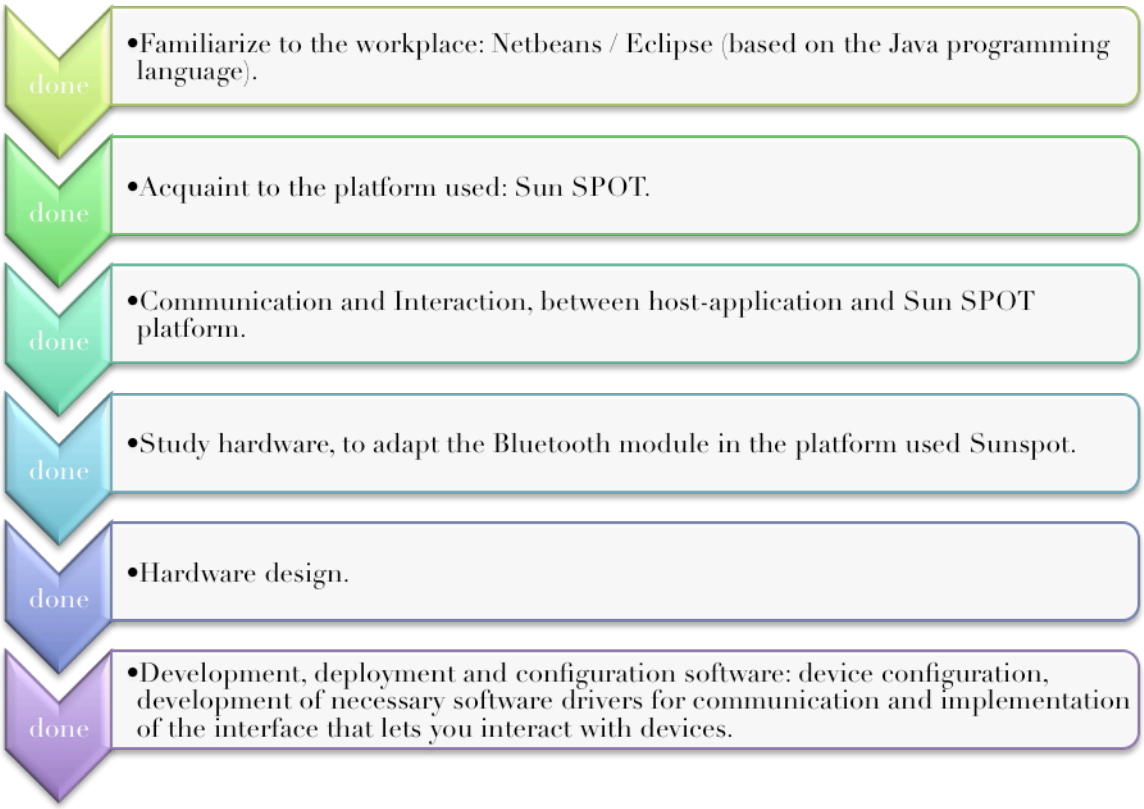
The project development has focused from the beginning, in the interaction with our environment providing ubiquitous access to end-users, in the WSN and mobile applications, to get it, we have followed some phases (such as studying the platform to use, study of the Software / Hardware involved, related work on WSN...) that establish the necessary guidelines to achieve complete the project.

Figure 4-1. WSN Application Areas [28]

Organization Study, allows you to set guidelines for how it will develop the project, an idealized time, we focused on what you want to work.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

The following steps were introduced in this idea to organization study:



Technical study corresponds to the possibility of developing the application at both hardware and software, once valued what you want to do and how you want to do it, assess whether the correspondence theoretically obtained can be carried into practice.

Depending on the applications you want to adapt the project has been developed to bridge the gap between the WSN and the final application through the Bluetooth module adapted to the platform.

In conclusion, taking into account all the characteristics denoted above, the bill becomes entirely feasible, both in technical study as economic terms.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

4.2 Development and Implementation of the System Components

To make the connection with SunSPOT has begun familiar with the devices involved, following the next steps:

In a first time, making the connection between SunSPOT and Host Application:

The communication established between SunSPOT and Host Application is through Base Station, to connect with the base-station they open the Radiogram Connections, this connections is a protocol used to access data between two devices, generally as client and server, based in IO communication.

The Host Application run the application becoming to connect with the Base-Station activating other initialization corresponding with it, like as: sharing base-station (to allow connect another Sun SPOTs), Mac Address, port used… then will open the IO Radiogram Connection, this will be explained on 'Software drivers for hardware platform'. In our evolution about how to connect with the WSN, the program implemented in the end interface, is the same from the start, for that, we will explain this on the next sections. For now we will focus on the general steps to get the final result.

Also, once initialized the base station (BS), it will be listening for Sun SPOTs ready to connect, on a specified port. For each device found, a new interface is runs, and creates a connection between host application and the Sun SPOT, the interface specifies the ID of the Sun SPOT with which we are connected, this ID will match with device's MAC address (IEEE Address).

In the first instance, this allows us to obtain data from the SUNSPOT with which we have connected, the data obtained from them are referred to Sensor's

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

values and at the same time send a order, meaning, getting data as: light value, temperature value, accelerometer data, or send to Sun SPOT a particular task as Blink LEDs.

In the follow figure, we have an aspect of this interface:

For each of the modules forming the interface, has been designated a number, thus differentiating the characteristics of each of them by assigned number.

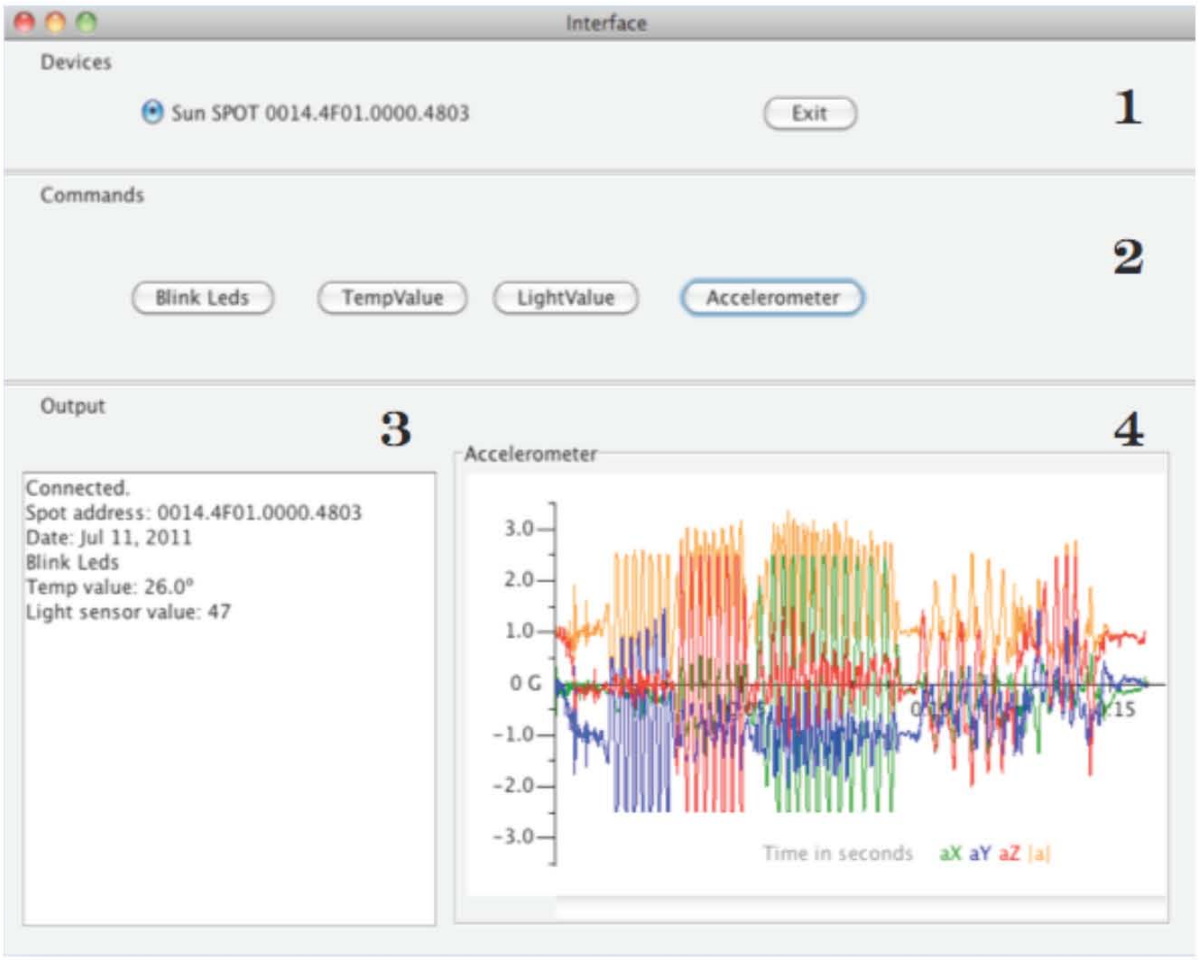
MODULE **1**: In this module the assigned name is 'Devices', specifies the Mac Address of the Sun SPOT that is already connected, here allows finish the application selecting the 'Exit' button.

MODULE **2**: This module contains the commands that allow us get data from remote SPOT or send a simple command. So, the module's name is 'Commands'.

MODULE **3**: Here, the module is dividing on two parts, one of them contains the Output line (module 3), and another is where it makes a graph from the data of remote SPOT (belonging to module 4). Both belong to the output module. The module 3, we can see state of connection, if it lost the connection with a remote SPOT through this output window we knows, date of the already connection, IEEE Address, and the action command (for the Light value and Temperature value, it show us the values of each event and in case to send a simple command as Blink LEDs, it specifies the action).

MODULE **4**: Show us the graphic makes from the data obtained when we press the 'Accelerometer', the graph also shows the legend with the meaning of each axis. The Accelerometer makes a measure of acceleration in three dimensions, by providing guidance on the location of the object on the earth plane relation, and taking into account gravity. The axes X, Y, Z represent the 3-Dimensions, which are drawn on the graph (green, blue and red, respectively) in the time that BS get data values in the three axes.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013



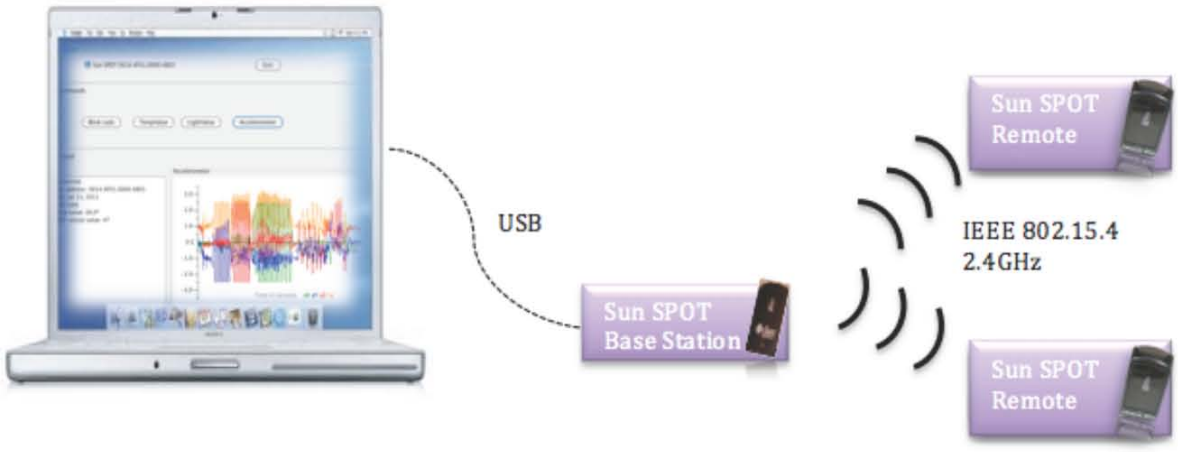


Figure 4-2. Interface implemented in the PC between Host Application and Sun SPOT and overview communication between them.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Once we know how is working the connections between them (Sun SPOT_A - Base Station - Sun SPOT_B) and all of them with the Host Application through a Base Station, then we can make the design of the hardware platform: Gateway, corresponding with the following section.

From there, we made a connection, for one hand, between Gateway and Sun SPOT_A, extending the connection on Host Application. Then trying to connect the Gateway with the Host Application. Once ready the connection with the Host Application the next step given was between Sun SPOT_A and Host Application through the Gateway. Thenceforward we are ready to make the connection between to Sun SPOTs (A and B) with the Host Application via Gateway, getting the data that end user want through an interface.

Subsequently, it explain how is working the interface in general terms. However, everything that refers to the internal configuration and hardware design will be explained in the following sections.

The final interface represents tasks pretty similar to the last one.

The mobile application is based on the same code that the PC application, the operation is the same and adds features that are also the same as the PC interface, with the exception of accelerometer graphical, which is not added in the mobile UI Application.

Then we will explain the main functions of the UI application for PC similar to in the UI application for mobile. Then we will represent the overview about the communication in both PC and Mobile and also the screen shot of the interface.

The UI for mobile application allows you to connect / disconnect the Bluetooth module and then select the device you want to connect from the list and once the connection to the remote device has been established, data can be request

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

to the remote node, like light value, temperature value and send a small flash LEDs, thus verify the user's interaction with the remote sensor node.

Below show the modules that make up the UI application for PC created for interacts with the remote node:

MODULE **1**: The first module represents the state of the connection, once connected to the Gateway through its BlueSMiRF module, shows the server URL you are connected, indicating the MAC address of the Bluetooth module. The buttons 'Connect' and 'Disconnect' will allow us to connect and disconnect remote Bluetooth module.

MODULE **2**: Is the 'output' window, here will be reflected all actions performed as well as the state of the tasks that are running at the moment. Reflects the state of the connection to the Bluetooth module, presents the data received from the Gateway (Temperature value, Light value, Action command-Accelerometer and Blink LEDs) and the actions that require end-user intervention to execute it, as is the if selecting a remote device to which we want to connect in a WSN.

MODULE **3**: 'Devices(s) Found(s)' a list of devices you want to connect in a WSN for the moment, our application is configured to receive a maximum of four available nodes from the WSN and to which we connect. This is because in testing the application of a single WSN, we have two nodes of Sun SPOT platform. This configuration does not have limitations, because in future applications, this parameter could be modified.

So, we select from this list the device with which you want to make the connection.

MODULE **4**: 'Commands'. As the last interface, here you find all the commands action enables to interact with a remote node.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

This commands actions are:

'Temp Value' / 'Light Value' get the temperature/light value from the remote sensor node.

'Blink LEDs' send an action to the remote sensor node, making the blink LEDs in 4 times.

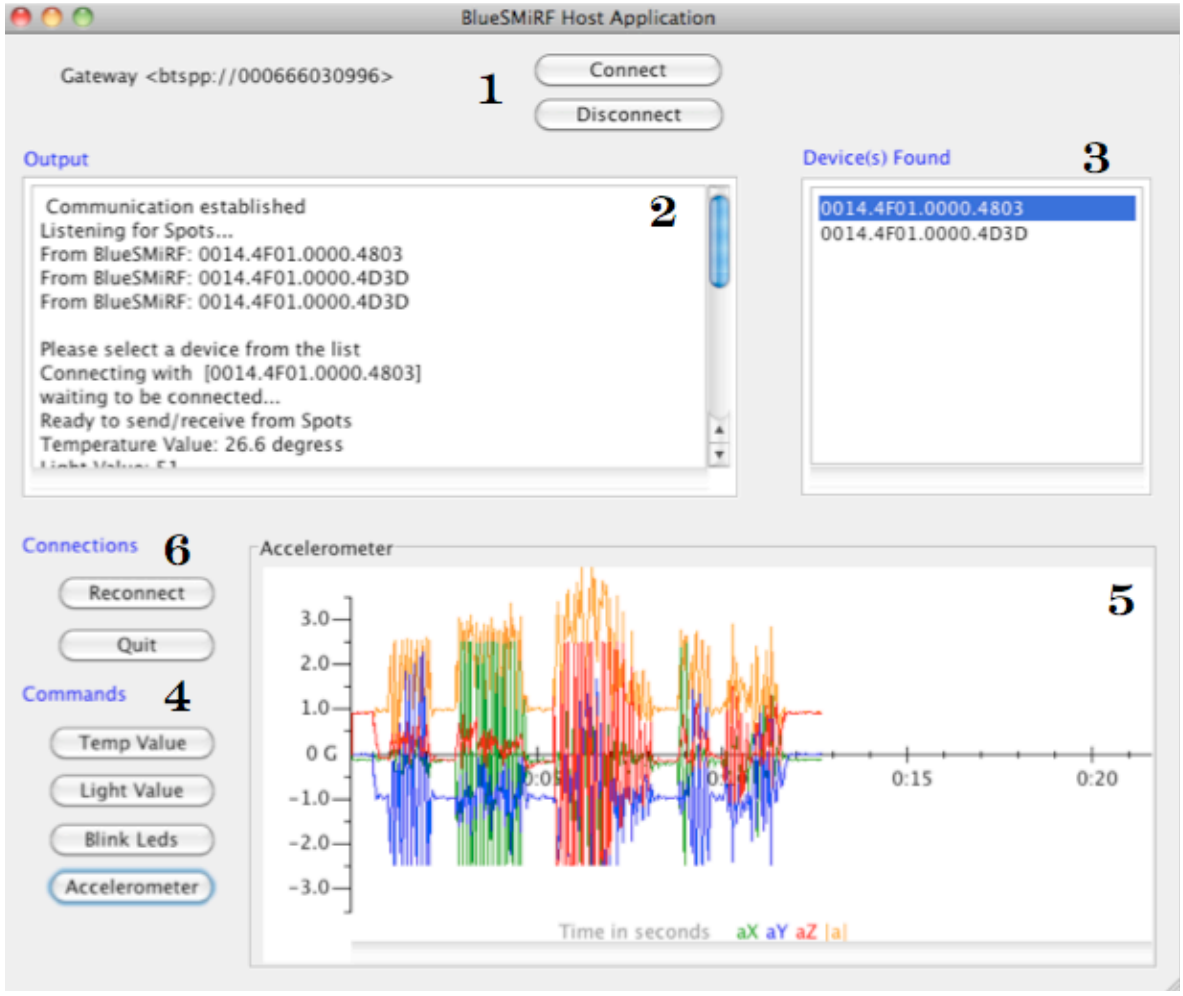
'Accelerometer' sends a request to get data from the Accelerometer in the remote sensor node, this button has two functions behind, these functions are: Get Data and Stop Getting Data.

MODULE **5**: Accelerometer Graph, represents data from the accelerometer in the node we are connected, indicating the same way that in the previous interface, the value of the accelerometer data in three dimensions, denoted by the same axes: X, Y, Z.

MODULE **6**: Concerning the remote connections, we can connect/disconnect from the remote node; select another devices from the list of devices found, and 'reconnect' with other sensor node or 're-establish' the connection with the last one.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

PC BASED UI APPLICATION



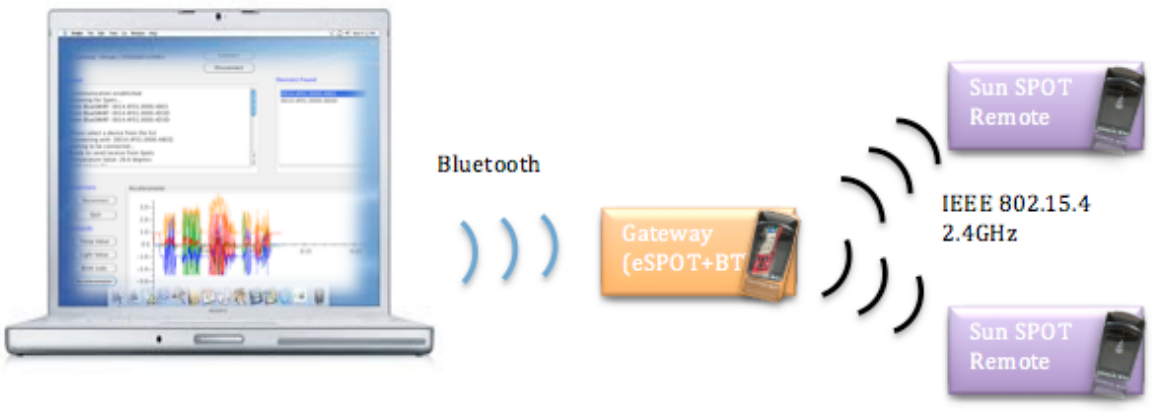


Figure 4-3. Interface in the host application to interacts with a remote SPOT & the schematic communication between devices through the Gateway

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Mobile phone based UI application

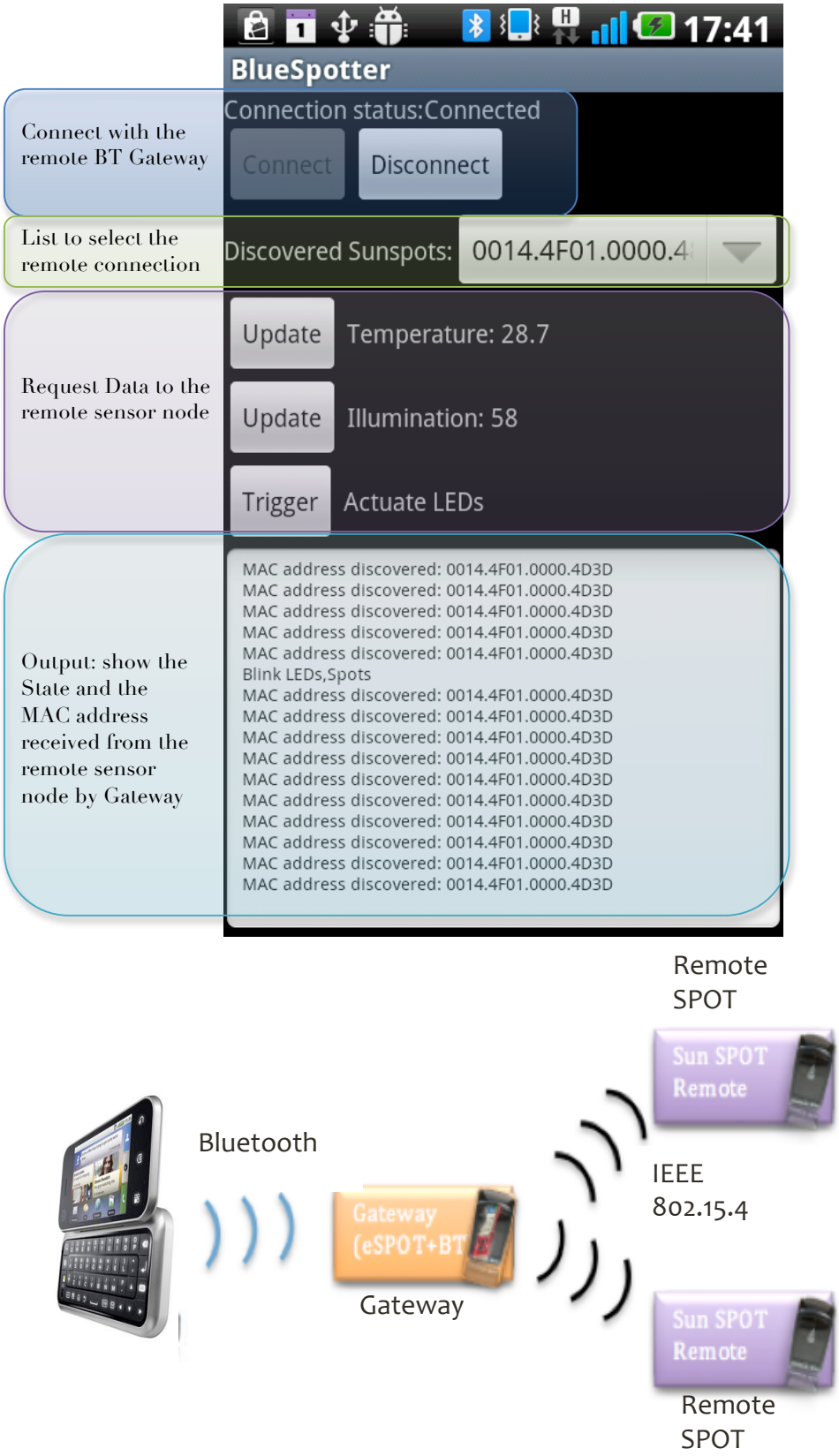


Figure 4-4. UI Application for mobile and Overview of the system communication

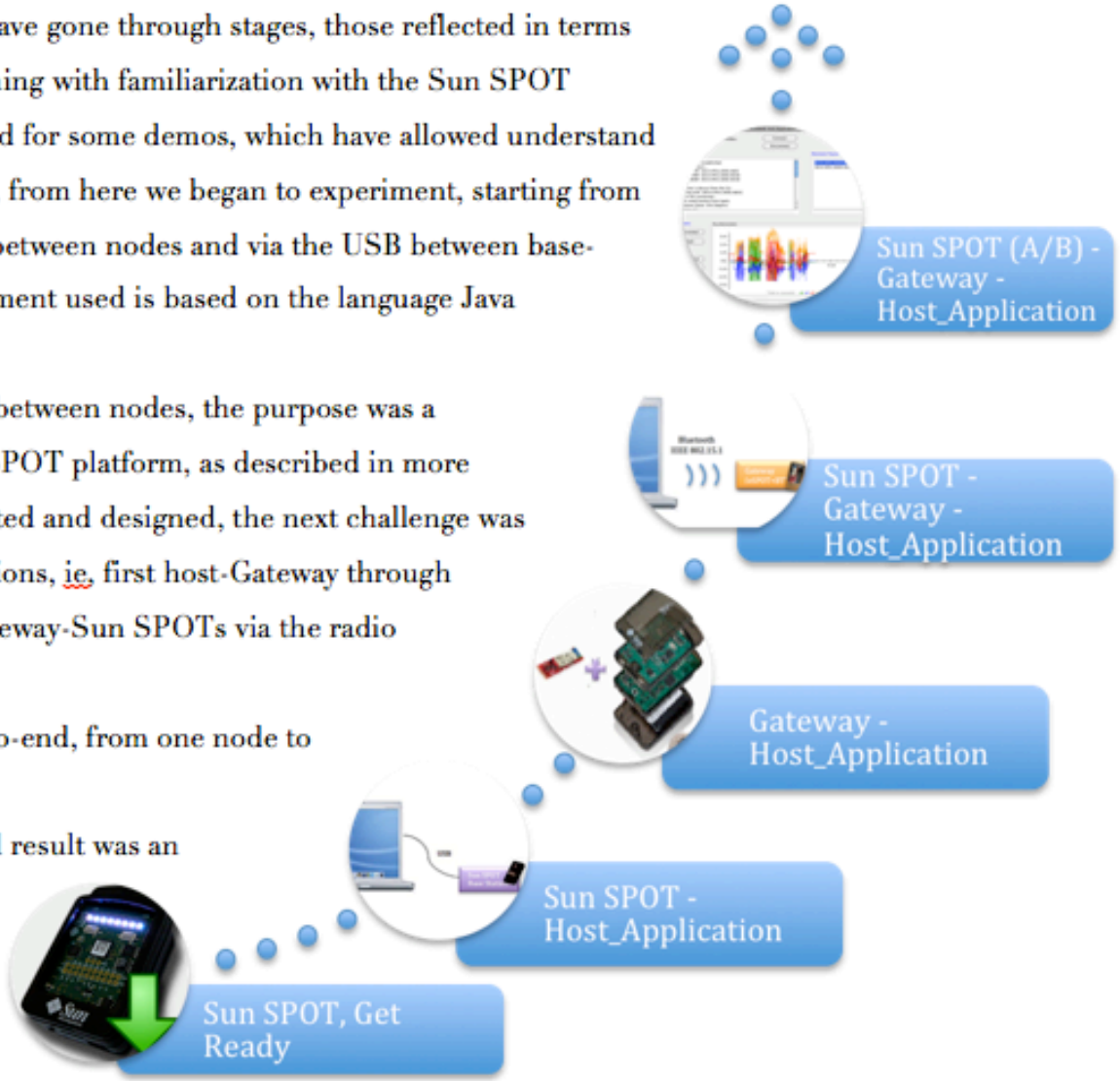
© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Throughout the project development, have gone through stages, those reflected in terms general in the chart on the right, beginning with familiarization with the Sun SPOT platform used, this platform is scheduled for some demos, which have allowed understand functionalities and properties of a node, from here we began to experiment, starting from the basics of communication (via radio between nodes and via the USB between base-station and host), the Netbeans environment used is based on the language Java programming.

Once familiar with the communication between nodes, the purpose was a Bluetooth module to adapt to the Sun SPOT platform, as described in more detail in other sections, once implemented and designed, the next challenge was to achieve communication in two directions, ie, first host-Gateway through the Bluetooth module and secondly Gateway-Sun SPOTs via the radio communication.

Finally, make the communication end-to-end, from one node to host application through Gateway.

Once mastered communication, the end result was an interface that allows you to select the node that we want to connect and get the data that the user wishes.



© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

4.3 The Hardware Platform

Gateway

To the design of the Gateway, we have used the Sun SPOT platform and a Bluetooth module RN-41, with the specifications mentioned in the previous chapter.

EAGLE is an easy tool to use for designing printed circuit boards.

In order to make printed circuit design and adapt the Bluetooth module RN-41 (hereafter BlueSMiRF) were taken into account, besides the technical specifications contained in the previous chapter, the schematics pertaining to the devices involved: Processor Board on Sun SPOT provided by Sun Microsystems, hereinafter 'eSPOT', as well as the BlueSMiRF module.

eSPOT, is the main board with the rechargeable battery. Also, contains the Main Processor (AT91RM9200, referenced in last chapter), thus allows us through a 'connector' to have access to USART connections, proportionate by this microcontroller; Memory; Power management circuit; 802.15.4 radio transceiver and antenna.

To allow the connection between boards has designed a custom library, corresponding to the DF17-30 HDR (HEADER). The connector is of the DF-17 Series with 30-pin.

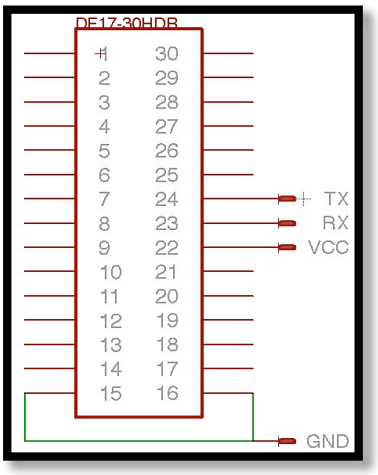
Figure 4-5. Hirose Connector: DF17-30 HDR. [27]

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

On the other side of the eSPOT board, incorporates the connector DF17-30R (RECEPT), thus being able to adapt our design.

EAGLE Design

Schematic of DF17-30 HDR connector: since the eSPOT platform, there are four connections that need to make the schematic represents the following figures:



UART-TX: Pin 24.

UART-RX: Pin 23.

VCC: Corresponding to pin 22.

GND: Pin 15 and 16.

Figure 4-6. Schematic of connector DF17-30 HDR generated in EAGLE.

Plate corresponding to the PCB: which will adapt the eSPOT module to the BlueSMiRF. Here we have taken into account measures the actual size of the board that we are replacing the Sun SPOT platform of 'eDemo Board'.

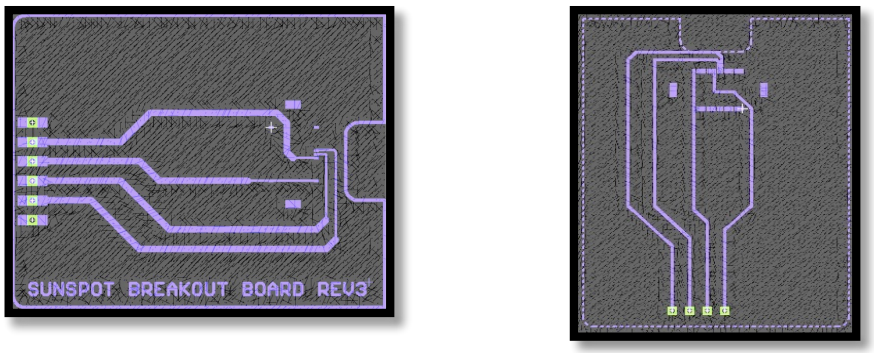


Figure 4-7. Represents the final outcome of the printed wiring board, to which we then added BlueSMiRF module connector and soldering points of interest mentioned above.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

At the top, represents the design of the DF17-30HDR (presented in Figure 1). And at bottom, represent the points that will allow the interconnection with the BlueSMiRF module.

BlueSMiRF

To establish communication between the module hardware and SPOT BlueSMiRF, the connection was made as follows [25]:

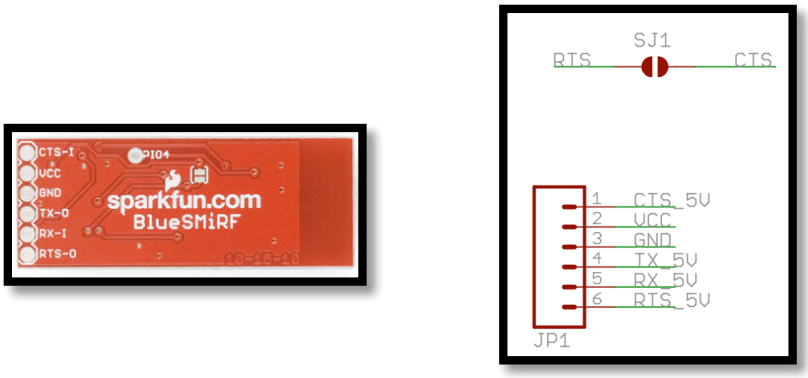


Figure 4-8. BlueSMiRF: Schematics & Module. [20]

- ✓ BlueSMiRF Radio TX with RX in the connector of Processor Board (ARM920T-based Microcontroller). UART_RX: UART receiver input.
- ✓ BlueSMiRF Radio RX with TX in the connector of Processor Board (ARM920T-based Microcontroller). UART_TX: UART transmitter output.
- ✓ BlueSMiRF Radio RTS and Radio CTS not connected, because between Bluetooth module and the Sun SPOT platform we will not consider the flow control, so is disabled.
- ✓ BlueSMiRF VDD 3.3 V (regulated power input) to VCC in the connector of Processor Board.
- ✓ BlueSMiRF GND to GND in the connector of Processor Board.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

State of the connection: The module has two LEDs, one blink at different speeds, indicating the status; the other directly reflects the state of the connection, passing HIGH state when you connect and LOW otherwise.

LED Status:

Configuration: 10 times per second (Red).

Startup/Configuration Timer: 2 times per second (Red).

Discoverable/ Inquiring/ Idle: Once per second (Red).

Connected: Solid ON (Green).

The Bluetooth incorporates certain characteristics such as **latency** and **performance optimization:**

The firmware should make decisions, automatically, the moment the data are received through the UART RX, the radio link. The default is to optimize performance. Sometimes the reception of data can be partial, which means it is divided into several parts, arriving in the same order they were sent, but sent at different intervals, this occurs in cases in which the space between the data incoming is not wide enough. One of the limitations of Bluetooth is which has algorithms that can cause quite significant latency between packets (greater than 10 ms) at certain times. If we add the delay that can introduce a host application, we therefore continued at the reception errors. There is a latency optimization method, which forces the radio to keep the data bursts together. This method can enable a debug command to the Bluetooth module itself "SQ, 16", setting the bit of latency in the firmware, to disable the command to use is "SQ, 0". To access the registry value, you can use the "GQ", extracts the data as HEX but is set to decimal.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

4.4 The Software drivers

To focus on software development, both incorporated algorithms for eSPOT platform configuration, including own configurations of Bluetooth module, will be presented below, as well as its structure and internal communication between them.

"BlueSMiRF"

To configure the BlueSMiRF module we need to enter in command mode, sending the characters '$$$' through the UART then we can change the data configuration and performance of Bluetooth module. [25]

Within the Bluetooth settings, we have several categories that encompass various commands, given that we will focus in the basic operation of the module, highlights only the commands needed for proper operation.

Bluetooth Configurations.

This is where you define the minimum requirements that the Bluetooth module must to have to be compatible with others devices.

These requirements define the Service Class, Service type and Class of Device, together all the characteristics and procedures, which has to have the module to be compatible and to allow interoperability with other Bluetooth devices.

Settings	Value
Bluetooth Service Profile.	0 SPP (Service Port Profile). Generic profile that allows defining specific profiles depending on the operation or use, as well as defining protocols and features that supports the model to use.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

	Allowing proper communication between Bluetooth devices using RFCOMM.
Device Mode	0 (Slave)
Baud Rate	115200 bps
Parity	None
Data bits	8 bits
Stop bits	1 bit
Power Mode	Auto Low power discoverable mode
Name of Device	Gateway-0996
Service Name	Bluetooth
Service Class	0002 Networking = 0x020000. Allows the configuration to determine the service offered, so when other devices are looking for a specific service, this information along with the device class, determine the function of the module and protocol to use.
Device Class	0300 LAN/Network Access Point = 0x0300. Using the Point-to-Point Protocol (PPP-used as a means of network access) over RFCOMM, to defines the procedures to enable Bluetooth devices to access LAN. Depends on SPP. Operating as if connected directly to the network. Does not require the use of any particular protocol.
Authentication	0 (Disabled)
Encryption	0 (Disabled)
Discovery Enabled	0x0200 Inquiry Scan Window: 320 msecs.
Connection Enabled	0x0200 Page Scan Window: 320 msecs.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Bonding	0 (Disabled)
Configuration Timer	60 seconds
SNIFF mode	0x0050 Power conservation method, radio wakes up at intervals = 50msecs
Default PIN	1234
Local Echo of RX chars in command mode	OFF
Connect/Disconnect Status String	Null

Table 4-1. Bluetooth Configurations

Processor Board - "eSPOT"

To configure the Gateway, based on the Sun SPOT platform, has created an algorithm that, among other configuration includes the following subroutines to make the connection between the Bluetooth module and eSPOT board.

The communication between the eSPOT board and the BlueSMiRF are performed through the USART connections.

USART Connections:

Universal Asynchronous Receiver-Transmitter performs data conversion from serial port to parallel format, transmitting the bits sequentially. [26]

The guide for the Sun SPOT platform in its section "Using input and output streams over the USB and USART connections", specifies everything we have to take into account, both open and close connections UART as the limitations we have to take into account and that inevitably enter the platform.

Then there will be a brief outline of how these connections have been made, taking into account outlined in previous chapters.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Access to the USART connections do not provide suitable connector at the top of the SPOT, it will connect our plate design, which in turn adapts the Bluetooth module and using that connection to enable communication between them.

USART connection opens the same way as the USB. This feature will be incorporated only once at the beginning of the program.

The parameters involved are to make the connections are 'input' and 'output', then specifying the 'serial' type we want to use both if USART as if we specify the connection is USB. In our case and we will refer to USART connections.

```
InputStream input = Connector.openInputStream("serial://usart");
OutputStream output = Connector.openOutputStream("serial://usart");
```

To read the input data can implement different subroutines, in short, the reading is done as follows:

```
byte b [] = new byte [0x0011] / / define a byte type variable where we store the input data.
input.read (b) / / read the input data and store it in the previous variable.
String s = new String (b, 0, b.length) / / convert data bytes read from a String, thus sending the reverse process.
```

Consequently, to send data over the 'output' using the following line:

```
output.write ("String". getBytes ());
```

'String', represents the string you want to send.

'getBytes ()' is a proper function that transforms strings into bytes and thus be sent by the Stream output.

LIMITATIONS USING USART IN THE SUN SPOT PLATFORM:

Take into account that in the boot the Sun SPOT, some characters are sent via the USART connection, corresponding to the configuration, thus affecting the data received. These characters are necessary to allow communication between the

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

application SPOT and SPOT client.

To avoid this problem, we introduce our subroutines for reading input data, a new task, which will clean the input buffer just after the restart of the device and prior to receiving the data of interest.

```
while ((i = input.available()) > 0) {
        input.skip(i);
    }
```

'input' represents the connection open entry

'i' length of characters currently available at the entrance.

The function 'skip' will ignore the data so far are taken into the input buffer.

"USART input use a 512 byte input ring buffer", which means that there may be buffer overflow, therefore, depends on the application that will be implemented to control data to be read fast enough to not to lose data, knowing the transmission rate and transmission rate.

One of the difficulties encountered in the implementation for receiving data related to this limitation. To read data sent from the Bluetooth module to Sun SPOT without losing them, we had to insert a small delay ('sleep time') between reading and writing, not significant for final execution of the program, but enough to keep data received via USART.

For reliable reception we recommend using the highest baud rate supported. In our case, the Bluetooth module is configured to the maximum transmission speed, thus ensuring better data reception.

On the other hand, the Gateway communicates with other devices based on the same platform (SUNSPOT) and forms the WSN. This communication is established through the Radiogram protocol.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

RADIOGRAM PROTOCOL:

Enables the exchange of packets between two points, the protocol defining 'radiogram' is based on datagrams.

To make such a connexion is necessary allocation of those ports on both ends, plus the IEEE address, which will mean that communication will be established for these ports and corresponding IEEE addresses.

In the first instance will use a 'Broadcast_Port' to locate the host application. This port will remain at the 'listening' applications 'customers'. The Radiogram Connection opens as follows:

```
rcvConn = (RadiogramConnection) Connector.open("radiogram://:" + BROADCAST_PORT);
```

Then the port type 'Connected_Port' will be used for sending commands and responses between the remote location and the host application. Since RadigramConnection supports broadcast mode, the outgoing connection is made by a port specified (Connected_Port) but being sent in broadcast mode, which means it will all "customers" ready to connect to the host application through the given port.

```
txConn = (DatagramConnection) Connector.open("radiogram://broadcast:" + CONNECTED_PORT);
```

Ports numbered 0 through 30 are reserved for system services, therefore, and in order to avoid conflict in the application, the ports used are: 42 (Host Name Server Protocol) and 43 (WHOIS), is to query and response protocol widely used to query the databases that store registered users. To connect the port to specify that we want to connect and provide the IEEE address end with which we connect. Once established the connection, packet switching is based on the datagram created, both sending and receiving.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Diagram of Communication between devices

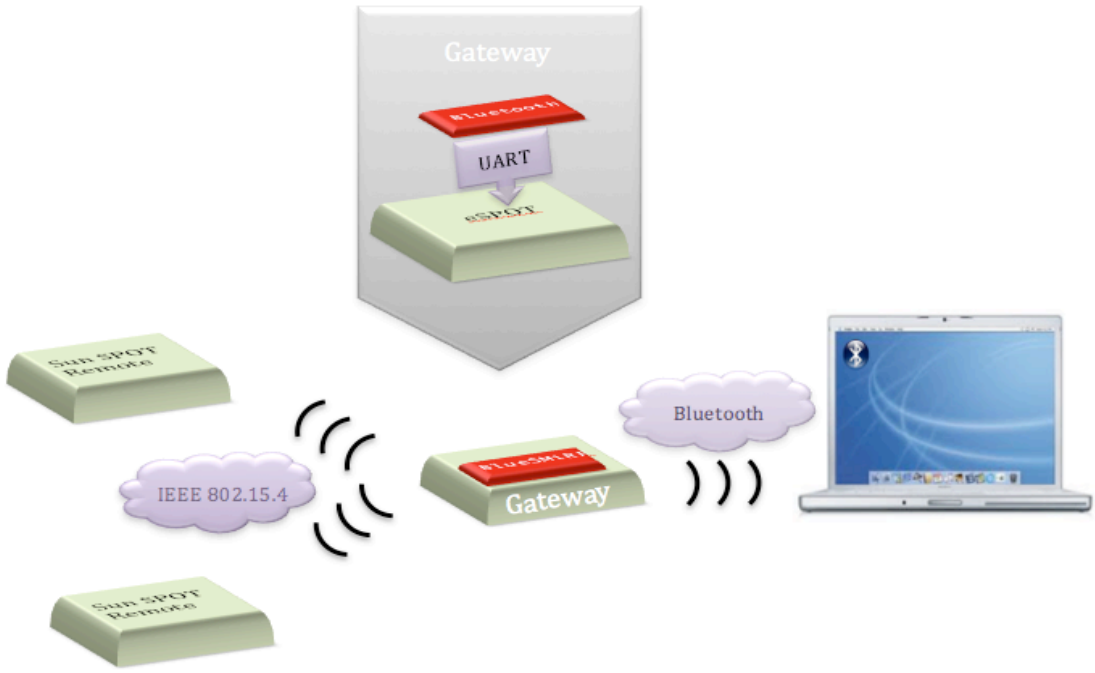


Figure 4-9. Communication Devices

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Format of messages sent from the Gateway and received in Eclipse Application:

All the message are in String format, limited by a coma to extract the interest message in the host application.

The Output Stream Sends the String through a subroutine that getBytes() from the String that we want to send.

ID = is the identifier of the message.
Then separated by coma Message.

ID	Message
0x01	Ok = Bluetooth connection is ready. (State of the connection). Or a message to show in the host application the state of the Gateway.
0x02	IEEE address from remote Spot.
0x03	If the message contains 'degrees' will show the temperature value or Light otherwise. Parameters in the Gateway: *@ lightValue parameter read from the remote Spot.* *@ tempC parameter read from the remote Spot with the temperature value in degrees.*
0x04	Accelerometer data in the three axes: x,y,z.

Table 4-2. Message Format Type on the Gateway

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Format of messages received in the Gateway and sent from Eclipse Application

We are expected to receive two bytes, these bytes mean:

ID	Type
0x00	The Bluetooth connection is established.
0x01	Request Light Value.
0x02	Request Temperature Value.
0x03	Blink LEDs (in 3 times).
0x04	Request Quit Server. Disconnect to the remote Spot.
0x05	Request Accelerometer Data.

Table 4-3. Message Format Type on the APP

Received packets to interact with the remote SPOT, once received at the Gateway, the latter has its own subroutines to perform the request to the remote SPOT, sending new packages.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Diagram Flow Communication of the Java code in the UI Application:

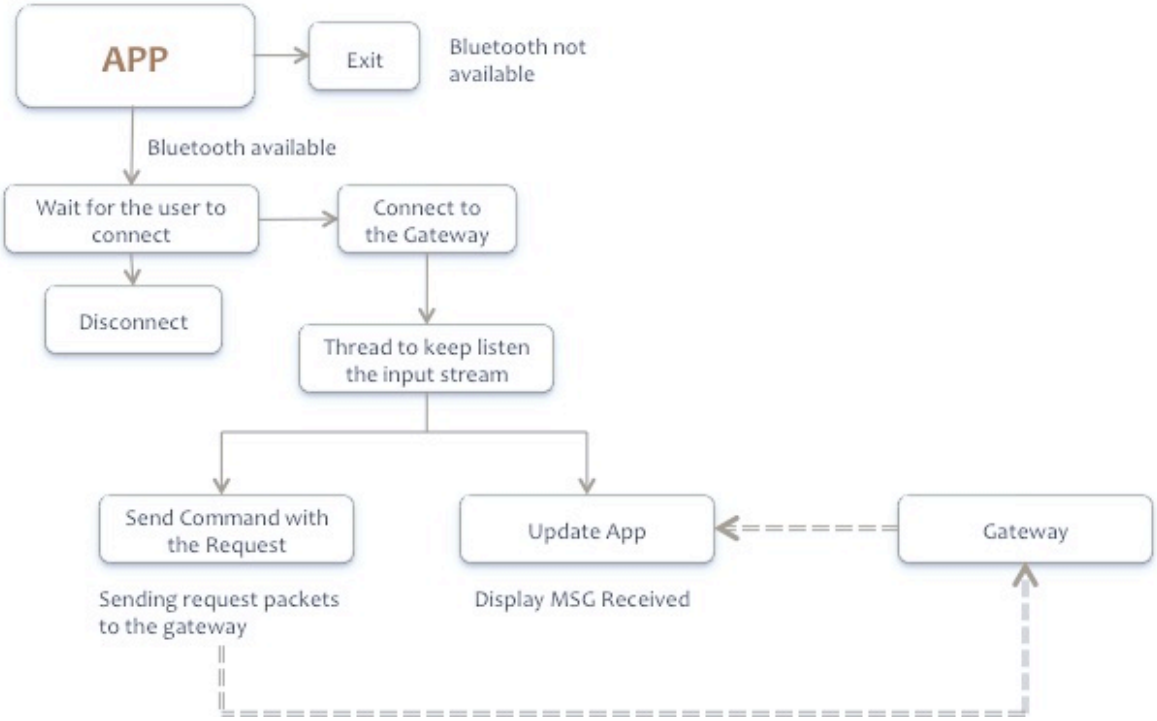
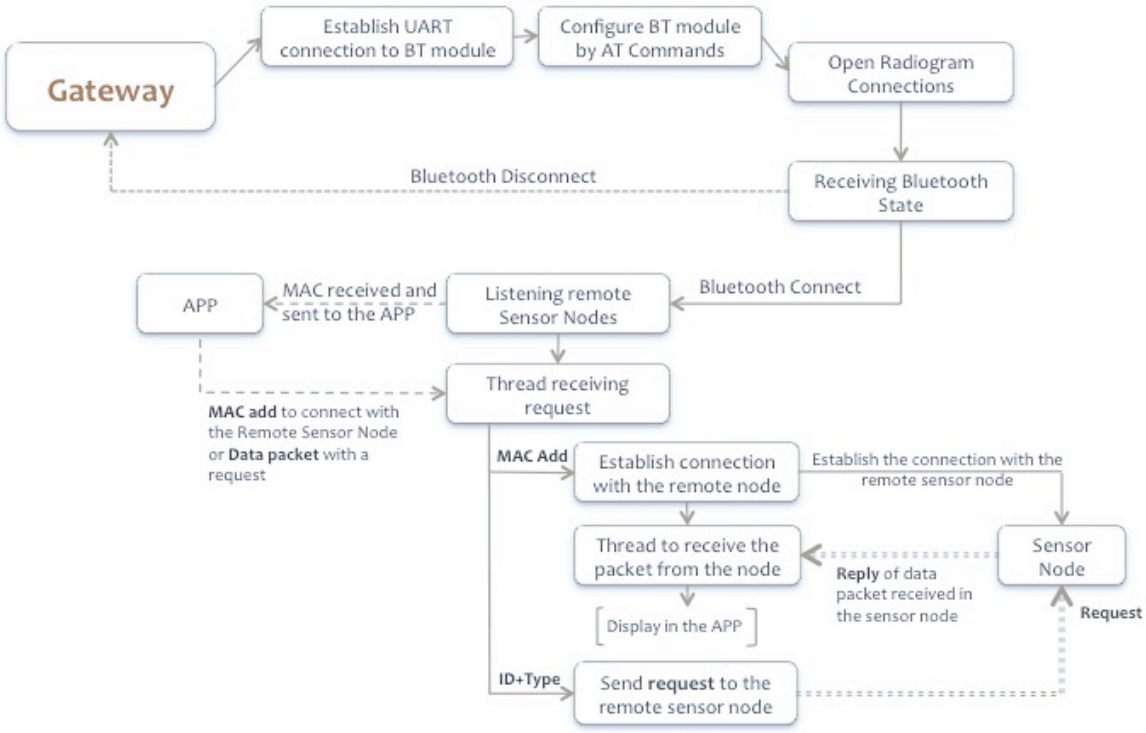


Diagram Flow Communication of the Java code in the Gateway:



© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Chapter 5

5. Conclusions

Despite the difficulties encountered during the project development, both in terms of software (one of them: the synchronization and waiting intervals between packets sent and received by communication UART) and in hardware level (as was the if you create a PCB to adapt the Bluetooth module to the platform Sunspot, this is because the eDemo board, only allows communications of low speed over UART and considering that high baud rate is equals to less waiting intervals, it could be "obsolete" the implementation in the board eDemo).

The purpose of this project was to create a Gateway based on Sunspot platform, which can communicate with a network of sensors and make this information accessible to mobile users.

To enable mobile communication, it was necessary to design a PCB to which it could adapt the Bluetooth module and connect the other side of our Processor Board on the Sun SPOT platform.

The Gateway is configured to obtain data from the nodes of a network of sensors, the application can interact with the device and obtain data from a particular node in a WSN found, has been developed primarily for the computer, being able to obtain sensor data on the node, such as light value, temperature value, accelerometer data, through which it has generated a graph that shows the mobility in the three axes of sensor data obtained as well as send simple tasks, in this case, there has been a flicker of LEDs (3 times) to check one more time connectivity to the remote device.

In turn, the applications deployed on the computer closes the connection to connect to another remote node at the same time, remains the "listening" to connect

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

new nodes available, this listener is obtained through the Gateway MAC addresses for each device found, and through which we connect.

On the other hand, the nodes run a simple application, which has shown that this operation is to record the packets that are prompted by the Gateway, and in every situation knowing the answer has to give, obtaining this so desired by the final application. The node remains "available" to connect once the connection information exchange in both directions (application - node, node - application) through the Gateway, that is, there is no direct communication between nodes in the WSN and end-user application, the interaction has been registered in our Gateway.

The Gateway, in a real situation, behaves as a server, staying on the "listen" for new connections from Bluetooth module to initiate the connection to the remote node. This allows access to any end user through the application, a WSN node.

As innovative system has provided a new approach to application in the field of WSNs, providing a simple sample application, as well as an implementation of low cost and adaptable to user needs.

Although the application has been developed as a simple demonstration example, the project provides the tools for improvement and may well provide a breakthrough in the growth of WSNs and technology.

Remarkably, the project has been developed from scratch, I mean that besides include phases described above, was designed from our own aims and purposes, keeping a simple idea but enough to reach the final application. So in this section, the improvements can be added to the project can be broken down into two branches principals:

At the level of application software, in the developed application, can be created more complex application environments than it is, could be include real applications, I mean that reflecting the activity of a global network WSN or whatever related with the

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

purpose of the WSN, in our case, the application was focused on a simple example application that enables data exchange and the establishment of communication, those being the main objectives of the implementation.

In terms of configuration, the devices (both nodes such as Gateway, remaining that includes Bluetooth module) have been configured analogous form to the work it's going to perform, by which I mean, that your configuration is intended for the interface implemented. So, we return to keeping it simple and basic to work with the prototype.

On both counts, you can add improvements, capable of supporting any application and deploy a generic interface.

Finally, the development of both the prototype as well as the host application, allow it to be a starting point for implementation of a Gateway, based on Bluetooth communication, it extending the access to end-users, and allow radio communication to accessing nodes in a WSN. Adding to turn features like low power consumption and low cost of implementation.

5.1 Personal Conclusions:

Develop this project was a challenge since the beginning so, my objectives achieved cover more than I expected, they are the following

Take another point of view in technologies with the depth study of Wireless Sensor Networks, Bluetooth, Zigbee and Sun SPOT platform used in the application. Also, get the enough knowledge to develop and understand the codes based on java programming.

Research and specification of the different system components: Sun SPOT platform, Bluetooth/Zigbee, allowed me to understand the communication and

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

functionality between them.

Design and implementation of a prototype hardware platform. Here, the most important is check the connections already made and the connection that we want to do, then double check in the program design and making sure that we will have the correct connections. I hope don't make the same mistake another time.

Development of the necessary software driver for the hardware platform (Gateway, Sensor Node, Application). Finally, get it work. Definitely the best one.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Appendix A

Spanish Summary

Este apéndice engloba un resumen de cada capítulo en español, para mayor información o detalle, sería conveniente consultar los capítulos en inglés.

A.1. Capítulo 1: Introducción

Después de más de una década de investigación, las redes de sensores inalámbricas (WSNs) están lentamente introduciéndose en el mercado a la vez que experimentando el despliegue comercial en diferentes ámbitos de aplicación. Ejemplo de tales aplicaciones iniciales son la monitorización (supervisión) energética del entorno en el hogar, de la salud, del medio ambiente, medición de características físicas o automatización de edificios.

Muchas de estas redes inalámbricas de sensores se basan en la tecnología de radio y red siguiendo el estándar Zigbee (en particular, el estándar IEEE 802.14.5 MAC-control de acceso al medio- y capa física), que está optimizado para la comunicación eficiente de la energía a bajas velocidades. Los datos de estas redes de sensores suelen acceder a través de dispositivos de gateway, que implementan una tarjeta de red con la interfaz de radio apropiada. Mientras que estas tarjetas de interfaz están disponibles para los dispositivos de entrada basados en PC, no existen implementaciones comerciales que sean apropiadas para los diferentes dispositivos de telefonía móvil basados en Gateway.

La idea principal del proyecto es proporcionar a los usuarios de teléfonos móviles el acceso ubicuo a los datos del sensor de WSN, implementado en el entorno circundante. Casi todos los teléfonos móviles de hoy en día, proporciona

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

una interfaz de comunicación Bluetooth para conectar dispositivos periféricos, como auriculares e intercambio de datos de corto alcance, con otros dispositivos pares y / o computadoras personales. Explotando este hecho, el propósito del proyecto es diseñar y desarrollar un Bluetooth Zigbee bridge y la compatibilidad de controladores correspondientes que permitan, a las aplicaciones en un móvil a otro teléfono, consultar los datos y acceder a una red de sensores basados en Zigbee a través de su interfaz Bluetooth.

La plataforma que se utiliza es de Sun SPOT estandarizado 802.15.4, que es capaz de interactuar con un nodo sensor (recogida de datos) o con los demás nodos de redes inalámbricas de sensores.

El objetivo principal es tener acceso a datos de sensores en la red de sensores por los teléfonos móviles y la creación de un entorno accesible desde usuarios con terminal móvil. Los dispositivos móviles están basados en Android. Básicamente se proporcionará un acceso adaptado para permitir la interconexión de extremo a extremo de teléfono móvil para redes inalámbricas de sensores.

A.2. Capítulo 2: Estado del Arte y Trabajo Relacionado

Los avances en materia de tecnología y comunicación, han permitido dirigir las investigaciones hacia el campo de las redes de sensores inalámbricas (WSNs), hoy en día emergente en las aplicaciones, no sólo militares sino comerciales. Como se ha referido en éste capítulo, las WSNs se desarrollan en diferentes campos de aplicación, y cada vez son más los avances obtenidos en dicha tecnología, permitiendo a su vez la incorporación de otras tecnologías como los dispositivos Bluetooth.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Inicialmente los sensores estaban condicionados para obtener datos como vibraciones, presión, sonido, luz, temperatura. Aunque sus primeros usos iban dirigidos a aplicaciones militares (como detección de intrusos, de sustancias químicas…) poco a poco se han ido extendiendo en aplicaciones convencionales y comerciales en diferentes campos, como por ejemplo: en el uso eficiente de la electricidad, entornos que requieren un alto nivel de seguridad (hoy en día utilizado en muchos entornos como aeropuertos, edificios gubernamentales…), sensores ambientales (temperatura, humedad…) e industriales (seguimiento de vehículos, control de flota…), en la automoción, en el campo de la medicina (pulsaciones, presión arterial…), domótica (hogares/edificios inteligentes…), entre otros.

Dado que, como se ha visto, es una tecnología que está evolucionando de manera exponencial, y a altas velocidades, se hizo inevitable la asociación de otras tecnologías que trabajaran a la par con las WSNs (como la integración móvil, anteriores a éstas: ordenadores personales PC como acceso remoto a las WSNs), para lograr así, el intercambio de datos, o la difusión de manera eficaz, de la información obtenida por dichos sensores, a un usuario final, y de este modo, trabajar con los datos obtenidos, consiguiendo un gran avance e interacción entre el entorno y ser humano.

En cuanto a redes de sensores inalámbricas, la integración móvil ha sido, sin duda, uno de los principales avances a la hora de ofrecer al usuario final acceso ubicuo a los datos del sensor en una WSN.

La incorporación de la tecnología Bluetooth, facilita la adaptabilidad, en cuanto a integración móvil se refiere. Explotando el hecho de que la mayoría de los teléfonos móviles poseen tecnología Bluetooth, y que las redes de sensores inalámbricas es un hecho que cada día proliferan en diferentes ámbitos de aplicación, surge la necesidad de iniciar el acceso remoto a dichas redes, incorporando como solución al acceso remoto: la integración móvil.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Como tecnología emergente, surgen diferentes proyectos relacionados, los cuales se corresponden con la utilización de terminales móviles para el acceso remoto, algunos incorporando mejoras a las tecnologías existentes, y otros enfocados a ser una innovación en los avances tecnológicos.

En la actualidad, se ha desarrollado hacia diferentes vertientes, una de ellas: la integración móvil en redes de sensores inalámbrica, a grandes rasgos nombraremos algunos de los trabajos realizados que competen con dicha integración y que hacen posible la proliferación de las mismas en las WSNs:

"Mobile Enabled Large Scale Wireless Sensor Networks" [1] Incorpora la tecnología móvil para redes de sensores inalámbrica, ofreciendo una creciente mejora en redes a gran escala. Introduce una nueva arquitectura de red para redes de sensores inalámbrica "multi-radio and mobile enabled wireless sensor network " (MEMOSEN), agregando terminales móviles a las redes de sensores inalámbricas; obteniendo de este modo un tipo de red híbrida de sensores inalámbricos, en la que los terminales móviles actúan como un sensor.

"A Heterogeneous Sensor Network Architecture for Highly Mobile Users"[2] Combina las redes "Personal Area Networks" (PAN) con las tecnología que presenta la WSN, logrando así llevar datos del sensor en una red corporal "Body Area Network" (BAN) a Internet. Presenta resultados para una red inalámbrica de sensores pequeña, enfocada en abordar el seguimiento a personas con gran movilidad en entornos de trabajos peligrosos, como es el caso de bomberos y cuerpo policial.

"Open Wireless Sensor Network Telemetry Platform for Mobile Phones" [3] desarrollado para crear un sistema de open-source, que crea un vínculo por el cual, los datos son recolectados por el usuario mediante un terminal móvil, a través de un Gateway, y enviados para su posterior procesamiento. Dichos datos se sobre-escriben en cada solicitud que realice el usuario a la WSN. La tecnología utilizada

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

entre la WSN y el terminal móvil es Bluetooth, esta vez, se ha creado un Gateway previsto de Bluetooth para conectar redes de sensores con terminales móviles a baja velocidad. Los terminales móviles son un recurso que permiten la conexión de WSN (de bajo consumo de energía) a Internet.

Varios sensores espaciados en una zona en concreto, recolectan datos que son almacenados por el Gateway, y éste a su vez, cuando le sea solicitado por el usuario, transfiere estos datos al terminal móvil. En nuestro caso, la interacción permite conectar con un nodo en concreto de la red de WSN, y obtener los datos que el usuario desee a través del Gateway.

La incompatibilidad de protocolos entre las tecnologías móviles y las WSNs ha creado una barrera a la hora de facilitar el acceso a las redes de sensores inalámbricas.

Este proyecto proporciona la conectividad Bluetooth a mayor velocidad, obteniendo los datos de la WSN sin necesidad de utilizar la red de Internet, y variando el enfoque de crear un subconjunto para descarga de datos de una WSN, a uno que permita el acceso extendido de usuarios con terminales móviles, a cualquier nodo de la WSN.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

A.3. Capítulo 3: Arquitectura del Sistema de componentes

En este capítulo se muestran las principales características de los dispositivos, que serán utilizados, teniendo en cuenta una serie de requisitos para lograr su uso eficiente.

Nodos de sensores se basan en la tecnología de radio y red Zigbee en el estándar IEEE 802.15.4, que está optimizado para la comunicación eficiente de la energía a bajas velocidades.

Para acceder a los datos de la red de sensores se ha diseñado un dispositivo Gateway, que implementa una tarjeta de red con interfaz de radio Bluetooth, para facilitar el acceso a los usuarios finales, a la red de sensores.

Nodo sensor: basados en la plataforma Sun SPOT .Sun SPOT consta de dos placas: Procesador (eSpot) y el sensor de placa (eDemo), cada uno desarrolla diferentes tareas.

Módulo Bluetooth: Bluetooth es un enlace de comunicación inalámbrica, que operan en la banda ISM sin licencia de 2,4 GHz con un salto de frecuencia del transceptor. Permite comunicaciones en tiempo real de datos entre hosts Bluetooth. El protocolo de enlace se basa en intervalos de tiempo.

La placa diseñada representa el resultado final de la placa de circuito impreso, a la que luego se añadió el módulo Bluetooth, realizando los puntos de soldadura de interés.

En este capítulo también se hace referencia a la pila de protocolos tanto del módulo Bluetooth como de la plataforma utilizada Sun SPOT.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

A.4. Capítulo 4: Diseño e implementación de la plataforma Hardware

Configuración Hardware

Gateway

Para el diseño del Gateway, se ha utilizado la plataforma Sun SPOT y el módulo de Bluetooth RN-41, con las especificaciones mencionadas en el capítulo anterior.

El software utilizado como herramienta de diseño, es EAGLE (Easily Applicable Graphical Layout Editor).

EAGLE es una herramientas de fácil uso para el diseño de circuitos impresos.

Para poder realizar el diseño del circuito impreso y adaptar el módulo Bluetooth RN-41 (en adelante BlueSMiRF), se tuvieron en cuenta los esquemáticos pertenecientes a los dispositivos implicados: Sun SPOT, proporcionados por Sun MicroSystem, en adelante 'eSPOT', más específicamente los representados en el plano 'Micro / Memory'; así como los del módulo BlueSMiRF.

Para permitir la conexión entre placas se ha diseñado una librería personalizada, correspondiente al conector utilizado DF17- 30 HDR (HEADER). Dicho conector es de la Serie DF-17 de 30 contactos.

Por otro lado la placa del eSPOT, incorpora el conector DF17- 30 R (RECEPT), pudiendo así adaptar nuestro diseño.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Módulo Bluetooth RN-41:

Para poder establecer la comunicación hardware entre el módulo BlueSMiRF y la eSPOT, el conexionado se hizo de la siguiente manera:

- BlueSMiRF Radio TX → RX eSPOT (ARM920T-based Microcontroller). UART_RX: UART receiver input.
- BlueSMiRF Radio RX ← TX eSPOT (ARM920T-based Microcontroller). UART_TX: UART transmitter output.
- BlueSMiRF Radio RTS y Radio CTS no conectados, ya que entre el módulo Bluetooth y la plataforma Sun SPOT no tendremos en cuenta el Control de Flujo, por lo tanto estará desactivado.
- BlueSMiRF VDD 3.3 V (regulated power input) VCC eSPOT.
- BlueSMiRF GND GND eSPOT.

Estado de la conexión. El módulo incorpora dos LEDs, uno de ellos parpadea a diferentes velocidades indicando el estado, el otro refleja directamente el estado de la conexión, que pasa a estado ALTO en el momento en que se conecta y BAJO en caso contrario.

Configuración Software

Para la configuración del Gateway, basado en la plataforma Sun SPOT, se ha creado un algoritmo, que entre otras tareas, incluye las siguientes configuraciones, para hacer posible la comunicación entre el modulo Bluetooth y la placa eSPOT.

La comunicación entre la placa y el módulo BlueSMiRF, se realizan a través de las conexiones USART.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

USART Connections:

Universal Asynchronous Receiver-Transmitter, realiza la conversión de datos de puerto serie a formato paralelo, transmitiendo los bits de manera secuencial.

Para la configuración del módulo Bluetooth se utilizó los AT Comands, los cuales nos permiten definir los requerimientos mínimos para permitir al módulo ser compatible con otros dispositivos Bluetooth.

La guía de la plataforma Sun SPOT en su apartado "Using input and output streams over the USB and USART connections", especifica todo lo que tenemos que tener en cuenta, tanto para abrir y cerrar las conexiones UART como las limitaciones que tenemos que tener en cuenta y que inevitablemente introducen la plataforma.

A continuación, se hará un breve bosquejo de cómo se han realizado estas conexiones

El acceso a las conexiones USART no los proporcionará el conector adaptado en la parte superior de la eSPOT, a él conectaremos nuestra placa de diseño, que a su vez, adapta el módulo Bluetooth y utilizando dicha conexión para hacer posible la comunicación entre ellos.

La conexión USART se abre del mismo modo que la USB. Esta función se incorporará sólo una vez, al inicio del programa.

Los parámetros implicados para realizar las conexiones entrada/salida son 'input' y 'output', especificando en 'serial' el tipo de conexión que queremos utilizar tanto para si es USART como si la conexión que queremos es USB. En nuestro caso y en adelante nos referiremos a la conexión USART.

```
InputStream input = Connector.openInputStream("serial://usart");
```


© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

```
OutputStream output = Connector.openOutputStream("serial://usart");
```

Para leer los datos de entrada se pueden implementar diferentes subrutinas, en definitiva la lectura se realiza de la siguiente manera:

```
byte b[] = new byte[0x0011]; // definimos una variable de tipo byte donde queremos almacenar los datos de entrada.

input.read(b); // leemos los datos de entrada y lo almacenamos en la variable anterior.

String s = new String(b, 0, b.length); // Convertimos estos datos leídos de bytes a una cadena de caracteres, logrando así el proceso inverso al envío.
```

En consecuencia, para enviar datos por la 'output' utilizamos la siguiente línea:

```
output.write("String".getBytes());
```

'String', representa la cadena de caracteres que se quiere enviar.

'getBytes()' es una función propia, que permite transformar cadenas de caracteres en bytes y de este modo ser enviados por la Stream de salida.

Limitaciones en el uso de USART en la plataforma Sun SPOT:

Hay que tener en cuenta, que en el reinicio del Sun SPOT, se envían algunos caracteres a través de la conexión USART, correspondientes con la configuración, afectando así a los datos recibidos. Éstos caracteres, son necesarios para permitir la comunicación entre la aplicación SPOT y el SPOT cliente.

Para evitar este problema, introduciremos a nuestras subrutinas de lectura de datos de entrada, una nueva tarea, que se encargará de limpiar el buffer de entrada, justo después del reinicio del dispositivo y antes de recibir los datos de interés.

```
long i = 0;
```


© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

```
while ((i = input.available()) > 0)
    input.skip(i);
```

'input' representa la conexión abierta de entrada

'i' la longitud de caracteres disponibles en ese momento en la entrada.

La función 'skip' permitirá obviar los datos que hasta ese momento se tengan en el buffer de entrada.

"USART input uses a 512 byte input ring buffer", lo que quiere decir que puede haber desbordamiento del buffer, en consecuencia, depende de la aplicación que se vaya a implementar, para controlar que los datos se lean con suficiente rapidez, con el fin de no perder datos, sabiendo la velocidad de transmisión y tasa de transmisión.

Una de las dificultades encontradas en la implementación para la recepción de datos, se relaciona con ésta limitación. Para poder leer los datos enviados del módulo Bluetooth al Sun SPOT, sin perderlos, hubo que insertar un pequeño retraso('sleep time') entre la lectura y escritura, poco significativo para ejecución final del programa, pero lo suficiente para no perder datos recibidos a través de la USART.

Para una recepción fiable se aconseja utilizar la máxima velocidad de transmisión compatible. En nuestro caso, el módulo Bluetooth se ha configurado a la máxima velocidad de transmisión, por lo que aseguramos una mejor recepción de datos.

Por otro lado, el Gateway establece comunicación con el resto de dispositivos, basados en la misma plataforma (SunSPOT) y que conforman la WSN. Dicha comunicación se establece a través del Protocolo de Radiogramas basado en datagramas.

Radiogram Protocol:

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Permite el intercambio de paquetes entre dos puntos, el protocolo de la definición de 'radiograma' se basa en datagramas.

Para poder realizar dicha conexión es necesaria la asignación de los mismos puertos en ambos extremos, además de la dirección IEEE, lo que implicará que la comunicación se establecerá por dichos puertos y con las direcciones IEEE correspondientes.

En una primera instancia utilizaremos un 'Broadcast_Port' para localizar la aplicación host. Este Puerto permanecerá a la 'escucha' de las solicitudes 'clientes'.

La manera de abrir las conexiones basadas en el protocolo Radiogram, se realiza lo siguiente:

rcvConn = (RadiogramConnection) Connector.open("radiogram://:" + BROADCAST_PORT);

A continuación el Puerto de tipo 'Connected_Port' se utilizará para el envío de comandos y respuestas entre el punto remoto y la aplicación host. Dado que RadigramConnection soporta el modo de difusión, la conexión saliente se realizará por un puerto especificado (Connected_Port) pero enviándose en modo de difusión, lo que significa que llegará a todos los 'clientes' dispuestos a conectar con la aplicación host, a través del puerto determinado.

txConn = (DatagramConnection) Connector.open("radiogram://broadcast:" + CONNECTED_PORT);

Los puertos numerados del 0 al 30 están reservados para servicios del sistema, por lo tanto, y con el fin de evitar conflicto en la aplicación, los puertos utilizados son: 42 (Host Name Server Protocol); y 43 (WHOIS), is a query and response protocol ampliamente utilizado para la consulta en la bases de datos que almacenan

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

los usuarios registrados.

Para establecer la conexión especificamos el puerto al que nos queremos conectar y proporcionamos la dirección IEEE del extremo con el que queremos conectar.

Una vez establecida dicha conexión, el intercambio de paquetes, se basa en los datagramas creados, tanto para enviar como recibir.

A.5. Capítulo 5: Conclusiones

A pesar de las dificultades encontradas a lo largo del desarrollo del proyecto, tanto a nivel de software (una de ellas: la sincronización y tiempos de espera, entre paquetes recibidos y enviados, por comunicación UART) como a nivel de hardware (como fue el caso de crear una PCB para poder adaptar el modulo Bluetooth a la plataforma de SunSPOT, esto es porque la eDemo, solo permite comunicaciones a través de UART a baja velocidades y teniendo en cuenta que mayor velocidad implica menor tiempo de espera, sería "obsoleta" la implementación en la placa eDemo).

La finalidad de este proyecto ha sido crear una plataforma basada en Sun SPOT, la cual pueda comunicarse con una red de sensores y hacer accesible ésta información a los usuarios móviles.

Para poder hacer posible la comunicación móvil, fue necesario del diseño de una PCB a la cual se pudiese adaptar el modulo de Bluetooth y por otro lado conectar a la Processor Board (eSPOT) de nuestra plataforma Sun SPOT.

El Gateway se ha configurado para obtener datos de los nodos de una red de

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

sensores, la aplicación permite interactuar con el dispositivo y obtener los datos de un nodo en particular, encontrado en una WSN, ha sido desarrollada en primera instancia para el ordenador, pudiendo obtener datos del sensor de dicho nodo, tales como: valor de luz; valor de temperatura; datos del acelerómetro, a través del cual se ha generado una gráfica que presenta la movilidad en los tres ejes del sensor del que se obtienen los datos; así como enviar tareas sencillas, en este caso, se ha realizado un parpadeo de LEDs (4 veces) para comprobar una vez mas la conectividad con el dispositivo remoto.

A su vez la aplicación implementada en el ordenador, permite cerrar la conexión para conectar con otro nodo en remoto, al mismo tiempo, permanece a la "escucha" de nuevos nodos disponibles para conectar, ésta escucha se trata de obtener a través del Gateway las direcciones MAC de cada dispositivo encontrado, y a través de la cual logramos establecer la conexión.

Por otro lado, en los nodos, se ejecuta una aplicación sencilla, que ha permitido comprobar el funcionamiento, se trata de registrar los paquetes que se le solicitan desde el Gateway, sabiendo así en cada situación la respuesta que tiene que dar, obteniendo de este modo lo deseado por la aplicación final. El nodo, permanece "disponible" para conectar, una vez establecida la conexión el intercambio de información en ambos sentidos (aplicación - nodo; nodo - aplicación) pasa por el Gateway, es decir, no hay comunicación directa entre nodos de la WSN y la aplicación usuario final, la interacción se ha registrado en nuestro Gateway.

El Gateway, en una situación real, se comporta como un servidor, permaneciendo atento de nuevas conexiones provenientes de modulo Bluetooth para iniciar la conexión con el nodo remoto. Esto permite la accesibilidad de cualquier usuario final, mediante la aplicación, a un nodo de la WSN.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Como sistema innovador, ha aportado una nueva vertiente de aplicación en el campo de las WSNs, proporcionando un aplicación de ejemplo sencillo, así como una implementación de bajo coste y adaptable a las necesidades de usuario.

Aunque la aplicación ha sido desarrollada como ejemplo sencillo de demostración, el proyecto proporciona las herramientas necesarias para introducir mejoras, pudiendo así aportar un avance en el crecimiento de las WSNs como tecnología punta.

Para terminar, el desarrollo de ambos: tanto aplicación como del prototipo, permiten ser un punto de partida de implementación de un Gateway, basado en comunicación Bluetooth, para permitir el acceso extendido a usuarios, y en comunicación vía radio para acceder a los nodos de una red WSN. Añadiendo a su vez, características como bajo consumo y bajo coste.

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

Bibliography

[1] Frank H. P. Fitzek, Frank Reichert. "Mobile phone programming and its application to wireless networking", 2007

[2] Andrea Goldsmith. "Wireless Communication", 2005

[3] F.L. Lewis. "Wireless Sensor Network", 2004

[4] Sergio R. Caprile. "Esquibí: Desarrollo de aplicaciones con comunicaciones remota basadas en módulos Zigbee y 802.15.4", 2009

[5] Jan Axelson. "USB complete: everything you need to develop custom USB peripherals", (2005- third edition)

[6] Zigbee Alliance. Available: www.zigbee.org

[7] Römer, Kay; Friedemann Mattern. "The Design Space of Wireless Sensor Networks", 2004

[8] Holger Karl; Andreas Willig. "Protocols and architectures for Wireless Sensor Networks", 2005

[9] Jun Zheng (Ph. D.), Jun Zheng, Abbas Jamalipour. "Wireless Sensor Networks: A networking perspective", 2009.

[10] Drew Gislason. "Zigbee Wireless Networking", 2008

[11] Albert S. Huang; Larry Rudolph. "Bluetooth Essentials for programmers", 2007

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

[12] Android. Available: http://www.android.com/
(Android Developer. Available: http://developer.android.com/index.html)

[13] Open Handset Alliance. Available:
http://www.openhandsetalliance.com/android_overview.html

[14] Canfeng Chen, Jian Ma, "Mobile Enabled Large Scale Wireless Sensor Networks", IEEE Conferences: Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, vol.1, pp. 333 - 338, 2006.

[15] Eliasson, J.; Chen Zhong; Delsing, J., "A heterogeneous sensor network architecture for highly mobile users", IEEE Conferences: Wireless Communication and Sensor Networks (WCSN), 2010 Sixth International Conference on, pp. 1-6, 2010

[16] Harnett, C.K.,"Open wireless sensor network telemetry platform for mobile phones", IEEE Sensors J., vol. 10, n

[17] Sun SPOT: SunSPOT world. Available: http://www.sunspotworld.com

[18] ATMEL. Documents specifications and characteristics. ATMEL (Rev. 1768I)-2009. Available: <www.atmel.com/dyn/resources/prod_documents/doc1768.pdf>

[19] Bluetooth V1.1 Core Specifications. Available:
https://www.bluetooth.org/Technical/Specifications/adopted.htm

[20] Bluetooth product. Available: http://www.sparkfun.com/products/9358

[21] Bluetooth Protocol Stack. Available:
http://developers.sun.com/mobility/apis/articles/bluetoothintro/index.html

[22] JAVA. Available: http://www.java.com/en/about/

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

[23] SUN ORACLE. Sun Labs. "Sun SPOT Programmer's Manual"- Release v6.0 (Yellow), November-2010

[24] Bluetooth Data Sheet. Roving Networks. "Class 1 Bluetooth® Module - DS-RN41-V3.1", 2009. Available: <http://www.rovingnetworks.com/documents/RN-41.pdf>

[25] Bluetooth Command Set. "Roving Networks Bluetooth™ Product User Manual". Version: 4.77, 2009. Available: <http://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-bluetooth-um.pdf>

[26] Sun Microsystem. Sun Labs. Sun SPOT "Theory of Operation" Red Release 5.0, June-2009. Available: <http://sunspotworld.com/docs/Yellow/SunSPOT-TheoryOfOperation.pdf>

[27] HIROSE Connector. Available: http://uk.mouser.com/ProductDetail/Hirose-Connector/DF1740-30DP-05V57/?qs=sGAEpiMZZMvlX3nhDDO4ACnyrefzrFf8lwLQmiEUgcU%3D

[28] National Instruments. Tutorial: Wireless Sensor Network, 2009. Available: http://zone.ni.com/devzone/cda/tut/p/id/8707

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013