Universitá degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria dell'Informazione

A Java application to display temperature, humidity and luminosity in a Wireless Sensor Network

Relatore:
Leonardo Badia

Laureanda:
Estefanía Recayte

Correlatore:
Moreno Dissegna

Anno accademico 2011/2012

con cariño:
a mi lindo ahijado,
a mi preciosa familia...

CONTENTS

Abstract

The aim of this thesis is to develop a Java application, using Android platform, to display in a mobile device the values taken by a Wireless Sensor Network (WSN). For this purpose, in this papers, firstly I give a general vision of what the Wireless Sensor Networks are. In particular, I wished to provide a comprehensive analysis of the main component of the WSN: the sensor node. I describe its hardware and software structural. The second part of the thesis is dedicated to the implementation of the application. It is listed which software were useful to the development and it explained how the application has been built. In the last part of the thesis it is shown the graphical result of the work obtained.

CHAPTER ONE

INTRODUCTION

During the recent years we have assisted to important advances in the networking field thanks to the exponential growth of technology. This is confirmed by the law of Moore: the number of transistors on integrated circuits doubles approximately every two years [1]. Thus, rapid advances in the areas of sensor design, inexpensive and low-power electric circuits, information technologies, and wireless networks have paved the way for the proliferation of so called: ”Wireless Sensor Networks” (WSN) which promise to revolutionize the way we live, work and interact with the physical environment.[2]

These networks assure the potential to interface the physical world with the virtual (computing) world on an unprecedented scale and provide practical usefulness in developing a large number of applications, including the protection of civil infrastructures, automation systems, industrial application, avionics, domotics and security, energy meter, greenhouses, workhouses, environment monitoring and an endless of new application [3]. However, the design of wireless sensor networks introduces formidable challenges, since the required body of knowledge encompasses a whole range of topics in the eld of electrical and computer engineering, as well as computer science.[3] Wireless sensor networks are currently being offered as a subject at advanced undergraduate and graduate levels at many universities around the world and has been recognized such as the most promise technology by different technology's analyst and qualified magazine. Technology Review, the magazine published by MIT, has identified this method to communicate as one of the technology that will change the world [5].

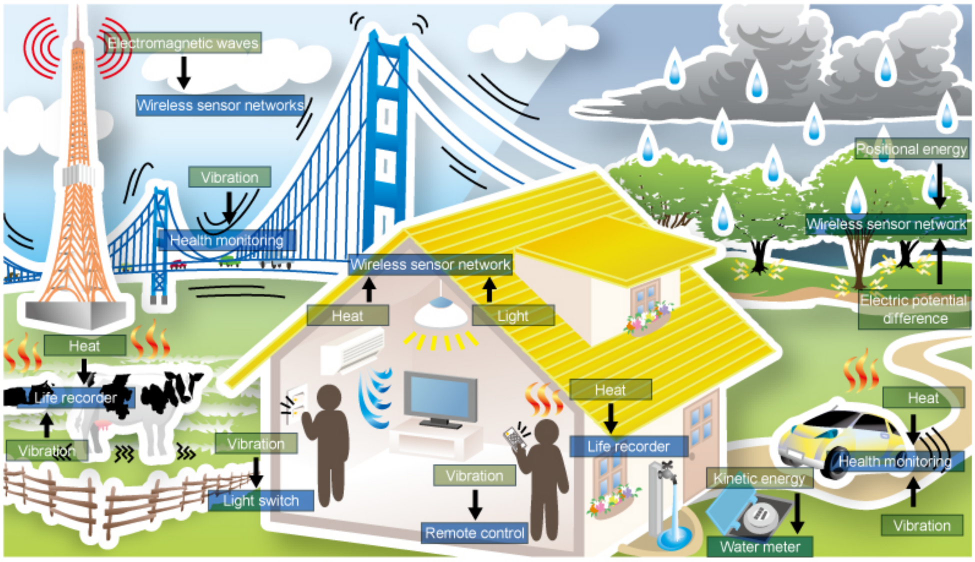


Figure 1.1: Possible applications of WSN

WSNs have been actively researched since the late 1990s and they are still in research and development stage. Over this time many groups of engineers from a different universities have implemented several testbeds and experimented with different kinds of deployment scenarios. At the beginning, the first deployments were test networks with few devices and a limited lifetime. A listing and short analysis of 50 environmental monitoring is given by Hart and Martinez [6]

The University of California at Berkeley planned several projects in a variety of sectors. The first example to mention is the implementation of Wireless Sensor Network for Structural Health Monitoring (SHM) [7]. It allows the estimation of the structural state (regarding civil works) and detection of their changes that might affect the performance. The chosen test bed is the Golden Gate Bridge in San Francisco Bay.

Another project realized is Wireless Sensor Networks for Habitat Monitoring [8]. The objective of project is the monitoring sea-bird nesting environment and behaviour is presented. In this case the test bed chosen is small island off the coast of Maine.

In Italy a team of specialist engineers from University of Padova was working in one of the largest WSN in Europe called "Wireless Sensor Network for city-Wide Ambient Intelligence" [9]. The aim of the project is to demonstrate the feasibility of large-scale, composed by around 350 sen-

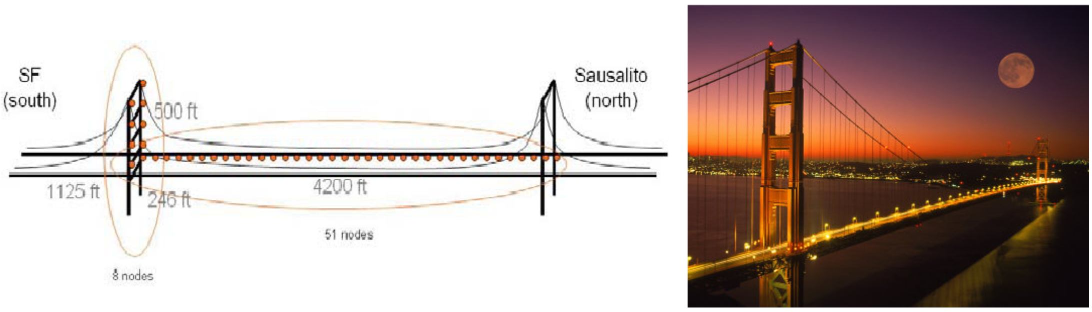


Figure 1.2: Wireless sensor Nodes in the Golden Gate Bridge

sors node, whereby tiny objects integrating one or more environmental sensors (temperature, humidity, luminosity), a microcontroller and a wireless transceiver are deployed over a large area, which in this case involves the buildings of the Department of Information Engineering (DEI) at Padova.[10]

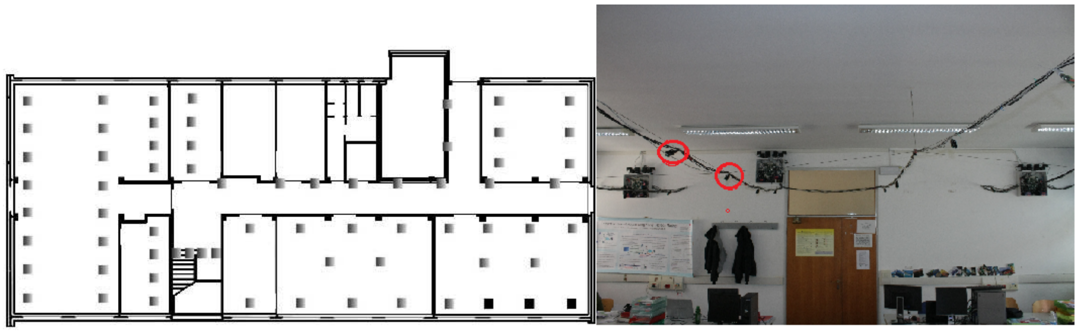


Figure 1.3: Wireless sensor Nodes at DEI

Further projects to mention include researchers from the University of Southampton have built a glacial environment monitoring system using WSNs in Norway [11]. They collect data from sensor nodes installed within the ice and the sub-glacial sediment without the use of wires which could disturb the environment. PODS (*Programmable Oceanic Devices*) in Hawaii monitor the weather conditions and rare plants [12]. CORIE (*Columbia River Ecosystem*) measures the temperature and pressure at stations located in the Columbia River [13]. NIMS (*Networked Infomechanical System*) monitors the solar radiation in forest environments [14].

Thus, technology is becoming extremely useful and nowadays one of the most important social needs is to have available information. We could collect a large amount of data to process every single action we take in our everyday routines. We are assisting a technology revolution: sensors, machines and computers are beginning to communicate each other wirelessly in ways that could dwarf traffic between familiar devices like smathphones. Recently research done by ONWorld (a business incorporation of research and specialize on smart technology) talks about spread of smart WSNs proves that over the past two year their market has doubled [15]. The article confirms that the installed wireless industrial field devices will increase by 553 percent between 2011 and 2016 when there will be nearly 24 million wireless-enabled sensors and actuators, or sensing points, deployed worldwide.

This thesis is structured in five chapters. The second chapter "Wireless sensor Networks" gives a general notion of WSNs, their application and requirements and analyses in detail the most important element that compose a WSN: the sensor node. The third chapter "Application Implementation" describes the application, the software requirement for the implementation and it is explained how the work has been realized. The fourth chapter "Application Performance" shows the graphical result obtained of the application. Finally, the fifth chapter draws the conclusions of the thesis.

CHAPTER

TWO

WIRELESS SENSOR NETWORKS

2.1 Wireless Sensor Networks

Wireless Sensor Networks consist of a large number of devices spread throughout a given area. These devices called sensors operate autonomously and are equipped with the capability of sensing the physical environment, data processing, and communicating wirelessly with other sensors. WSNs are a very specific class of the so-called ad hoc networks [10]. Ad hoc network are characterised by the fact that devices are self-organized and are able to communicate without a centralized base station. WSN differs from a ad hoc network because a sensor network is made up of tinies devices which are capable to sense and take measures from the surrounding environment.

In a wireless ad hoc network, a group of wireless nodes spontaneously form a network without any fixed and centralized infrastructure. Whenever two nodes want to communicate, intermediate nodes are called upon to forward packets and form a multi-hop wireless route. Due to possibilities of node mobility, the topology is dynamic and routing protocols are proposed to search for end-to-end paths [16]. The network nodes rely on peers for all or most of the services needed and for basic needs of communications. Due to the absence of centralized control and management, nodes rely on fully distributed and self-organizing protocols to coordinate their activities. Distributed protocols need to accommodate dynamic changes at any given time: a node may join or leave the network arbitrarily, links may be broken and nodes may be powered down as a result of node failures or intentional user actions [17].

If the WSN wants to communicate with another network (e.g. through the Internet) exists a special node call sink node which typically is connected to others networks through a gateway.

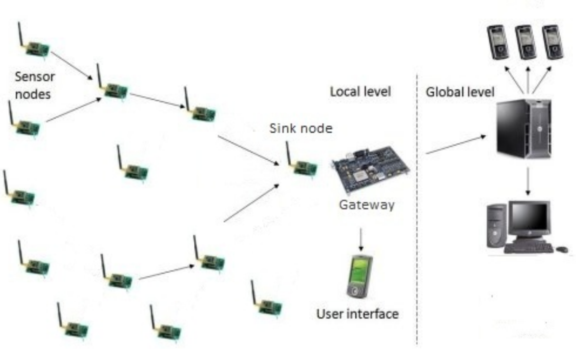


Figure 2.1: A sample WSN scenario

2.2 Sensor Node

The sensor node is capable of performing some processing, gathering sensed information and communicating with other nodes in the network. A typical structure of a sensor hardware(see Figure 2.2) consists of a microprocessor, a memory , analogyc sensors (transductor amd actuator), analog to digital converters (ADCs), a data transceiver and a power supply. Recent implementation of wireless sensors also incorporate a serial connection (e.g. though a USB interface) which is used for programming but can also be employed for power supply.

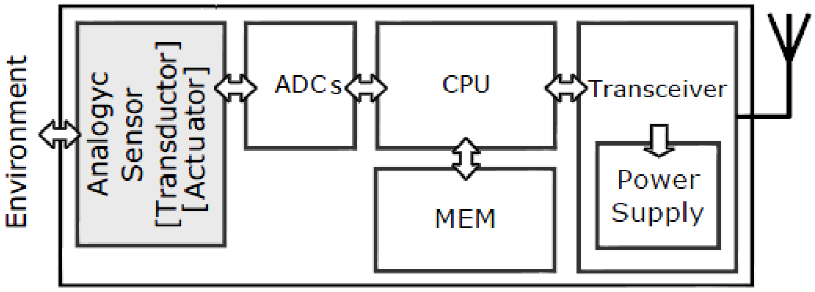


Figure 2.2: Architecture of a sensor node
[18]

2.2.1 Hardware

The **transductor** is device able to measure physical phenomena or environment conditions. It transforms this data in an electric signal. Transductor for instance could be analog sensors of temperature, humidity, luminosity, pressure, as well as smoke detectors or toxic substance detectors. [19] Instead, actuators are device able to interact with the environment in different ways. Examples of actuators are mechanic arms, valves, sprinklers but also alarms, loudspeakers, video projectors. A sensor node can include several transductors and actuators which is what determinates its the cost, functionality and the energy the overall network.

The **Central Processing Unit** provides the necessary *intelligence* to the device for operate autonomously. Different topologies of logic circuit can behave as CPU. Commonly a microcontroller with reduced consumption is chosen. It is also possible to incorporate an integrated circuit such as Field Programmable Gate Array (FPGA) or a Digital Signal Processing (DSP) or an Application Specific Integrated Circuit (ASIC) to reduce the computation times.

The **Analog to Digital Converter** ADC translate the continuous electric quantities (a voltage) received from the transductor to a discrete representation in digital form. Analogously the node often includes a Digital to Analog Converter (DAC) which performs the reverse operation, transforming numerical signals from the microcrontroller to electric signals to pilot actuators.

The **memory unit** is useful to store the data read and to store the application and the operating system.However, low-power microprocessors have limited storage, typically less than 10 kbytes of RAM ($RandomAccessMemory$) for data and less than 100 kbytes of ROM (*ReadOnlyMemory*) for program storage. Designers typically incorporate larger amounts of flash storage, perhaps a megabyte, on a separate chip [19].

The **transceiver** has the function to communicate with others nodes. The communications between sensors nodes are via radio. WSN radios consume about 20 mW and the amount of energy increases rapidly with distance. More energy is required for data communication than any other process.

The **power supply** is a crucial system component, sensor nodes operate with limited energy budgets and the energy consumption of the sensor node must to be controlled.

The possible methods to provide power to the node can be classify in three groups: store energy (i.e., batteries), distribute power to the node (i.e., a wire) and scavenge available ambient power at the node (i.e., a solar cell). It is also possible a combination of this methods [21].

In a WSN it is very important to reduce the power consumed above if we want the a longest lifetime of devices. Thus, it is possible to reduce the power consumed with developing design methodologies and architectures which help in energy aware design of sensor networks. The lifetime of a sensor network can be increased significantly if the operating system are designed to be energy aware. Power management in radios is very important because radio communications consumes a lot of energy during operation of the system. Another aspect of sensor nodes is that a sensor node also acts a router and a majority of the packets which the sensor receives are meant to be forwarded [20]. Therefore, using an efficient routing it possible balance the energy consumption between nodes.

Traffic can also be distributed in such a way as to maximize the life of the network. It is better that the load of the traffic be distributed more uniformly throughout the network.[20]

Devices are designed to avoid excessive energy consumption to minimise the average current drawn. Thus, there are three state wherein a node can be put: sleep state, operational state and idle state:

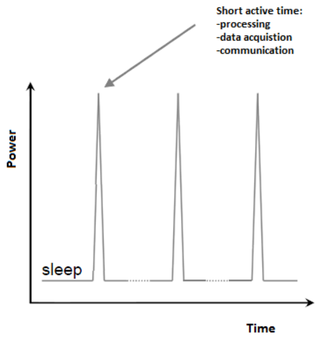


Figure 2.3: States of Sensor Node

- **Operational state:** consists in transmit state and receive state. The time that the device spends in this state is very short: milliseconds or less.

- **Idle state:** A transceiver that is ready to receive but is not currently receiving anything is said to be in an idle state. In this idle state, many parts of the receive circuitry are active, and others can be switched off.

- **Sleep state:** significant parts of the transceiver are switched off. These sleep states differ in the amount of circuitry switched off and in the associated recovery times and start up energy.

2.2.2 Software

In addition to hardware platforms, several software platforms have also been developed specifically for WSNs. Among these, the most well known and diffused platform is the TinyOS [22] which is specially designed for resource constrained low power, low memory processors. TynyOS, as the definition of its website, is an open-source operating system developed by Berkeley University for wireless embedded sensor networks. TinyOS incorporates a component based architecture, which minimizes the code size and provides a flexible platform for implementing new communication protocols. Its component library includes network protocols, distributed services, sensor drivers and data acquisition tools, which can be further refined for a specific application requirements. TinyOS is based on an event driven execution model that enables fine-grained power management strategies. Most of the existing software code for communication protocols today is written for the TinyOS platform and a wide community uses TinyOS in simulation to develop and test own algorithms and protocols.

Commonly, the applications to interface with the user are developed in C programming and be compiled from the specific microcontroller of the node sensor used.

Example of sensor node

TelosB and Tmote sky are example of sensor node, both were developed by University of California. They were developed to have a low power consumption, easy to use and to increase the hardware and software robustness.

TelosB is the most sophisticated versions of the COTS *(Commercial Off-the-Shelf component)* motes [23]. It has low power consumption, better radio range, radio interface with better noise immunity and various power down modes.

TelosB has the feature that it can be in sleep state for the majority of the time and wake up very fast to operate and returns to the sleep state

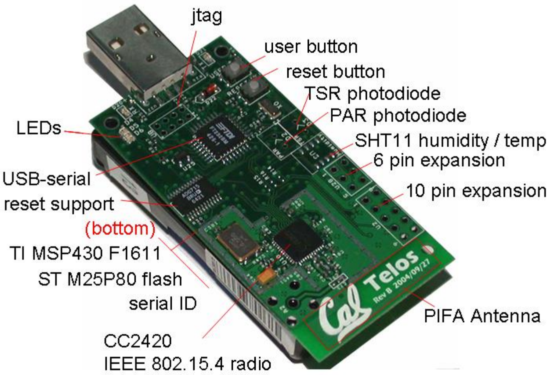


Figure 2.4: TelosB

once the process is completed. Hence, the power consumption is minimal.

TelosB is powered by two AA batteries and has also a USB port. If the platform is connected to the USB port for programming or communication, power is given from the host computer. If the platform is always connected to the USB interface, no battery pack is required.

TelosB nodes use the CC2420 radio chip for communications in the 2.4 GHz band, in accordance to the IEEE 802.15.4 standard to exchange and use information in a seamless manner with other instruments.[24] Their maximum transmission power is 1mW within the 2400-2480 MHz bandwidth and an effective data rate of 250 kbps [10]

Other main features are: incorporated onboard antenna, low current usage, microcontroller of 8 MHz with 10 kb RAM, programming and information accumulation through USB, 1 MB outside flash for information logging, 12bit-ADC and 12bit-DAC, optional sensor suite including incorporated temperature (range: -40° C to 123.8° C), light (range: 320nm to 730nm) and humidity sensor (range: 0 to 100 %) [24].

TelosB is completely compatible with the open source TinyOS distribution. The energy efficient software developed by manufacturers to support big scale, self-assembling sensor networks.

2.2.3 Temperature and Humidity Sensor

A commonly sensor integrated on TelosB is SHT11 (see Figure) manufactured by Sensirion AG. A unique capacitive sensor element is used to measure relative humidity, while temperature is measured by a band-gap sensor. Both sensors are produced with a CMOS technology, are coupled to a 14bit A/D converter and a serial interface circuit and provide a fully calibrated digital output [26].

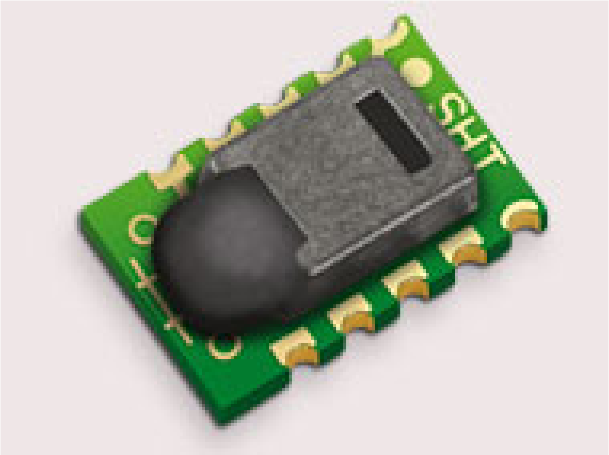


Figure 2.5: Sht11 humidity and temperature sensor

The capacitive sensor method consist of convert physical properties into electrical signals. For example, in the parallel plate model, that is a capacitor consisting of two parallel conductive plates separated by a dielectric. The dielectric for SHT11 is a polymer which absorbs or releases water proportional to the relative environmental humidity with a certain permittivity ϵ [3].

The capacitance is determined as:

$$C = \frac{\epsilon \cdot A}{d} \tag{2.1}$$

where A is the plate area and d is the distance between the two plates. Thus, if the permittivity of the dielectric changes, even the capacitance, and thereby the humidity changes.

2.3 Applications

Applications for WSNs may either be traditional ones, in which the wired cabling is replaced by WSN, or completely new. The most convenient implementation is where human observation is almost impossible, where wired cabling are difficult to install or the cost is excessive. Their rapid and easy deployment is due to the small size node and maintenance free operations. Further, the integration and local processing of data measuring several different qualities enable a completely new type of applications.

The physical quantities that can be measured are diverse. The most common measured qualities are temperature, humidity, pressure, acceleration (vibration), sound, light, infrared magnetic fields, radiation, location

(GPS), chemical composition, and mechanical stress. As a consequence of extensive possibilities, the envisioned applications for WSNs are manifold. The miniaturization of sensing devices allows the embedding of small WSN nodes as an unnoticeable part of the everyday life. The most typical application domains for WSNs are the following: [25] [34]

- **Home automation:** WSNs are key building blocks for smart homes. The central control of lighting, heating, ventilation and air conditioning, local and remote management of home appliances, motorized blinds and curtains are examples of usage possibilities [27].

- **Environmental monitoring:** Random deployments on large-scale areas make the monitoring of agriculture and wildlife easier. It can be used also for animal tracking, forest surveillance, flood detection, and weather forecasting. Further, WSNs may be used in catastrophe (e.g. wildfire, earthquake, tsunami) warning systems and in disaster relief [28].

- **Industrial Sensing:** WSNs have a potential for replacing traditional cabling in monitoring and control systems in factories. Sensor nodes can be deeply embedded into machines to monitor their health and to ensure safe operation [29].

- **Military:** In military applications, the rapid deployment of WSNs allows instant use of data. Usage scenarios of WSN data are wide-ranging in intelligence, include large-scale acoustic ocean surveillance systems, detection of submarines and targeting. Further, WSNs can be a part of the communication infrastructure [30].

- **Security:** Compared to existing wired alarm systems, WSNs allow easier deployment, adaptivity, and error robustness. The increasing demand for security and alarm system is evident in home, office, and other public environments such as airports and factories [31].

- **Traffic control:** WSNs allows the easy vehicles monitoring and control of traffic conditions especially at peak times. Temporary situations such as roadworks and accidents can be monitored. Further, the integration of monitoring can significantly improve the city traffic management and reduce the emission of carbon dioxide [32].

- **Health care:** WSNs can be used into a hospital building to track and monitor physiological data of patient and all medical resources. Body sensor networks, a kind of special sensor implanted for healthcare purpose can measure blood pressure, body temperature and electrocardiograph [33].

Independent of the application domain, the main functionalities of WSN applications are data gathering and processing. The type of gathered data and the nature of processing depend on the application. In spite of the diversity of applications and domains, four main tasks that are independent of application domain can be identified:

- **Monitoring:** Detect the value of a parameter in a given location or the coverage area of the network. Typically, the task is completed using periodic measurements.
- **Event Detection:** Detect the occurrence of events of interest and their parameters
- **Object classification:** Identify an object or event. In general, this task requires the combination of data from several sources and a collaborative processing to conclude the result.
- **Object tracking:** Trace the movements and position of a mobile object within the coverage area of the network.

In addition to the data gathering. WSNs may also implement control or actuating functionality. The control can be either integrated into the network or controlled by a central entity. The integrated control makes decisions based on the information available locally whereas the central control entity can exploit data available from the entire network. On the other hand, the integrated control operations are faster because of the lack of data routing delay. The control operations are performed either by integrated actuators though external Input/Output or by separate actor nodes.

2.4 Requirements for WSNs

The requirements for a deployed WSN depend of the application and the operating environment. Hence, it is not possible to establish a specific generalization of the requirements. Nonetheless, WSN applications possess several common characteristics, based on which general requirements for the node platforms, protocols, and applications can be defined [34][35].

- **Fault tolerance:** It may be that nodes fail or be blocked certain amount of time for hardware or software reasons (due to harsh environment, depletion of batteries, or external inference make networks prone to errors). The failure of a node results in disconnection from the network. Thus, WSNs must be robust against failures or individual nodes or even all nodes in a certain location [36].

- **Lifetime:** The network lifetime is a crucial issue in WSNs. The nodes are battery-powered or the energy is scavenged from the environment and their maintenance is difficult. Energy saving and load balancing must be taken into account in the design and implementation of WSN protocol platforms, protocols, and applications [37].

- **Scalability:** The number of nodes is WSN is typically high and it is important for to provide redundancy and to improve the fault tolerance of the network. While the scale partly depends on the covered area, the replication of nodes, limited sensing coverage, and applications requirements also call for dense deployments in small areas. Thus, the WSN protocols must deal with high densities and numbers of nodes [38].

- **Realtime:** WSNs are tightly related to the real world. Therefore, strict timing constraints for sensing, processing, and communication are present in WSNs. For example, on a realtime identification of an event, first the stimulations from sensors must be captured, the obtained data processed, and the event identified. Finally, the result should be passed instantly through the network [39].

- **Security:** The need for security in WSNs is evident, especially in heath care, security, and military applications. Most of the applications relay data that contain private or confidential information. In general, the security requirements for WSNs are difficult to fulfil due to complex and time consuming algorithms and the limited resources of nodes [40].

- **Production cost:** The number of nodes in WSNs is high, and once nodes run out of batteries they are replaced by new ones. Further, WSNs are envisioned to be everywhere. Therefore, to make the deployments possible, the nodes should be extremely how cost [41].

- **Programmability:** Not only will it be necessary for the nodes to process information, but also they will have to react flexibly on changes in their tasks. These nodes should be programmable, and their programming must be changeable during operation when new tasks become important.A fixed way of information processing is insufficient [42].

CHAPTER

THREE

APPLICATION IMPLEMENTATION

The main aim of the thesis is to design and implement an application which allows the user to select a sensor from all the deployed working sensors, in order to provide the environment measures (temperature, humidity, luminosity) and display the data read by using a mobile phone.

The rest of this section discusses the main contributions brought by this thesis, i.e.:

- the software utilized to implement the application
- the simulation network utilized
- the development of the user interface

3.1 Software Requirement

The development of the user friendly application exploited using the Android platform [43]. Android is an open-source software stack created for mobile phones and other devices and is built on top of Linux kernel and GNU software. The software stack of the Android runs Java applications using Java core libraries. Each instance of Java application runs on its own Virtual Machine(VM) called Dalvik [43].

Android is freely available to manufacturers for customization, there are no fixed hardware and software configurations [44]. However, Android itself supports features. The following listed (*from "Beginning Android Application Development", Wei Meng Lee*)[44] are some of them which used in the project:

- **Storage:** Uses SQLite, a lightweight relational database, for data storage. SQLite has been used to save the data read and images used by the application development [45].

- **Connectivity:** Supports GSM/EDGE (*Global System for Mobile Communications - Enhanced Data rates for GSM Evolution*) , IDEN (*Integrated Digital Enhanced Network*), CDMA (*Code Division Multiple Access*), EV-DO (*Evolution-Data Optimized*), UMTS (*Universal Mobile Telecommunications System*), Bluetooth, LTE (*Long Term Evolution*), WiMAX (*Worldwide Interoperability for Microwave Access*) and WiFi (e.g. standard IEEE 802.11 [46]) which have be used for this application.

- **Messaging:** Supports both SMS (*Short Message System*) and MMS (*Multimedia Messaging Service*).

- **Web browser:** Based on the open-source WebKit, together with Chromes V8 JavaScript engine

- **Media support:** includes support for the following media: H.263 , H.264 (in 3GP or MP4 container [49]),MPEG-4 SP (*Moving Picture Experts Group 4 Simple Profile*, AMR (*Adaptive Multi Rate*), AMR-WB (*Adaptive Multi Rate Wideband*) (in 3GP container), AAC (*Advanced Audio Coding*) , HE-AAC (*High-Efficiency Advanced Audio Coding*)(in MP4 or 3GP container), MP3 (*Moving Picture Expert Group-1/2 Audio Layer 3*) (e.g. file type: .mp3 [47], MIDI (*Musical Instrument Digital Interface*) (e.g. file type: Type 0 and 1 (.mid, .xmf, .mxmf) [48]), OggVorbis ,WAV (*WAVEform audio file format*), JPEG (*Joint Photographic Experts Group*), PNG (*Portable Network Graphics*), GIF (*Graphics Interchange Format*), and BMP (*bitmap image file*).

- **Hardware support:** Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor, and GPS (*Global Positioning System*).

- **Multi touch:** Supports multi-touch screens

- **Multi tasking:** Supports multi-tasking applications.

- **Flash support:** Android 2.3 and upper version support Flash 10.1

- **Tethering:** Supports sharing of Internet connections as a wired/wireless hotspot.

The software required by any Android application are free and readily available on the Web:

- **Eclipse:** [50] is the integrated development environment(IDE) and is a multi-language software development environment featuring an extensible plug-in system. It can be used to develop various types of applications, using languages such as Java, Ada, C, C++, COBOL, Python, etc.
- **JDK (*Java Development Kit*):** a software development kit (SDK) to product Java programs.
- **SDK (*Software development kit*)**: provides the API *Application programming interface* libraries and developer tools necessary to build, test, and debug apps for Android.
- **ADT (*Android Development Tools*):**is a plugin for Eclipse that provides a professional-grade development environment for building Android apps. It is a full Java IDE with advanced features to help building, testing, debugging and packaging Android apps.
- **AVD (*Android Virtual Device*)**: is an emulator configuration that allows to model an actual device by defining hardware and software options to be emulated.

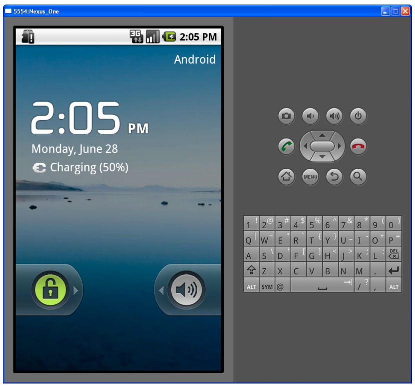


Figure 3.1: Example of Android Virtual Device

Further software used to develop the application:

- **Apache:** a public-domain open source Web server developed by a loosely knit group of programmers and anyone can adapt the server for specific needs [51].

3.2 Android and Sensors

As a side note, we give some insights about Android platform and sensors. Android supports several sensors via the class android.hardware.SensorManager. It is possible to identify the sensors that are on a device by creating an instance of the SensorManager class, which is done by calling the getSystemService() method. Sensor types supported by the Android platform are: accelerometer, temperature, gravity, gyroscope, light, linear acceleration, magnetic field, orientation, pressure, proximity, relative humidity and rotation vector [52].

3.3 TestGateway

Because input environment values cannot be directly extracted from wireless sensors, for this thesis it has been used a simulated virtual scenario TestGateway project developed by the SIGnet group mentioned in the introduction. TestGateway replicates exactly the behaviour and performance of a real WSN. Thus, to calculate the temperature, humidity and luminosity a simple function has been used that transforms a given ADC data into the corresponding value of the parameters.

TestGateway is the simulation of a WSN and below is a diagram showing how TestGateway works. The simulation has many tasks but in this figure it is represented only the one concerning the application.

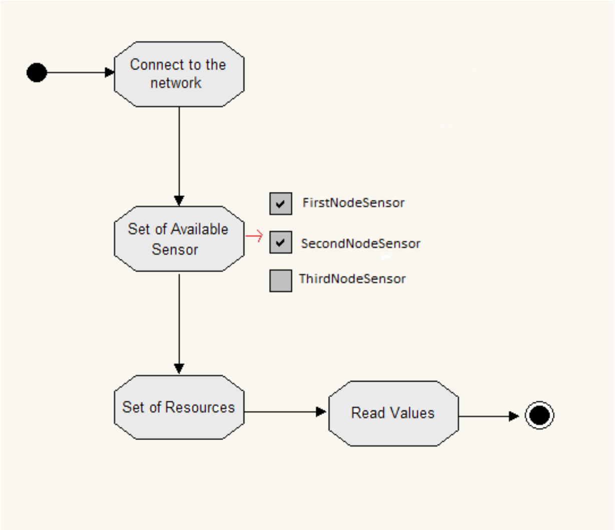


Figure 3.2: Diagram of TestGateway

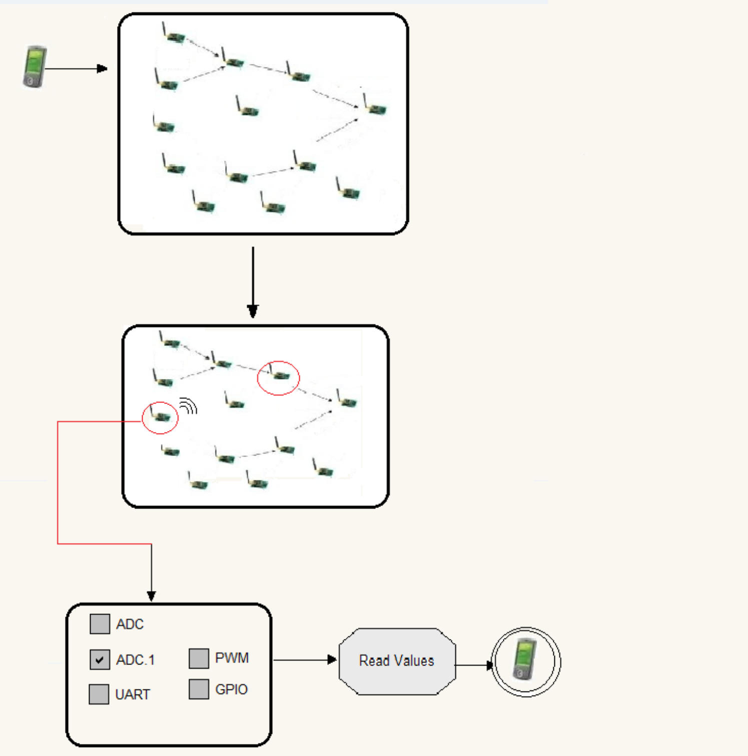
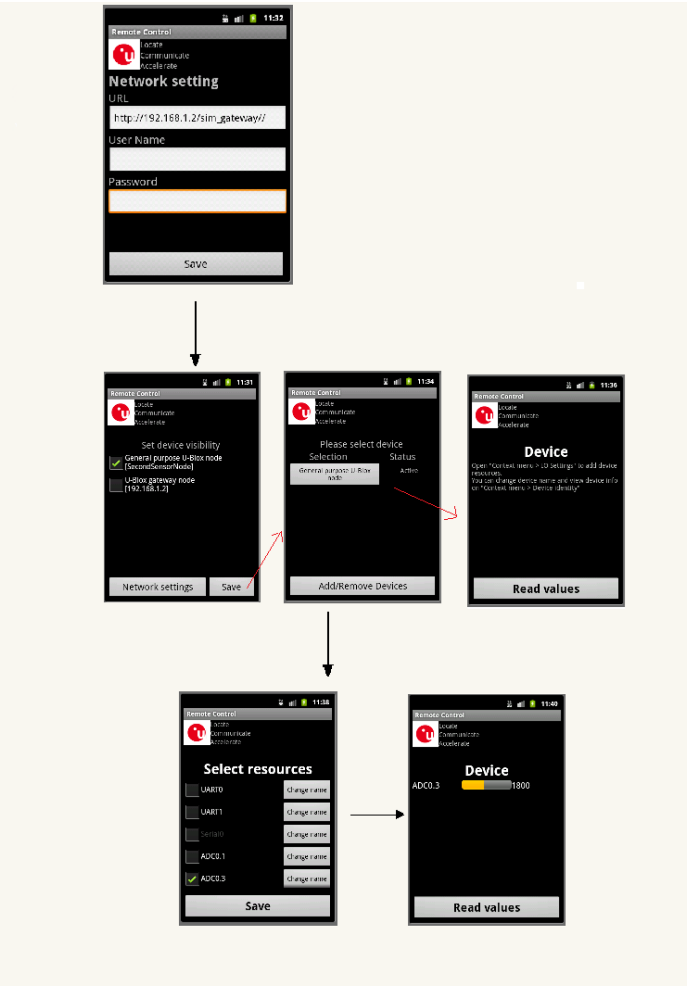


Figure 3.3: Diagram of TestGateway

The work of TestGateway interface is easy and intuitive to understand. The client identify itself via mobile with username and password, and requires the access to the wireless sensor network. The application accepts and responds showing which sensors are available for the interrogation. The client selects the device in which is interested, then the application, through a menu, shows the possible resources to choose for the data reading.

The following diagram shows the Activities of TestGateway. Activity is an application component that provides a screen with which users can interact in order to do something. Each activity is given a window in which to plot its graphic user interface [55].



http://192.168.1.2/sim_gateway//

Open "Context menu > IO Settings" to add device resources.
You can change device name and view device info on "Context menu > Device identity"

Figure 3.4: Activities of TestGateway

3.4 User interface development

Once it was explained how the simulated network works, now we analyze the steps made to run the application and to display the graphic interface with the sensor information visualization.

The Java application (Remote Control) which interact with TestGateway owns a subclass of the super class ”peripheral” for any resources available for the consulting. The class contain a method that allow the access to the file which contains the information of the resource.

The first step was the creation of the pertinent sub-class for the application,that is, those referred to temperature, humidity and luminosity. Each one of these sub-class performs dual functions and have dual methods. .There is a method call get() which has a special role. I must to clarify that to make easier it was decided to take value from an existing resource (correspond to an existing file) and to transform it in the physical measure to display. The resource chosen was the ADC. The method takes the value allocated to the ADC and transforms it in a temperature, humidity and luminosity value. This is done by using a simple linear function matching each dimension. We assume to use a 12bit- ADC so the possibility value span from 0 to 4095. Thus, the lower(highest) data of the ADC corresponds at the lower(highest) values that each physical property can assume.

The following graphics plot the relation between ADC and the measures.

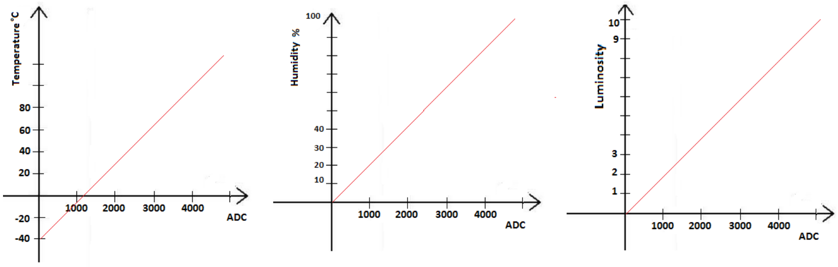


Figure 3.5: ADC Fuction

A widget was added as type button (typed View Data) in the Device Activity (widgets are user interface (UI) elements to use on the Application screen [56]). After clicking the View Data button the user will be directed to Secondary Activity in order to visualize the obtained values.

The layout of Secondary Activity consists in of widgets as type ImageView and two buttons. In the first column the application will display temperature, humidity and luminosity icons and beside them their values. The values will change every time that the value of the ADC changes.

Thus, for the management of graphical images used to display on the screen a small size database was created. The database are stores the paths of the picture allocation useful for the display. The application executes a query which returns the image that will put in each widget as type imageView.

Finally, Secondary Activity gives the possibility to turn back and take a new data and/or save the data reading in a database which also saves current data and current time.

The application has been tested on a Samsung mobile having Android 2.3.6 version.

CHAPTER

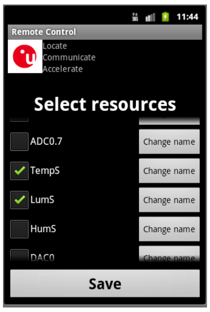
FOUR

APPLICATION PERFORMANCE

This chapter illustrates the final result of the work. The following pictures show screen captures taken while running the application. Screen captures were made on Dalvik Debug Monitor Server (DDMS) [53]. DDMS is a debugging tool that is integrated with Eclipse to provide screen captures from android device or emulator and services like port forwarding, thread, Logcat and more information [54]. The pictures show the user interfaces and functional implementations added of the TestGateway activities explained in the above section. It is also shown the new activity (Secondary Activity) created for the visualization of temperature, humidity and luminosity.

4.1 Select Resources

This activity shows the resources which a sensor node owns. In this activity were added the three environment measures wanted to be sensed (called TempS, LumS, HumS). When the user clicks on the check box the resources are activated for the use.

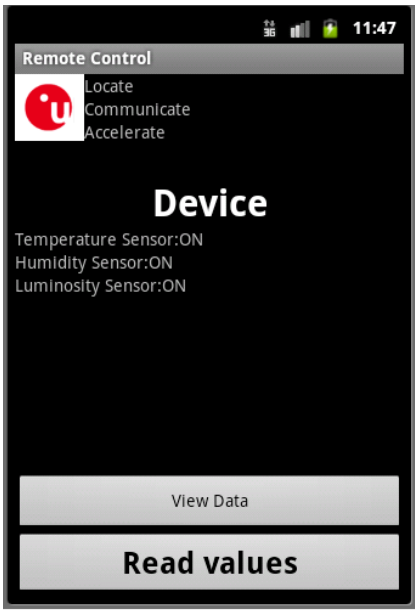


☐ ADC0.7
☑ TempS
☑ LumS
☐ HumS
☐ DAC0

Figure 4.1: New Resurces added

4.2 Device

In the Device Activity the function was added which informs the user that the resource were switched on. The button ”View Data”, which has also been added, directs the user to the Secondary Activity for the visualization of data.



Device

Temperature Sensor:ON
Humidity Sensor:ON
Luminosity Sensor:ON

Figure 4.2: Resource On

4.3 Data Values

The figure 4.3 represents the layout of the Secondary Activity. On the screen are displayed the corresponding readings related to two different values of the ADC. On the left it is shown the values for an ADC equal to 1800. In the right it is shown the values for an ADC equal to 1300. The button at the bottom of the screen allows the user either turn back or to save the data readings. By salving the data the user will be directed to a new activity that informs the success of the action.

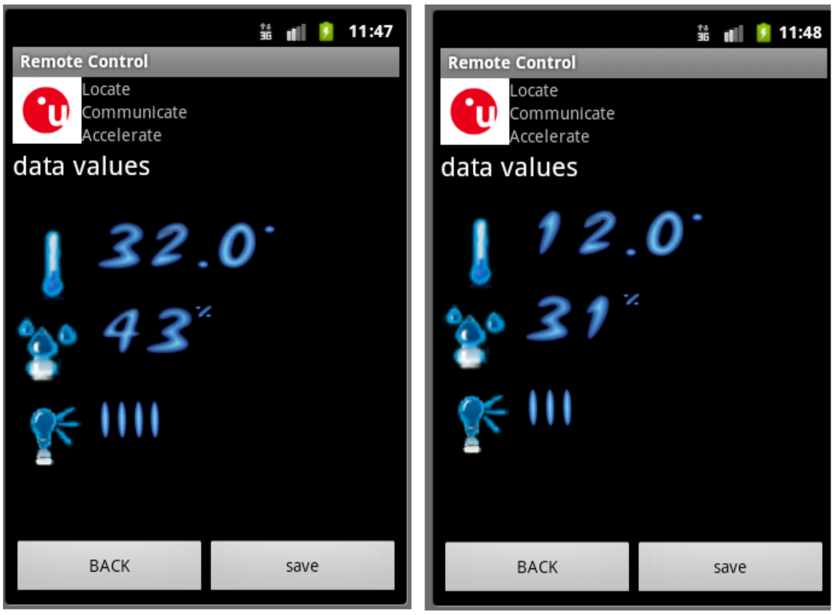
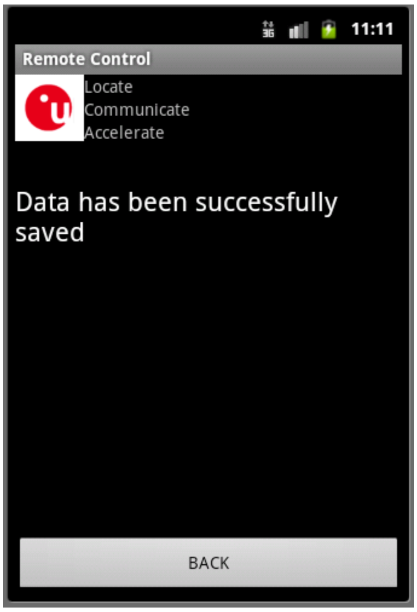


Figure 4.3: Data values



Data has been successfully saved

Figure 4.4: Data Saved

CHAPTER

FIVE

CONCLUSION AND FUTURE WORK

Nowadays to have information at hand is becoming not only a convenience but almost a need. The way to have the information that we want is easy, we just have to take our mobile phone. It will be interesting if in a near future all of us will be able to control our houses with a simple pocket device. We can affirm that is possible mixing together WSNs and mobile application technologies. The application field of this set of technology is very large and developers and programmers are still working to integrate them in our life.

The goals of the thesis have been achieved using Java in Android platform. Several possible approaches and techniques to develop the interface can be implemented. Therefore, Java was used because of its simplify and usability challenges.

The application realized might apply in a real WSN to take real values in a real practical scenario. There are many ways that the application could be improved. For example, others functions cam be added that allow the user to research through the data saved which node presents the highest temperature, or which node receives less light. Going further, it will be interesting if the application by itself takes values in intervals of times and announces the user (e.g. an acoustic signal) when the data goes over or below a specific threshold.

BIBLIOGRAPHY

[1] http://it.wikipedia.org/wiki/Legge_di_Moore

[2] *Wireless Sensor Network, an information approach* Feng Zhao, Leonidas Guibas, Series Editor, David Clark, M.I.T

[3] *Fundamentals of Wireless Sensor Network, Theory and Practice* Wiley Series on Wireless Communications and Mobile Computing Series Editors: Dr Xuemin (Sherman) Shen, University of Waterloo, Canada Dr Yi Pan, Georgia State University, USA, 2008

[4] *Algorithms and Protocols for Wireless Sensor Networks* A. Boukerche, Wiley Sons, New Jersey, USA, 2009

[5] Technology Review (MIT), February 2003, *10 Emerging Technologies That Will Change the World* from http://www.technologyreview.com/read_article.aspx?id=13060&ch=infotech

[6] *Environmental Sensor Networks: A revolution in the earth system science?*, May 2006 from http://eprints.soton.ac.uk/263093/1/esn.pdf

[7] *Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks*, from http://www.cs.berkeley.edu/~binetude/work/ipsn07_ggb.pdf

[8] *Wireless Sensor Networks for Habitat Monitoring*, from http://www.polastre.com/papers/wsna02.pdf

[9] http://www.unipd.it/uniscienze/articoli/WISEWAI.html

[10] *The Wireless Sensor Networks for City-Wide Ambient Intelligence (WISE-WAI) Project*, Casari P., Catellani A., Cenedese A., Lora C.,

Rossi M., Schenato L., Zorzi M. 2009 from http://cariparo.dei.unipd.it/

[11] *Glacial environment monitoring using sensor networks*, Martinez, Padhy, Riddoch, Ong (2005), Proc. of Workshop on Real-World Wireless Sensor Networks

[12] http://www2.hawaii.edu/~esb/prof/pub/ijhpca02.pdf

[13] http://www.stccmop.org/

[14] http://nesl.ee.ucla.edu/fw/kansal/nims_mtts05.pdf

[15] *24 Million Industrial Wireless Sensing Points in 2016*, from http://www.onworld.com/news/newsiwsn.html

[16] http://netlab.boun.edu.tr/papers2/pimrc04.pdf

[17] *Ad Hoc and Sensor Networks* YI PAN YANG XIAO EDITORS, 2006 NOVA science publisher.

[18] *Wireless Sensor Networks*a Survey. Computer Networks, AKYILDIZ, SANKARASUBRAMANIAM, CAYIRCI

[19] *Overview of sensor networks* Guest editors introduction, David Culler, Deborah Estrin, Mani Srivastava, Published by the IEEE Computer Society, 2004

[20] *Sensor Network:an Overview* from http://www.eps2009.djinod.com/docs/09-02-01/sensor_networks__an_overview.pdf

[21] *Handbook of sensor networks, Algorithms and Architectures* Ivan Stojmenovic, Series Editor Wiley, University of Ottawa, 2005

[22] *TinyOs* an open source operating system for WSN from htpp://www.tinyos.net

[23] http://ipadapplication-development.com/article/telosb-the-wireless-sensor-network

[24] http://hire-webdeveloper.com/article/telosb-mote-features-and-specifications/

[25] *Ultra-Low Energy Wireless Sensor Networks in Practice* Kuorilehto M., Kohvakka M., Suhonen J., Hamalaienen P., Hannikainen M., Hamalainen T., 2007, Wiley, England.

[26] `http://elabs.altervista.org/articoli/SHT11/SHT11.php`

[27] `HOMEAUTOMATIONTHETARGETMARKETANALYSIS` `http://diromero.wordpress.com/2011/10/07/home-automation/`

[28] *A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses*Hui Liu, Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, 21-25 Sept. 2007

[29] *Wireless sensor networks for industrial applications* Xingfa Shen Inst. of Modern Control Eng., Zhejiang Univ., Hangzhou, China Zhi Wang ; Youxian Sun Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on, 15-19 June 2004

[30] *Application of Wireless Sensor Network to Military Information Integration*Diamond, S.M. Training Squadron Two, Whiting Field Ceruti, M.G., Industrial Informatics, 2007 5th IEEE International Conference on, 23-27 June 2007

[31] *Instrumenting wireless sensor networks for real-time surveillance* Chen, P. ,Dept. of Electr. Eng. & Comput. Sci., California Univ., Berkeley, CA, Songhwai Oh ; Manzo, M. ; Sinopoli, B. ; Sharp, C. ; Whitehouse, K. ; Tolle, O. ; Jaein Jeong ; Dutta, P. ; Hui, J. ; Schaffert, S. ; Sukun Kim ; Taneja, J. ; Zhu, B. ; Roosta, T. ; Howard, M. ; Culler, D. ; Sastry, S., 15-19 May 2006

[32] *A realtime dynamic traffic control system based on wireless sensor network* Chen Wenjie Dept. of Comput. Sci. & Eng., Fudan Univ., Shanghai, China Chen Lifeng ; Chen Zhanglong ; Tu Shiliang, Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on, 14-17 June 2005

[33] *BODY SENSOR NETWORK A WIRELESS SENSOR PLATFORM FOR PERVASIVE HEALTHCARE MONITORING* `http://csis.pace.edu/~marchese/CS396x/L3/p077-080.pdf`

[34] *Wireless sensor networks, SERIES IN COMMUNICATIONS AND NETWORKING* Ian F. Akyildiz, Mehmet Can Vuran, Georgia Institute of Technology, USA WILEY 2010

[35] *Protocols and Architectures for Wireless Sensor Networks* Holger Karl and Andreas Willig, 2005 John Wiley & sons

[36] *A Survey on Fault Tolerance in Wireless Sensor Networks* Luciana Moreira Sá de Souza, Harald Vogt, Michael Beigl, Germany

[37] *On the Lifetime of Wireless Sensor Networks* ISABEL DIETRICH, FALKO DRESSLER Dept. of Computer Science, University of Erlangen, Germany

[38] *Scalability of Sensor Networks* Ameesh Pandya, Huiyu Luo, and Gregory Pottie, Department of Electrical Engineering University of California, Los Angeles

[39] *New challenges in wireless sensor networks: fault tolerance and real time* J. V. Capella, R. Ors, A. Bonastre, J. J. Serrano, Technical University of Valencia Camino de Vera s/n, 46022 Valencia, Spain

[40] *Security and privacy in sensor networks* Haowen Chan, Carnegie Mellon Univ., Pittsburgh, PA, USA Perrig, A. , published in:Computer, Oct. 2003

[41] *ANALYSIS ON DEPLOYMENT COST AND NETWORK PERFORMANCE FOR HETEROGENEOUS WIRELESS SENSOR NETWORKS*, Dilip Kumar, Trilok. C Aseri, R.B Patel, International Journal of Computer science & Information Technology (IJCSIT), Vol 1, No 2, November 2009

[42] *Programming Paradigms and Middleware for Sensor Networks*Kay Romer Institute for Pervasive Computing ETH Zurich

[43] `http://source.android.com/compatibility/index.html`

[44] *Beginning Android Application development* Wei Meng Lee, Published by Wiley Publishing, Crosspoint Boulevard Indianapolis, 2011

[45] `http://developer.android.com/reference/android/database/sqlite/package-summary.html`

[46] http://en.wikipedia.org/wiki/IEEE_802.11

[47] http://it.wikipedia.org/wiki/MP3

[48] http://en.wikipedia.org/wiki/MIDI

[49] http://pam.wikipedia.org/wiki/3GP

[50] http://www.eclipse.org/downloads/

[51] http://www.apache.org/

[52] http://developer.android.com/reference/android/hardware/Sensor.html

[53] http://redomino.com/it/labs/progetti/google-android/documentazione/materiale/usare-lo-strumento-dalvik-debug-monitor-service-ddms/

[54] http://developer.android.com/tools/debugging/ddms.html

[55] http://developer.android.com/guide/components/activities.html

[56] http://developer.android.com/reference/android/widget/package-summary.html