Design and Implementation of a Wireless Sensor Network for Smart Home Applications

PRODROMOS-VASILEIOS MEKIKIS

Master’s Degree Project
Stockholm, Sweden September 2012

XR-EE-RT 2012:020
Design and Implementation of a Wireless Sensor Network for Smart Home Applications

PRODROMOS-VASILEIOS MEKIKIS

Stockholm 2012

Master’s Thesis at Electrical Engineering
Supervisor, Examiner: Carlo Fischione
Abstract

In smart homes devices take partial control of the house and make decisions that increase its safety and functionality. Due to the high cost of cable installation, wireless sensor networks are considered to be a good choice for smart home systems. In smart homes, to reliably detect events such as intrusions, gas leakages, or accidents, is an essential functionality. A correct control action, such as alarming or shutting gas pipes relies entirely on reliable event detection. Given that reliability is the major concern in these devices, detection solutions should be found that make decisions with the smallest possible cost. A way to achieve this is by using detection and data fusion techniques so measurements from multiple sensors make the final decision whether the event has happened or not. Furthermore, estimation aided detection in every sensor is essential so as to provide a noiseless environment for the decision concerning the happening of an event in each node of the network. In this thesis, several distributed detection techniques are reviewed and their suitability for low power sensor networks is investigated. A wireless sensor network is designed and completely implemented in order to test the functionality and the reliability of the methods. In the presence of a detected event the sensor network sends a Twitter notification to the user and, meanwhile, actuates a control decision that could solve the detected problem. The experiments show that the studied detected methods are able to offer reliable performance even in the presence of high noises in the measurements. It is concluded that wireless sensor networks can be effectively used in smart home applications, provided that detection methods of low complexity and reliability are implemented.

Key words:
Wireless Sensor Network, Optimal data fusion, Detection, Estimation, Kalman filter, TinyOS, telosB platform, Twitter notification, Actuation
Acknowledgements

I would like to thank my examiner Carlo Fischione for all his valuable help and guidance throughout this thesis. Furthermore, I would like to thank my family and friends that supported me during this period.
Contents

1 Introduction ........................................... 1
   1.1 Objective ........................................ 1
   1.2 Related work ..................................... 2
   1.3 Thesis Organization ............................. 2

2 Wireless Sensor Networks ............................. 5
   2.1 Network .......................................... 6
      2.1.1 Peer-to-Peer ................................... 6
      2.1.2 Star .......................................... 6
      2.1.3 Cluster-Tree ................................. 6
      2.1.4 Mesh ......................................... 7
   2.2 Applications ...................................... 7
   2.3 Building Automation ............................. 8
      2.3.1 Characteristics ............................... 8
      2.3.2 Classification ............................... 9
      2.3.3 Applications ................................. 11
      2.3.4 Conclusions ................................ 12
   2.4 RFID in WSN ...................................... 13
      2.4.1 Analysis ..................................... 13
      2.4.2 Conclusions ................................ 14
   2.5 Modulation on WSN ............................... 15
      2.5.1 Spreading ..................................... 16
      2.5.2 Benefits of spread-spectrum .................. 18
      2.5.3 Direct-Sequence Spread Spectrum (DSSS) ....... 18
      2.5.4 Chirp Spread Spectrum (CSS) ................ 20
   2.6 Security .......................................... 21
      2.6.1 Ways to attack ............................... 21
      2.6.2 Ways to secure ............................... 22
      2.6.3 Scrambler .................................... 24
      2.6.4 Security in CC2420 ......................... 29
      2.6.5 Data Reliability ............................. 30
2.6.6 Conclusion .................................................. 30
2.7 Localization .................................................. 30
  2.7.1 Ranging .................................................... 31
  2.7.2 Positioning ................................................ 32
  2.7.3 Refinement ................................................ 32
  2.7.4 Conclusion ................................................ 33

3 Detection and Estimation ........................................ 35
  3.1 Detection ................................................... 35
    3.1.1 Introduction to detection ................................ 35
    3.1.2 Optimal data fusion .................................... 38
    3.1.3 The case of uncertain prior probabilities ............... 39
    3.1.4 Detection cases depending on the nature of the system ... 40
  3.2 Estimation ................................................... 47
    3.2.1 Kalman filtering ........................................ 47
    3.2.2 Sequential measurements from one sensor ................. 49
    3.2.3 Dynamic sensor fusion ................................... 49
  3.3 Discussion ................................................... 51

4 System: Software and Hardware .................................. 53
  4.1 Hardware ..................................................... 53
    4.1.1 Nodes .................................................... 53
    4.1.2 Sensors ................................................... 54
    4.1.3 Gateway .................................................. 55
  4.2 Software ...................................................... 55
    4.2.1 TinyOS .................................................... 56
    4.2.2 Graphical User Interface ................................ 57
  4.3 Graphical User Interface tutorial ............................ 58
    4.3.1 General Description ..................................... 58
    4.3.2 Operation ................................................ 61
    4.3.3 Conclusion ................................................. 62

5 Simulations and Experiments ................................... 65
  5.1 Estimation .................................................... 65
    5.1.1 Kalman filtering performance on the final decision ....... 65
  5.2 Detection and data fusion .................................... 68
    5.2.1 Setup of the simulation .................................. 69
    5.2.2 Simple data fusion ....................................... 69
    5.2.3 Data fusion with adjustable $P_F$ and $P_M$ ............... 69
    5.2.4 Data fusion using knowledge from the neighbors ........... 70
  5.3 Discussion .................................................... 71
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Peer-to-Peer and Cluster-Tree network topologies</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Mesh and Star network topologies</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Spread spectrum communication system</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Spreadsed signal</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Resistance to interference in spreading</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Resistance to interception in spreading</td>
<td>19</td>
</tr>
<tr>
<td>2.7</td>
<td>Coded sequence creation in DSSS</td>
<td>20</td>
</tr>
<tr>
<td>2.8</td>
<td>Direct-Sequence Spread Spectrum modulation technique</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>A 4-bit standard LFSR with polynomial $x^4 + x^3 + 1$</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>A 4-bit additive scrambler with polynomial $x^4 + x^3 + 1$</td>
<td>25</td>
</tr>
<tr>
<td>2.11</td>
<td>A 4-bit additive descrambler with polynomial $x^4 + x^3 + 1$</td>
<td>26</td>
</tr>
<tr>
<td>2.12</td>
<td>A 4-bit multiplicative scrambler with polynomial $x^4 + x^3 + 1$</td>
<td>27</td>
</tr>
<tr>
<td>2.13</td>
<td>A 4-bit multiplicative descrambler with polynomial $x^4 + x^3 + 1$</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Parallel topology with fusion center</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Flowchart with the algorithm for the local estimates</td>
<td>43</td>
</tr>
<tr>
<td>3.3</td>
<td>Probability density functions of the two hypotheses and the $\gamma$ threshold</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Photo Sensitivity of the light sensors</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Graphical User Interface</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Initial window of the GUI</td>
<td>59</td>
</tr>
<tr>
<td>4.4</td>
<td>The Properties window</td>
<td>60</td>
</tr>
<tr>
<td>4.5</td>
<td>The Graphs window</td>
<td>61</td>
</tr>
<tr>
<td>4.6</td>
<td>The Legend window</td>
<td>62</td>
</tr>
<tr>
<td>4.7</td>
<td>One node has detected an event but the global decision is $u_i = 0$</td>
<td>63</td>
</tr>
<tr>
<td>4.8</td>
<td>The global decision is $u_i = 1$ and the gateway becomes red</td>
<td>63</td>
</tr>
<tr>
<td>4.9</td>
<td>Twitter notification example</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>Kalman filtering</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>Adjustable $P_{F_i}$ for every sensor</td>
<td>70</td>
</tr>
</tbody>
</table>
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>BAS</td>
<td>Building Automation Systems</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CSS</td>
<td>Chirp Spread Spectrum</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DOS</td>
<td>Denial-of-Service</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct-Sequence Spread Spectrum</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency-Hopping Spread Spectrum</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>LRT</td>
<td>Likelihood Ratio Test</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum <em>A Posteriori</em></td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-Peer</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio-Frequency Identification</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>THSS</td>
<td>Time-Hopping Spread Spectrum</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

A Wireless Sensor Network (WSN) is a system of distributed autonomous sensors, which are called nodes, that monitor the status of the space in which they are operating. The nodes route the acquired data in packets wirelessly to a main location which is called gateway or sink. Each node is consisted of four important parts:

1. An antenna, that receives and transmits data,
2. sensors, that measure the conditions of the environment,
3. a CPU, that is responsible for the control of the node,
4. and a power unit, that supplies energy to all the other parts.

During the last decade wireless sensor networks have become very popular. A lot of research and many applications have been developed. Area monitoring, environmental sensing and industrial monitoring are only few examples among the most important applications that WSNs have revolutionized. However, they are not yet the standard in autonomous sensing and decision making due to their lack in reliability. For example, the cost of a missed or a false alarm event could have very high cost. How could we understand invalid data due to a malfunction of a sensor?

Many techniques are being studied for this purpose and a lot of research is being done. In this thesis we study the estimation theory in order to filter out the noise from the sensor measurements and the distributed detection theory so as to increase the reliability of the final decision of the system. In that way we can increase the reliability of a WSN.

1.1 Objective

The main goal of this thesis is to apply estimation and distributed detection techniques on a wireless sensor network operated using telosB motes that run TinyOS.
Such a main goal is obtained by the following sub-tasks:

- Investigation of the possibility to build a wireless sensor network with RFID nodes.
- Investigation of the possible security techniques on the network.
- Investigation on the modulation techniques.
- Implementation of the WSN with telosB motes.
- Implementation of a graphic user interface for the surveillance of the WSN.

1.2 Related work

There is much research both on distributed detection and estimation. In this thesis, we consider detection theory based on the book "Distributed Detection and Data Fusion" by Pramod K. Varshney [Var96]. The optimal fusion rule therein proposed adopted in this thesis and is used for the distributed detection part. Furthermore, the book chapter of Y. Xu, V. Gupta and C. Fischione on "Distributed Estimation" [XGF13] was the basis for the estimation using sequential measurements from one sensor as well as for the investigation of the dynamic sensor fusion.

1.3 Thesis Organization

In chapter 2, the basic theory and characteristics of wireless sensor networks are presented. In the beginning there is an introduction to the various topologies, afterwards a description of the Building Automation Systems with their characteristics and applications, then an analysis of the possibility to use the RFID technology in wireless sensor networks and finally some investigation on the modulation, security and localisation in wireless sensor networks.

In chapter 3, the fundamental principles of detection and estimation theory are presented. All the available techniques that fit with the thesis’s purpose are discussed. In the end there is a discussion where it is decided which of these techniques will be adopted for the design of the wireless sensor network.

In chapter 4, a description of the hardware and software that was used is presented. Furthermore, there is a tutorial on the graphical user interface that was implemented in order to control and visualise the system.

In chapter 5, the reader can evaluate the results of the thesis through the simulations and experiments that were conducted on the suggested wireless sensor network both from the detection theory and estimation point of view.
1.3. **THESIS ORGANIZATION**

Chapter 6 makes a proposal for a smart home wireless sensor where all the suitable techniques are listed according to the requirements of the final user.

Finally, chapter 7 is the conclusion of this thesis where there is a discussion on the results of the work, a critique which lists the problems of the system and future work and improvements that the designed wireless sensor network could benefit from.
Chapter 2

Wireless Sensor Networks

A wireless sensor network is a collection of nodes that interact with each other in order to form a network. Each node consists of processing capability, memory, an RF transceiver, power source and various sensors depending on the application. There are numerous application areas that wireless sensor networks are used in including environmental, military, medical, building automation, entertainment and many others.

In order to take advantage of the processing capabilities, a node needs an operating system. There are many operating systems for wireless sensor networks such as TinyOS which is written in nesC, Contiki OS and LiteOS that are written in plain C. All of them are specifically designed for wireless sensor networks and each of them has advantages and disadvantages.

For the routing capability there is a need for protocols. IEEE 802.15.4 is the most significant standard which specifies the physical layer and media access control (MAC) for low-rate wireless personal area networks. It is maintained by the IEEE 802.15 working group and is the basis for the ZigBee and WirelessHART specifications that further extend the standard by developing the upper OSI layers.

One of the biggest limitations for the operation of a WSN is the power source. The need for cordless solutions limits WSNs to have relatively small lifetime if nodes use a battery as a power source. There are techniques that can increase the lifetime up to a couple of years by putting the nodes into sleep mode when their operation is not needed. Furthermore, there are solutions in which the nodes scavenge power from the environment or they are event-triggered. In these situations the nodes can operate without any problem for many years.

Another important aspect of a WSN is security. There are many ways to attack a WSN by exploiting weaknesses in software implementation or by manipulating the hardware of a node. However there are also methods to secure a WSN mainly by means of encryption.

In the following chapters a more extensive description is given that covers each
of the above aspects of a wireless sensor network.

2.1 Network

An important decision that affects the design of a WSN is the topology of the network. There are four topologies that are common and they are briefly discussed in the following paragraphs.

2.1.1 Peer-to-Peer

In Peer-to-Peer (P2P) networks each node communicates directly with another node without a gateway that collects all the information. Each node is able to function as both a client and a server in contrast to the client-server model where clients send data and the server receives them. A P2P network is illustrated in figure 2.1.

2.1.2 Star

Star networks are connected to a centralized communication hub which is called gateway or sink. The nodes cannot communicate directly to each other, but all communications must be routed through the gateway. In this scheme the gateway is the server and the rest nodes the clients. In figure 2.2 a Star network is shown.

2.1.3 Cluster-Tree

In this kind of networks there is a central hub called Root node whose children, one level down, are the servers of their star network. This topology is a combination of Peer-to-Peer and Star topology. An illustration of the Cluster-Tree topology is shown in figure 2.1.
2.1.4 Mesh

Mesh or multi-hop networks allow data to 'hop' from node to node, thus each node serves as a relay for the others, until they reach their final destination. This type of network is one of the most complex, but it is self-healing in the sense that nodes collaborate to propagate the data in the network.

The self-healing capability makes a network to operate when a number of nodes breaks down or a connection goes bad. As a result the network is more reliable as it is common to exist more than one paths between a source and a destination in the network. The decision to which node will the data be propagated is usually taken according to the power level of the relay node, the traffic in this node or the shortest path. In figure 2.2 a mesh network is illustrated.

2.2 Applications

There is a countless number of WSN applications today and even more applications that are planned to be developed in the future. In this section it is shown a brief list of the most important applications.

- Area monitoring, e.g to detect enemy intrusion
- Forest fires detection
- Air pollution monitoring
- Machine health monitoring
- Data Logging
- Industrial sense and control applications
CHAPTER 2. WIRELESS SENSOR NETWORKS

- Greenhouse monitoring
- Structural monitoring, e.g. to prevent damages from earthquakes
- Building automation

From the list above it is obvious that wireless sensor networks could help a lot on the protection of the environment and make people’s life easier and more comfortable.

2.3 Building Automation

Building automation systems (BASs) are the kind of systems that consist of a number of sensors, actuators and controllers and provide automatic control in procedures that otherwise would have been done by humans. Communication among these components takes place over a network, whose design depends on several constraints. A smart home includes building automation systems in order to provide security and comfort to the residents of the house and save energy.

The concept of “smart home” is not something completely new. BAS have been researched since mid 60’s by the industry when electronics and communication protocols were still considered as the technology of the future [Bot79]. Back then BAS were intended for large buildings and factories, since the cost was too high to consider them in a typical family house.

After some years when the technology had matured enough and electronics became cheaper, smart or intelligent homes started to become possible for the future. In the paper of [Skr87] there is a very interesting description of how the intelligent home would look like 25 years later in 2010. Many of the systems that are described in the paper are considered obvious today and we depend on them in our everyday life. The author says that the key elements that will provide the possibility to build such kind of systems into the future are Informatics and Ergonomics meaning software programs and man-machine interfaces respectively.

History has been confirmed once again and building automation today is feasible for every home and is starting gradually to gain more attention. However even today these systems have some limitations that we will try to overcome using appropriate techniques. The biggest future challenges are security, integrity and the possibility to control building facilities from any device with Internet access.

2.3.1 Characteristics

The dominant goal of Building Automation Systems is to make processes more efficient, provide comfort and security. In order to achieve their goal, these systems should have the following characteristics:
2.3. BUILDING AUTOMATION

- **Security** It is very important for BASs to protect themselves from adversaries both in the physical and the digital world. The system should have algorithms that detect malicious use and reject commands that are not genuine.

- **Robustness** The systems should work always as expected. In case of malfunction the system should be able to inform the user in order to prevent unwanted events.

- **Data Integrity** Together with robustness are the most important factors of a BAS. The system should be able to understand when the value of a measurement in the gateway is different from the actual value that a sensor measured.

- **Energy efficiency** Energy consumption of the BAS should not be too large. Depending on the purpose of the system the energy consumption has different limitations, but in any case the system should not consume more energy that a certain level.

- **Cost** One of the reasons that a building automation system could be possible to be installed today in every home is the low cost. Mass production could be a key factor for this purpose.

- **Ease of use** A BAS for home should be easy to use since it is not intended for professional users. An interface that controls the whole system should be provided for the simple user.

- **Maintenance** The easier maintenance the better. The ideal would be for the simple user to be able to support the system without the constant need of a professional.

**Benefits** If the building automation systems fulfill all of the above characteristics, then there are many benefits. For instance, the owner of the system could save time and money by a) energy savings, b) energy efficiency, c) monitoring, d) remote access, e) improved building management and f) central access.

2.3.2 Classification

There are many different types of building automation systems and each of them have different applications, limitations and abilities. In this section we will describe BASs in two different ways, regarding connectivity and criticalness.
CHAPTER 2. WIRELESS SENSOR NETWORKS

Wired vs Wireless systems

One very important aspect of a BAS is the type of connectivity, because the choice changes completely the design of the system. There are two different ways to connect the nodes of such systems, with wires or wirelessly.

**Wired BAS**  Wired systems are usually used when the applications need to be robust. Usually they are used in large buildings and factories where the production is depended on them. There are three main protocols for the wired BASs

- BACnet, whose development began in 1987
- LonWorks, whose development began in 1988 and
- KNX, whose development began in 1990.

All of them are using a twisted pair for the connection among the nodes and the gateway and a repeater/router for the IP tunneling/routing. Those protocols are used many years in the industry and there is a lot of research on them. Their main advantage is the robustness and data integrity, but the installation and maintenance cost is high due to the wiring. This problem can be solved by the wireless BASs. Of course there are also other notable protocols for wired BAS that are not mentioned here.

**Wireless BAS**  It is easy to understand the benefits of the wireless systems. You can place sensors where no cabling is possible, there are not predefined connection points, mobility of wireless nodes and reduced installation cost. However there are also many challenges. The nodes usually operate with batteries and have ultra-low resources. Furthermore there is the problem of security and data integrity in a much bigger grade than in the wired systems. Also the engineers have to think about the interference caused by near-by devices and other similar problems that are caused due to the open medium. Many of those problems have been at least partially solved during the last decade when wireless systems started gaining the attention of the industry. Today there are three main wireless protocols in building automation

- ZigBee, which uses the IEEE 802.15.4 in the MAC layer
- enOcean, which harvests energy from the environment and
- KNX, which is the same protocol as the Wired-KNX but for RF.
In the following list there is a comparison between Wired and Wireless systems:

<table>
<thead>
<tr>
<th>Wired</th>
<th>Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust and reliable</td>
<td>Relatively unreliable by nature</td>
</tr>
<tr>
<td>Expensive installation</td>
<td>Cheap installation</td>
</tr>
<tr>
<td>Unlimited resources</td>
<td>Low power and memory</td>
</tr>
<tr>
<td>Static Network</td>
<td>Mobile network</td>
</tr>
<tr>
<td>Higher security</td>
<td>Lower security</td>
</tr>
</tbody>
</table>

Safety Critical vs Security Critical systems

Another way to characterize a building automation system is its criticalness. According to [NG10] and [GK10] BASs can be distinguished by two different types of services, safety- and security-critical services.

Safety-Critical Service A safety-critical service is a service where the intention is to safeguard from direct or indirect accidental impact on the health of people. For example such a service is a fire alarm BAS or a system that detects high gas levels etc.

Security-Critical Service In contrast to a safety-critical service, a security-critical service, such as physical-access control or physical-intrusion detection, is a service that protects assets from intentional malicious attacks by security countermeasures.

Other kind of services Of course there are also other kind of systems that provide less critical services. For example in a smart home there are systems that are developed to take care of the energy efficiency of the house or systems that are entertainment-oriented. These systems do not provide critical services, but they play an important role for the comfort of the user.

2.3.3 Applications

In this section a list of different applications is presented.

- **Lighting** can be turned on, off, or dimmed based on time of day, or on the measurements of sensors, photosensors and timers.
• **HVAC** Heating, ventilation and air-conditioning can be controlled by a BAS and monitored by the user in order to save energy and have less cost. An air handler can mix return and outside air so less temperature change is needed. Some external air is needed to keep the building’s air healthy.

• **Security** This application exists before the concept of the smart house. Intrusion alarms have been in the industry for a long time but today can be enhanced with cameras or other sensors that can detect easier and more efficiently the kind of intrusion.

• **Safety** Carbon monoxide and carbon dioxide sensors can be used to alarm if levels are too high. Refrigerant sensors can be used to indicate a possible refrigerant leak. Water sensors can be used to indicate a possible water leakage from pipes.

• **Domotics**\(^1\) are usually used if there is a detection that compromises the safety of the building so as to take appropriate measures. For example when there is a water leakage from a pipe, the system could close the water supply until the problem is fixed etc.

• **White Goods** Water temperature controllers, automated kitchen equipment and other white goods can be controlled automatically or from a distant user.

• **Health-Care Monitoring** A very important application for elderly and people with special needs could be greatly benefited by building automation systems that help them inform their doctors or make their life easier in any other way.

### 2.3.4 Conclusions

From the previous sections we can see that building automation systems are very important for the buildings not only of the future but also for today. The advantages are so many that there is absolutely no reason to not install a system that doesn’t only save money and energy in a building but also increases its safety and security. Wireless sensor networks can provide such a system with lower cost than wired systems, but they still need research in order to increase their security and data integrity thus to become more robust and reliable.

\(^1\)The term domotics is a contraction of the words: domus (lat. = home or house) and informatics (=the science concerned with the collection, transmission, storage, processing and display of information).
2.4 RFID in WSN

In this section it will be discussed whether RFID is suitable for wireless sensor networks on smart building applications. Section 2.4.1 gives an analysis on all the available RFID techniques, their constraints and benefits. In section 2.4.2 are discussed the most suitable RFID systems and in which cases are RFID better that other technologies.

2.4.1 Analysis

Radio frequency identification (RFID) is a relatively old technique that is able to identify objects from a specific distance. This kind of systems consist of readers and tags. Tags are usually attached on an object that needs to be identified and the reader receives the identity of each tag from a specific distance. There are two major types of this technology that differ in many aspects from each other, inductive and radiative RFID.

**Inductive systems** Inductive systems exploit magnetic fields for the communication between the tag and the reader. Their main characteristic is the short range which is comparable to the reader antenna (usually less than 1 meter) since the energy from the reader antenna is falling away as the cube of distance. Furthermore they operate in low (LF 125/134 kHz) and high frequencies (HF 13.56 MHz) which give them the opportunity for higher skin depth comparing to radiative systems. The tags in inductive systems are typically passive.

**Radiative systems** On the other hand radiative systems operate in ultra high (UHF 900 MHz) and microwave (2.4 GHz) frequencies. Their antenna is comparable in size to the wavelength and they use electromagnetic waves for the communication between the reader and the tag. The intensity of the EM wave falls off as the square of the distance traveled. However due to the high frequency, the skin depth is much smaller which on the other hand makes radiative systems easier for shielding.

One other way to distinguish RFID systems is the tag’s demand on power. The last decade RFID have become popular mainly because of their ability to scavenge power from their reader. The tags that use this useful property are called passive and they operate with the power that their antenna received from the signal of the reader. A small processor and memory is powered and they backscatter the identity of the tag to the reader. Furthermore, there are the semi-passive tags which use a local power supply for the operation of their circuit, however they use backscattered communications for their respond to the reader. Finally, active tags is called the category of tags that don’t require power from the reader, but they have their own source. Those types of tags can operate in a more complex way by sending not only
their ID but also other information. Their range, depending on the application, can reach more than 100m. In the following list the reader can see the most important characteristics for each type of RFID:

- Inductive RFID use LF and HF. Their range is comparable to antenna size. They can penetrate thicker materials. They use coil antennas.

- Radiative RFID have longer range, higher data rate but their penetration to conductive materials is negligible. They use dipole antennas.
  - UHF RFID operate at 900 MHz, have a range longer than 20m and suffer from 3rd order distortion caused by near-by readers.
  - Microwave RFID operate at 2.4 GHz and they have larger data rate than UHF

- Types of tags
  - Passive tags use power from their reader to support their circuit and to send information back. Short range.
  - Semi-passive tags have a local battery for their circuitry but still use backscattered communication for their respond to the reader. They have long range. They are not so reliable because of the nature of EM waves.
  - Active tags have a local source for all of their operations. Their range depends on the applications and can be up to hundreds of meters. They are more reliable than semi-passive tags.

2.4.2 Conclusions

As it is obvious from section 2.4.1, RFID systems can cover a wide range of different applications. In order to choose the most suitable technology, one should first define the constraints of the application and then choose according to the different factors like range, skin depth, frequency, data rate and power requirements.

In the project of wireless sensor networks for smart building applications there are a number of constraints. First of all the distance between the nodes and the gateway will be more than 5 meters. Secondly, all the nodes will be located inside walls, which we should consider because of the skin depth. Also, we don’t need high data rates but we care about interference due to near-by readers. Finally there is a strong need for low power consumption from the node side. According to these limitations of the project we come to a number of conclusions:

- Inductive coupling is not suitable due to the very short range.
Radiative systems can satisfy the range factor.

Passive tags are not suitable for this application.

Semi-passive tags need a local battery but still use less power than active tags. However they are less reliable.

Active tags are the most suitable for the purpose of the project. They are reliable and cover all the limitations except for the power supply. In active tags we need a local power which is responsible for all of the operations of the tag.

Both 900 MHz and 2.4 GHz systems are capable to satisfy the needs of the project.

To sum up, active and semi-passive tags that operate at 900 MHz or 2.4 GHz are capable to operate as expected. However, both tags need a local battery to operate which means that the nodes will stop operating when their battery is over. Unfortunately, this stops making RFID the most suitable choice among other technologies.

2.5 Modulation on WSN

One of the most important problems to solve in wireless communications is how to share an antenna so that as many users as possible can communicate simultaneously. The problem of multiple access has been investigated for many years and many solutions have been used.

In the beginning, Frequency Division Multiple Access (FDMA) was used in which every user occupied 30KHz from the total bandwidth. The design of this channel access method, which was used in the first generation networks (1G), is very simple, synchronization is easy and there is no interference among users in a cell. However there is narrowband interference, static spectrum allocation and frequency reuse. Furthermore, the need for high Q analog filters was one more reason to find a more efficient method.

Time Division Multiple Access (TDMA) is used in second generation (2G) networks and in this method each user shares a 30KHz channel with three other users. Each user gets a time slot, so this method is better suited for digital communications and provides higher capacity. However the need for better quality of service and even higher capacity led to the implementation of Code Division Multiple Access method or CDMA.

According to this method, each group of users is given a shared code (pseudorandom sequence). Many codes occupy the same channel, but only users associated
with a particular code can communicate. A receiver cannot demodulate the transmission without knowledge of the pseudorandom sequence that was used to encode the data.

To sum up, CDMA employs spread-spectrum technology and a special coding scheme (where each transmitter is assigned a code) to allow multiple users to be multiplexed over the same physical channel. More information about CDMA can be found on [Vit95]. In this report we will discuss about this spread-spectrum technology which is also used in the IEEE 802.15.4 protocol [80206] that our motes communicate with.

### 2.5.1 Spreading

As we have already mentioned, spread-spectrum techniques are methods by which a signal generated in a particular bandwidth is deliberately spread in the frequency domain, resulting in a signal with wider bandwidth [KRL95]. Spread spectrum communication technology has been used in military communications for over half a century, primarily for two purposes: to overcome the effects of strong intentional interference (jamming), and to hide the signal from eavesdroppers (covertness). Both goals can be achieved by spreading the signal’s spectrum to make it virtually indistinguishable from background noise.

Spread spectrum generally makes use of a sequential noise-like signal structure to spread the normally narrowband information signal over a relatively wideband (radio) band of frequencies. The receiver correlates the received signals to retrieve the original information signal. A scheme of the method is shown in figure 2.3.

Frequency-Hopping Spread Spectrum (FHSS), Direct-Sequence Spread Spectrum (DSSS) which is also used in IEEE 802.15.4, Time-Hopping Spread Spectrum (THSS), Chirp Spread Spectrum (CSS), and combinations of these techniques are forms of spread spectrum.
Figure 2.4: Spreading operation spreads the signal energy over a wider frequency bandwidth

**Theoretical justification**

In information theory, the Shannon-Hartley theorem gives the maximum rate at which information can be transmitted over a communications channel of a specified bandwidth in the presence of noise. The theorem establishes Shannon’s channel capacity for such a communication link, a bound on the maximum amount of error-free digital data that can be transmitted with a specified bandwidth in the presence of the noise interference, assuming that the signal power is bounded, and that the Gaussian noise process is characterized by a known power or power spectral density. According to the Shannon-Hartley channel-capacity theorem

\[
C = B \log_2 \left(1 + \frac{S}{N}\right),
\]

(2.1)

where \(C\) is the channel capacity in bits per second, \(B\) is the bandwidth of the channel in hertz, \(S\) is the average received signal power over the bandwidth, measured in watts, \(N\) is the average noise or interference power over the bandwidth, measured in watts and \(S/N\) is the signal-to-noise ratio (SNR) of the communication signal to the Gaussian noise interference expressed as a linear power ratio.

There is an elegant interpretation of this equation, applicable for difficult environments, for example, when a low \(S/N\) ratio is caused by noise and interference. This approach says that one can maintain or even increase communication performance (high \(C\)) by allowing or injecting more bandwidth (high \(B\)), even when signal power is below the noise floor. We can modify equation 2.1

\[
\frac{C}{B} = \left(\frac{1}{\ln 2}\right) \ln \left(1 + \frac{S}{N}\right) = 1.443 \times \ln \left(1 + \frac{S}{N}\right).
\]

(2.2)
By applying the MacLaurin series development

\[
\ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots + \frac{(-1)^{k+1} x^k}{k} + \cdots, \tag{2.3}
\]

we get

\[
\frac{C}{B} = 1.443 \left( \frac{S}{N} - \frac{1}{2} \left( \frac{S}{N} \right)^2 + \frac{1}{3} \left( \frac{S}{N} \right)^3 - \cdots \right), \tag{2.4}
\]

\(S/N\) is usually low for spread-spectrum applications. Assuming a noise level such that \(SNR << 1\), equation 2.4 becomes

\[
\frac{C}{B} = 1.443 \frac{S}{N}. \tag{2.5}
\]

Therefore in order to send error-free information for a given noise-to-signal ratio in the channel, one needs only to perform the fundamental spread-spectrum signal-spreading operation and increase the transmitted bandwidth. The result of the spreading operation is illustrated in figure 2.4.

2.5.2 Benefits of spread-spectrum

Resistance to Interference

Resistance to interference is the most important advantage in spreading. Only the signals that contain the spread-spectrum key are despreaded properly and all the rest, which are intentional or unintentional interference and jamming signals are rejected. That rejection also applies to other spread-spectrum signals that do not have the right key. Thus different spread-spectrum communications can be active simultaneously in the same band, such as CDMA. The resistance to interference in spreading is illustrated in figure 2.5.

Resistance to Interception

Another advantage provided by the spreading techniques is the resistance to interception. As we said in 2.5.2, non-authorized listeners cannot decode the original signal because they do not have the proper key. Without the right key, the spreaded signal appears as noise to all the other listeners and due to the reduction of the spectral density, the signal levels are even below the noise floor. Thus, the original data are invisible. Figure 2.6 illustrates this benefit.

2.5.3 Direct-Sequence Spread Spectrum (DSSS)

The Direct-Sequence Spread Spectrum is one of the most popular modulation techniques among spread-spectrum technologies and the technique that IEEE 802.15.4
uses. With the DSSS technique, the pseudorandom sequence is applied directly to data entering the carrier modulator.

The pseudorandom code symbols are called "chips", each of which has a much shorter duration than an information bit. That is, each information bit is modulated by a sequence of much faster chips. Therefore, the chip rate is much higher than the information signal bit rate and the modulator sees a much larger bit rate, which corresponds to the chip rate of the pseudorandom sequence. Figure 2.7 shows the processing of the initial data before the mixer.

Modulating an RF carrier with such a code sequence produces a direct-sequence-modulated spread spectrum with \((\sin x/x)^2\) frequency spectrum, centered at the carrier frequency.

The main lobe of this spectrum has a bandwidth twice the clock rate of the
modulating code, and the side lobes have bandwidths equal to the code’s clock rate. Direct-sequence spectra vary somewhat in spectral shape, depending on the actual carrier and data modulation used.

Finally, the receiver can then use the same pseudorandom sequence in order to reconstruct the information signal. The most common type of direct-sequence-modulated spread-spectrum signal is illustrated in figure 2.8.

### 2.5.4 Chirp Spread Spectrum (CSS)

Chirp Spread Spectrum is another interesting spread-spectrum technique that uses wideband linear frequency modulated chirp pulses to encode information. It is worth mentioning because it combines DSSS and UWB strengths. A chirp is a sinusoidal signal whose frequency increases or decreases over a certain amount of time.

As with other spread spectrum methods, CSS uses its entire allocated bandwidth
to broadcast a signal, making it robust to channel noise. Furthermore, this technique is also resistant to multi-path fading even when operating at very low power, because the chirps utilize a broad band of the spectrum. However, unlike direct-sequence spread spectrum (DSSS) it does not add any pseudo-random elements to the signal to help distinguish it from noise on the channel, but it relies instead on the linear nature of the chirp pulse. Additionally, Chirp Spread Spectrum is resistant to the Doppler effect.

2.6 Security

Wireless sensor networks for building automation aim to improve control and management of the technical infrastructure of a building. In that way they can increase user comfort, reduce operational and maintenance costs, provide security and safety.

Each node is a valuable equipment for the whole system since it can influence the decision of the fusion center. The fusion center could take some wrong decisions that could cause unpredictable situations depending on the application.

From this point of view, security is of utmost importance for every wireless sensor network for building automation. The cost of applying security to a network system is a lower bound on packet size for the encryption and the computations needed for the encryption/decryption scheme.

According to [PK05] secrecy can be achieved by obscurity or security. Obscurity means that it will take some effort to get the information, and one hopes that no one really wants to pay the cost. So the important issue when designing the security of a wireless sensor network is to define the cost of the information which depends upon the application. If the information obtained costs more than its value, then the network is considered secure enough.

2.6.1 Ways to attack

In order to know how to secure a system you should first identify the different ways that an adversary can attack. The goal of an adversary is to have access to system’s functionality, modify the transmitted valuable information or cause problems to the normal operation of the system. There are two main types of attacks:

**External** is the characterization of an attack that does not affect the network but, as an eavesdropper would do, hears all the transmitted packets and tries to understand its content [HC03]. The adversary uses just an antenna inside the range of the wireless sensor network in order to collect all the data.
CHAPTER 2. WIRELESS SENSOR NETWORKS

Internal is the kind of attack that the adversary uses equipment similar with the one that the original network has, so as to modify packets, drop packets and in general affect the normal operation of the network.

Attack analysis

Network attacks are the kind of attacks that interfere with data during the original transmission or control data. A network attack can be

- Denial-of-Service (DOS) attack
- Insert malicious data
- Modify original data

Device attacks are the ones that attack a device to gain access to the original data that the device is controlling. A device attack can be

- Software attack, by exploiting weaknesses in software implementation
- Physical attack, by manipulating the hardware of the device
- Side-channel attack, in which the adversary observes external parameters in order to collect information

Now that we have analyze the different kind of attacks the requirements analysis for the protection of a sensor network becomes clearer. In the next section there are description of methods that can offer obscurity to a wireless sensor network.

2.6.2 Ways to secure

Encryption is one of the most powerful ways to secure a network. By using encryption, the transmitted data packets become "invisible" to other listeners because they can not understand the language that is used. The data packets are encrypted with a key from the transmitting device and decrypted by the receiver. In that way the packets are protected while they are traveling to an open medium like air.

There are many different types of encryption methods and they can be divided by two criteria. The first criterion is whether they operate on a message block of some fixed size (block cipher) or on a continuous stream of data (stream cipher). However the second criterion is more important and is about the type of key used. There are two types keys, the secret and the public key. In a secret key system (symmetric) each node has its own key that is private. On the other hand, in a public key system (asymmetric) each node has both its own private key and a public key that is known to every other node.
Secret key algorithms  Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are secret key algorithms. In DES the data are first permuted using a permutation P. The message is divided into two 32-bit parts and sixteen rounds of computations are performed. Each of the computations has a different 48-bit key derived from the original key. Decryption uses the same method run in reverse order. AES is a more advanced algorithm that has replaced DES. The problem with secret key systems is the distribution of the keys. That problem can be solved with public key systems with the cost of more computations.

Public key algorithms  The most interesting public key algorithm is RSA. The basic parameter computation procedure is as follows:

- Generate prime numbers \( p \) and \( q \) and compute \( n = pq \)
- Let \( \phi = (p - 1)(q - 1) \)
- Choose integer \( e, 1 < e < \phi \), such that the greatest common divisor of \( e \) and \( \phi \) is 1
- Compute a secret exponent \( d, 1 < d < \phi \), such that \( ed = 1 \mod \phi \)
- \( p, q \) and \( \phi \) are kept secret
- The public key is \( (n, e) \) while the secret key is \( (n, d) \)
- Encryption is the computation \( y = x^e \mod n \); decryption is \( x = y^d \mod n \)

In that way we can secure the transmission without facing the problem of securely distributing keys. But what about all the other kind of attacks? In security there is a solution for every problem that an adversary creates. The difficulty is to prevent those actions. In order to encounter the network attacks a network should be built in a way that identifies fake data packets and ignores them. Reliability as explained in section 2.6.5 is one way to identify fake or problematic nodes.

Furthermore, the architecture of the network should be such that each node support each other in the Denial-of-Service attack. If the security of a network in a building automation system has all those features then the system should be considered secure enough. However there is no system that a stubborn adversary cannot break in and the user should always have that in mind.
2.6.3 Scrambler

Scrambler is a device that pseudo-randomly changes the values of some bits in a data block or stream by using a Pseudo-Random Sequence Generator and a modulo-2 adder (XOR). In that way it encodes a message at the transmitter, so as to make it unintelligible at a receiver which is not equipped with the suitable descrambling device [Ped08].

Furthermore, a scrambler can be used for the “whitening” of the spectrum of a data block or a stream. Consequently the spectrum is spread so that no strong spectral component will exist, thus reducing electromagnetic interference. The pseudo-randomness is usually accomplished using an Linear-Feedback Shift Register (LFSR). The different types of scramblers and their functionality are explained in the followings.

Linear Feedback Shift Register

As it is already mentioned, LFSRs are the most essential parts of a scrambler and descrambler. They are used as pseudo-random sequence generators in order to generate pseudo-random numbers, pseudo-noise sequences, fast digital counters or whitening sequences. The mathematics of a cyclic redundancy check, used to provide a quick check against transmission errors, are closely related to those of an LFSR.

In a standard LFSR configuration the bit positions that affect the next state are called the taps. In figure 2.9 the taps are 3 and 4. The rightmost bit of the LFSR is called the output bit. The taps are XOR’d sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream.

A maximum-length LFSR produces a m-sequence which means that it cycles through all possible $2^n - 1$ states within the shift register except for the state where all bits are zero, in which case it will never change.
2.6. SECURITY

Figure 2.10: A 4-bit additive scrambler with polynomial $x^4 + x^3 + 1$

The arrangement of taps for feedback in a standard LFSR can be expressed in a finite field arithmetic as a polynomial mod 2. This means that the coefficients of the polynomial must be 1’s or 0’s. This is called the feedback or characteristic polynomial. For example, if the taps are at the 3rd and 4th bits (as shown in figure 2.9), the feedback polynomial is

$$x^4 + x^3 + 1. \quad (2.6)$$

The ‘one’ in the polynomial does not correspond to a tap, but to the input to the first bit (i.e. $x^0$, which is equivalent to 1). The power of the terms represent the tapped bits, counting from the left. The first and last bits are always connected as an input and output tap respectively.

**Scramblers and Descramblers**

There are two types of LFSR-based scramblers, called additive and multiplicative scramblers. Both are described below, along with their corresponding descramblers.

**Additive scrambler** In the Additive or Synchronous scramblers and descramblers, the LFSR is connected to the data stream by means of just an additional modulo-2 adder which is a simple XOR gate. A device of this type is shown in figure 2.10. In this device $x(n)$ represents the data to be scrambled at time $n$, $k(n)$ represents the "key" produced by the LFSR, and $c(n)$ represents the scrambled codeword. A similar device is presented on figure 2.11 which is the corresponding descrambler.

The device of the figure 2.10 causes the value of a bit in the main data stream to be flipped when the LFSR produces a ‘1’. Therefore, if this device is synchronized with the one in figure 2.11 it will cause the same bit to flip again, hence returning them to their original values.
A mathematical explanation of the scrambling/descrambling logic for the recovery of $x(n)$ can be shown as follows. At time $n$,

$$c(n) = x(n) \oplus k(n).$$

(2.7)

Furthermore,

$$k(n) = k(n-3) \oplus k(n-4),$$

(2.8)

thus

$$c(n) = x(n) \oplus k(n-3) \oplus k(n-4),$$

(2.9)

which is the scrambled codeword and the input for the descrambler. At the descrambler side we have,

$$y(n) = c(n) \oplus k(n).$$

(2.10)

Assuming that the two circuits are synchronized,

$$y(n) = c(n) \oplus k(n-3) \oplus k(n-4),$$

(2.11)

and finally

$$y(n) = x(n) \oplus k(n-3) \oplus k(n-4) \oplus k(n-3) \oplus k(n-4).$$

(2.12)

Moreover, we know that $A \oplus A = 0$ and that $A \oplus 0 = A$, so we get the expected result which is $y(n) = x(n)$.

The disadvantage of this approach is that synchronism is required, in which both ends must start from the same initial state. In practice this can be achieved by sending a sequence of known symbols. For example, in the Ethernet 100Base-TX interface, where a similar device is employed, synchronism occurs after a sequence of about 20 idle symbols are sent.
Multiplicative scrambler  A slightly different approach is the multiplicative or asynchronous scrambler/descrambler. This pair, which is presented in figures 2.12 and 2.13, is self-synchronizing, meaning that they do not need to start from the same initial state like in the additive approach. However, this self-synchronization process might take up to N bits (clock cycles), so the first N values of y(n) should be discarded.

The mathematical explanation is as follows. At time n,

\[ c(n) = x(n) \oplus k(n). \]  
(2.13)

Furthermore,

\[ k(n) = c(n - 3) \oplus c(n - 4), \]  
(2.14)

thus

\[ c(n) = x(n) \oplus c(n - 3) \oplus c(n - 4), \]  
(2.15)

which is the scrambled codeword and the input for the multiplicative descrambler. At the side of the descrambler we have,

\[ y(n) = c(n) \oplus k(n). \]  
(2.16)

Because

\[ k(n) = c(n - 3) \oplus c(n - 4), \]  
(2.17)

we get

\[ y(n) = c(n) \oplus k(n - 3) \oplus k(n - 4), \]  
(2.18)

and finally

\[ y(n) = x(n) \oplus c(n - 3) \oplus c(n - 4) \oplus c(n - 3) \oplus c(n - 4) = x(n). \]  
(2.19)

The disadvantage of this approach is that errors are multiplied by T + 1, where T is the number of taps. Therefore, if one bit is flipped by noise in the channel during
transmission, 3 bits will be wrong after descrambling with the device of figure 2.13. It is important to mention also that when the errors are less than N bits apart, less than T + 1 errors per incorrect bit might result because the superposition of errors might cause some of the bits to be unintentionally corrected.

**Example** In this paragraph we will show an example of this technique to understand the way that multiplicative scramblers work and to show their limitations. We will use the scrambler and descrambler pair of figure 2.12 and 2.13 respectively. Let’s assume that the scrambler starts from an initial state of ”0101” and the descrambler from ”1010” in order to show the asynchronous function.

In the beginning, the scrambler will process the data stream ”1001010010”. Table 2.1 shows the functionality step by step. The final result of the scrambling is ”0111110011”. This will be unintelligible for every receiver that has not a suitable descrambling device. In the following table we show step by step the procedure of
### 2.6. SECURITY

<table>
<thead>
<tr>
<th>Step</th>
<th>bits</th>
<th>k</th>
<th>c</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1010</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0101</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1010</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1101</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1110</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1111</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0011</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1001</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.2: The final result after the descrambling procedure

As we can see from Table 2.2, the final result is equal with the initial data stream except for the first N=4 values which is due to the asynchronous nature of the multiplicative descrambler as we explained earlier.

#### 2.6.4 Security in CC2420

The nodes that were used in this project are TelosB from Memsic (old Crossbow) technology. This mote has the same design as the Tmote Sky mote from Sentilla. The main blocks of hardware on this board are the MSP430 (the MSP430F1611) microcontroller and the CC2420 radio chip. The microcontroller of this mote operates at 4.15 MHz and has a 10 kBytes internal RAM and a 48 kBytes program Flash memory.

IEEE 802.15.4 media access control layer defines different security modes based on the AES block cipher:

- Null: neither encryption nor authentication. Default mode.
- Counter mode (CTR): Encryption only. No authentication.
- Cipher Block Chaining Message Authentication Code (CBC-MAC): Authentication only.
- Counter with CBC-MAC (CCM): Encryption and authentication.

For the last two modes, the authentication tag can be 32, 64 or 128 bits. The CC2420 radio chip supports three types of in-line security modes, leveraging the same underlying 128-bit AES encryption:
• Counter Mode Encryption (CTR)
• Cipher Block Chaining Message Authentication Code (CBC-MAC)
• Counter with CBC-MAC (CCM).

Although the encryption provided by the radio chip is strong enough for WSNs, it consumes a lot of power thus it is not suitable for power efficient applications.

2.6.5 Data Reliability

After security is applied in a network someone would believe that the received data packets will be original and unmodified. However there are several problems that could cause the reception of wrong results. A broken node, an adversary that managed to take over a node, radio problems etc are some of the many problems that a wireless network could encounter.

Data reliability starts with the calibration of all nodes in order to know in the beginning that all the nodes have specific observations under specific circumstances. After that the system should evaluate constantly the observations coming from every node. If there are unrealistic observations from a node then this node should be suspected for providing unreliable results and reported to the owner of the network.

There aren’t many automated methods that can be applied if there is such a problem in a node. The network could have a reputation system in which each node is assigned with a weight according to their reliability. However there is no such thing as an absolute guarantee of reliability. Detection and estimation could increase reliability as explained in the following chapters.

2.6.6 Conclusion

As it is argued before, a network cannot be ever completely secure. It depends on the effort of the adversary to find ways to infiltrate into the network. The security design of a network should be done according to the application and the cost of the information. AES and RSA algorithms provide enough security by applying encryption techniques on the transmitted data packets. TelosB platform uses the CC2420 RF chip which supports AES encryption. This together with detection and estimation techniques that increase the data reliability should make secure enough a wireless sensor network for building automation.

2.7 Localization

Localization is a technique where the nodes of a WSN communicate with each other in order to locate themselves in the grid where the WSN is placed in. The
information on the position and orientation of the nodes is useful for the efficient routing in mesh networks, for automated self-configuration, for better controlling of the WSN and finally for the visualization of the position of the nodes on the graphical user interface.

There are two types of localization algorithms. The first type is the \textit{centralized} localization where the data are collected at a fusion center and the second one is the \textit{distributed} localization, where each node of the WSN is able to localize itself in the grid.

A localization problem is consisted of three parts

- \textbf{Ranging}, to determine the distances among the nodes;
- \textbf{Positioning}, to find the approximate position of the nodes;
- \textbf{Refinement}, to increase the reliability of the final result.

In the following sections there is a description of these parts.

\subsection*{2.7.1 Ranging}

The most common \textit{ranging} technique that measures relative distance among nodes is the Received Signal Strength Indicator (RSSI). RSSI is the voltage measured by the radio chip’s RSSI circuit. It is the simplest technique because it does not require additional bandwidth or energy.

It is apparent that the RSSI technique is quite simple, but it produces very noisy measurements because of the shadowing and fading phenomena. For instance, an obstacle can cause the signal to lose power on its way to the receiver’s antenna. This is causing a wrong measurement which, however, could be refined in the refinement process.

Another ranging technique is the \textit{Time-of-Arrival} in which the relative distance is measured by using the signal propagation delay. This technique is more accurate and less noisy than the RSSI, but the nodes have to be accurately synchronized. Of course the synchronization is a very difficult task, especially in low-cost hardware platforms.

As it is obvious, the technique used in this thesis is RSSI since the hardware is not capable to have accurate synchronization. However, accuracy in localization was not a major goal of this work. Nevertheless, with refinement it is possible to minimize the estimation error variance if needed.

Finally, there are many more advanced techniques for accurate estimation of the position of network nodes. In [AFH11], the estimator combines heterogeneous information coming from pre-existing ranging, speed, and angular measurements, which is jointly fused by an optimization problem where the squared mean and
variance of the localization error is minimized. In [AF11] in order to achieve a high accuracy, the fusion of heterogeneous sensor information is used. The method does not assume any motion model of mobile nodes and is based on a Pareto optimization.

2.7.2 Positioning

In the positioning part there are algorithms that find an approximation of the position of every node. There are many methods for this task, but the most interesting methods are the following

- **Min Max Algorithm**, which locates a node by using squares with sides equal to twice the estimated distances among nodes.

- **Lateration**, which locates a node by using overlapping circles with diameters equal to half the estimated distances among nodes.

- **Ring Overlapping Algorithm**, which locates a node by using overlapping regions drawn from circles with diameters equal to half the estimated distances among nodes.

All of the above techniques for positioning are low cost and complexity and require RSSI as a ranging technique. There should be an *a priori* knowledge on where are the nodes in the beginning of the operation. However there are positioning techniques for the cases in which the initial position of the nodes is uncertain, like

- Global positioning system (GPS)
- Ad Hoc Localization systems
- Ad Hoc Positioning System (APS)
- Multi-Dimensional Scaling
- Optimization

The above techniques are more complex, but provide results even if the initial position of the nodes is unknown to the system.

2.7.3 Refinement

The refinement part of the localization problem tries to minimize the error of the positioning. The most commonly [Mas07] used refinement methods are the following

- Bayesian Estimators
2.7. LOCALIZATION

- Optimization of the cost function [Fis11]
- Filtering, like Kalman filter or Monte Carlo estimation
- Cooperative localization

More information on refinement can be found in [SFJSV08] and [SFJ06].

2.7.4 Conclusion

To sum up, localization is a very interesting area in which a WSN can locate all of its nodes in order to increase its functionality. RSSI is the best solution for low-cost WSN, but it is noisy due to shadowing and fading phenomena. For the positioning phase there are many algorithms that are separated into two categories. The first category is when there is knowledge on the initial position and the second category when the initial position of the nodes is unknown. The refinement process helps to reduce the error caused by the previous phases.

In this thesis only the first step of a centralized localization has been made. Since star network topology is used, the gateway measures the distance from every node and the result is displayed on the graphical user interface. However, as a future work, it would be interesting to continue with the other phases too in order to complete the localization process.
Chapter 3

Detection and Estimation

Any unwanted random addition to a signal is called noise. Noise is present everywhere around us and it affects vastly sensor measurements and wireless transmissions. There are many techniques that are trying to eliminate noise in every level of a wireless transmission, however due to noise’s stochastic nature, there is not any completely efficient procedure. For that reason detection and estimation can help to determine the real state of the environment by taking advantage of the domain knowledge.

In section 3.1 there is a discussion about the most common detection techniques. In the beginning there is an introduction to detection, later a brief analysis of the fusion techniques and finally the methods used when there is uncertainty in the prior probabilities. In section 3.2 an analysis of the estimation using the Kalman filter is discussed both for the sequential measurements from one sensor and for the case of dynamic fusion of measurements from multiple sensors. Finally, there is a discussion about the detection and estimation techniques in section 3.3.

3.1 Detection

3.1.1 Introduction to detection

It is important for building automation systems to be reliable. A false or a missed detection may cost a lot, so we have to be completely sure for each decision. In detection problems the objective is to distinguish between some finite number of possibilities making the best decision in the presence of uncertainty and any prior knowledge or notions of the cost of the various decisions. In a group decision making system, multiple sensors observe a common phenomenon [CV07]. Sensors can act as simple data collectors, in which case the system is centralized, or they can be provided with processing capabilities in order to transmit a compressed version
CHAPTER 3. DETECTION AND ESTIMATION

Figure 3.1: Parallel topology with fusion center

of their data to a fusion center. This kind of system is called a decentralized or
distributed system.

There are three major topologies used for distributed signal processing [VV97].
These are called parallel, serial and tree configurations. In figure 3.1 is illustrated
the parallel configuration of \( N \) sensors. To apply the detection theory in a simpler
way, we assume that the sensors do not communicate with each other and that there
is no feedback from the fusion center to any sensor.

In binary detection problems there are two hypotheses:

- \( H_0 \): desired signal absent;
- \( H_1 \): desired signal present.

According to the parallel configuration shown in figure 3.1, \( y_i \) denotes an observa-
tion that is available at the \( i \)th sensor \( S_i \). The \( i \)th sensor employs the mapping
3.1. DETECTION

rule \( u_i = \gamma_i(y_i) \) and passes the quantized information \( u_i \) to the fusion center. Based on the received information from all \( N \) sensors \( \mathbf{u} = (u_1, u_2, \ldots, u_N) \), the fusion center makes the global decision \( u_0 = \gamma_0(\mathbf{u}) \) that decides either \( H_1 \) or \( H_0 \).

Several criteria are used in order to make decisions about whether the signal is present \( (u_i = 1) \) or not \( (u_i = 0) \). For instance a decision can be made according to which of the conditional probabilities is higher. This is the maximum \textit{a posteriori} (MAP) criterion and it maximizes probabilities that can only be known after the fact of a set of observations. In this criterion the algorithm chooses \( H_1 \) if \( P(u_1|z) > P(u_0|z) \) and \( H_0 \) otherwise, where \( z \) is the probability for the event to happen.

Another useful criterion in detection is to try minimize the Bayes risk function, \( \mathcal{R} \), which is given by

\[
\mathcal{R} = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P_j P(\text{Decide } H_i \mid H_j \text{ is present}). \tag{3.1}
\]

In this formula we assign costs to the probabilities, \( C_{ij} \) which represent the cost of declaring \( H_i \) when \( H_j \) is present. The minimization of the risk function can result in the likelihood ratio test (LRT)

\[
\begin{align*}
\frac{p(u|H_1)}{p(u|H_0)} &> \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}, \\
& \quad H_1 \\
\frac{p(u|H_0)}{p(u|H_0)} &< \frac{P_1(C_{01} - C_{11})}{P_0(C_{10} - C_{00})}, \\
& \quad H_0
\end{align*} \tag{3.2}
\]

where \( P_i, i = 0, 1 \) is the prior probability for the event to happen or not.

There are two different possibilities for the observation \( y \) which yield to four different conditional probabilities. The conditional probability that there is no signal and the observation is correct is represented by \( P(u = 0|H_0) \). On the other hand if there is no signal but the observation is positive, then we get a false alarm and the probability is \( P(u = 1|H_0) = P_F \). If there is a signal but the system missed it we get the probability of miss \( P(u = 0|H_1) = P_M \) and when the system detects correctly the signal we get the probability of detection which is \( P(u = 1|H_1) = P_D \).

The expression for the minimum achievable probability of error by the optimum Bayesian detection system it is known as the Kolmogorov variational distance and is given by

\[
\mathcal{R}_{\min} = \frac{1}{2} - \frac{1}{2} \int_Z |P_1 p(u|H_1) - P_0 p(u|H_0)| du. \tag{3.3}
\]

In that way the cost of false detection and miss is minimized and thus the results are more reliable.
3.1.2 Optimal data fusion

Using the above methods each node can reach an optimal decision according to some probability rules. However, sometimes a malfunction or a strong disturbance could yield to a wrong decision. By using multiple sensors it is possible to make a comparison between the different sources and to make a global decision about the state of the environment. There are many methods that could be used in order to achieve a data fusion. Data fusion rules are often implemented as 'k out of n' logical functions. This means that if $k$ or more nodes decide the hypothesis $H_1$, then the global decision is $H_1$.

$$u = \begin{cases} +1, & \text{if } u_1 + u_2 + \cdots + u_n \geq 2k - n \\ -1, & \text{otherwise.} \end{cases}$$ (3.4)

However, in this section we consider a more general formulation of the data fusion problem. We will focus into an optimal data fusion technique presented in the paper of [CV86]. According to this paper, the data fusion problem can be viewed as a two-hypothesis detection problem with individual detector decisions being the observations. As we mentioned in subsection 3.1.1, the optimum decision rule is given by the likelihood ratio test. If we assume the minimum probability of error criterion, that is, $C_{00} = C_{11} = 0$, and $C_{01} = C_{10} = 1$ and use the Bayes rule to express the conditional probabilities we obtain

$$\frac{p(H_1|u)}{p(H_0|u)} > 1.$$ (3.5)

By taking the log-likelihood ratio test and expanding the above formula we can express the data fusion rule as

$$f(u_1, u_2, \ldots, u_n) = \begin{cases} +1, & \text{if } a_0 + \sum_{i=1}^{n} a_i u_i > 0 \\ -1, & \text{otherwise} \end{cases}$$ (3.6)

where the optimum weights are given by

$$a_0 = \log \frac{P_1}{F_0},$$ (3.7)

$$a_i = \log \frac{1 - P_{M_i}}{P_{F_i}}, \text{ if } u_i = +1$$ (3.8)

and

$$a_i = \log \frac{1 - P_{F_i}}{P_{M_i}}, \text{ if } u_i = -1$$ (3.9)
3.1. DETECTION

In that way we can easily get an optimal fusion rule with the help of all nodes. Each of the \( n \) nodes will send to the fusion center \( u_i \), which is the local decision and the fusion center will just have to solve a simple equation (3.6) in order to reach the optimal result.

3.1.3 The case of uncertain prior probabilities

The Bayes criterion for the design of decision rules requires the knowledge of the \textit{a priori} probabilities \( P_0 \) and \( P_1 \), which, depending on the system, may not be available. In such situations there are feasible alternatives that could solve this particular problem, the minimax criterion and the Neyman-Pearson Test [Var96].

**Minimax Detection**

In the minimax criterion, one uses the Bayes decision rule corresponding to the least favorable prior probability assignment. We can rewrite the Bayes risk as a function of the prior probability \( P_1 = 1 - P_0 \) and the decision regions through \( P_M \) and \( P_F \). In this way we can obtain

\[
\mathcal{R} = C_{00}(1 - P_F) + C_{10}P_F + P_1[(C_{11} - C_{00})+(C_{01} - C_{11})P_M - (C_{10} - C_{00})P_F]. \quad (3.10)
\]

For a fixed cost assignment, the optimum Bayes threshold and the risk \( \mathcal{R}_{\text{opt}} \) vary with \( P_1 \). The risk \( \mathcal{R}_{\text{opt}} \) for the optimal Bayes test is a continuous concave downward function of \( P_1 \). By fixing the \( P_1 \) to a value \( P_1' \) the Bayes risk \( \mathcal{R}' \) is a linear function of \( P_1 \) tangent to the optimum Bayes curve. So the optimum Bayes test is when the value \( P_1' \) makes the tangent horizontal. In that way the derivation of the Bayes risk function in regard to \( P_1 \) should give

\[
(C_{11} - C_{00}) + (C_{01} - C_{11})P_M - (C_{10} - C_{00})P_F = 0. \quad (3.11)
\]

This equation is known as the minimax equation and determines the test if the maximum of the risk is interior to the interval \((0, 1)\) and that \( \mathcal{R}_{\text{opt}} \) is differentiable everywhere.

**Neyman-Pearson Test**

In many practical situations there is uncertainty in both the prior probabilities and the cost assignments. In such cases the Neyman-Pearson Test is ideal for the solution of the detection problem. However in these cases there is a false alarm rate constraint. In general it is desirable to make \( P_F \) as small as possible and \( P_D \) large. The Neyman-Pearson test utilizes this need in order to give fixed values to \( P_F \) and \( P_D \) in order to make the decision. Under this formulation, it is important
to mention that the parallel fusion network topology without a fusion center is not appropriate because systemwide probabilities of detection and false alarm cannot be defined. Therefore a fusion center is always assumed to be present.

The fusion rule for the Neyman-Pearson test has the same structure as the fusion rule obtained in subsection 3.1.2 for the optimal data fusion except for now the prior probabilities are changed with a threshold $\Gamma$ which is the Lagrange multiplier. The Lagrange multiplier method was employed in order to solve the constraint optimization problem. The fusion rule is given by

$$
\sum_{j=1}^{N} \left[ u_j \log \frac{1 - P_{M_j}}{P_{F_j}} + (1 - u_j) \log \frac{P_{M_j}}{1 - P_{F_j}} \right] > \log \Gamma.
$$

(3.12)

The decision rule at the local detector $k$ is minimized when we set

$$
\begin{align*}
    u_k &= 1, \\
    \frac{p(y_k|H_1)}{p(y_k|H_0)} &= \frac{t_k}{u_k} = 0
\end{align*}
$$

(3.13)

where

$$
t_k = \Gamma \frac{C^k}{C_i^k},
$$

(3.14)

and

$$
C_i^k = \sum_{u^k} [P(u_0 = 0|U_k = 0, u^k) - P(u_0 = 0|u_k = 1, u^k)]P(u^k|H_i).
$$

(3.15)

This set of equations can be solved simultaneously under the constraint of $P_F$ to yield the optimal solution.

### 3.1.4 Detection cases depending on the nature of the system

Before making the decision on which detection technique to use for one system, one should first examine all the possible solutions according to the nature of the system. In this section a separation between systems that observe rare events and systems that observe frequent events is presented.

#### Case of rare events

In the case of rare events it is safe to suppose that the probability of the event to happen is very low. On the other hand the probability to have absence of an event
3.1. DETECTION

is very high. In that way, the prior probabilities can be estimated by assumption in order to apply the optimal data fusion.

The data fusion problem can be viewed as a two-hypothesis detection problem with individual detector decisions being the observations. The optimum decision rule is given by the likelihood ratio test. If we assume the minimum probability of error criterion, where the cost of success is zero, $C_{00} = C_{11} = 0$, and the cost of failure is one, $C_{01} = C_{10} = 1$ and use the Bayes rule to express the conditional probabilities, we obtain

$$
\frac{p(H_1|u)}{p(H_0|u)} > 1,
$$

(3.16)

By taking the log-likelihood ratio test and expanding the above formula we get

$$
\log \frac{P(H_1|u)}{P(H_0|u)} = \log \frac{P_1}{P_0} + \sum_{s_+} \log \frac{1 - P_{M_i}}{P_{F_i}} + \sum_{s_-} \log \frac{P_{M_i}}{1 - P_{F_i}} > 0, \quad H_1
$$

$$
\frac{H_0}{H_0}
$$

(3.17)

where $S_+$ is the set of all $i$ such that $u_i = +1$ and $S_-$ is the set of all $i$ such that $u_i = -1$.

Therefore, we can reach the data fusion rule as it was described in section 3.1.2 which is

$$
f(u_1, u_2, \ldots, u_n) = \begin{cases} 
+1, & \text{if } a_0 + \sum_{i=1}^{n} a_i u_i > 0 \\
-1, & \text{otherwise}
\end{cases}
$$

(3.18)

where the optimum weights are given by

$$
a_0 = \log \frac{P_1}{P_0},
$$

(3.19)

$$
a_i = \log \frac{1 - P_{M_i}}{P_{F_i}}, \text{ if } u_i = +1
$$

(3.20)

and

$$
a_i = \log \frac{1 - P_{F_i}}{P_{M_i}}, \text{ if } u_i = -1
$$

(3.21)

In order to be able to calculate the optimal fusion rule, we have to make an assumption on the prior probabilities. As it is already mentioned, in the case of rare events, we will have a very low $P_1$ and a $P_0$ close to 1. In that case it is safe to assume that $P_1 = 1/10000$ and $P_0 = 1 - P_1$. Furthermore, we already know that the probability of absence of the event $P_0$ plus the probability of false alarm is equal to
one $P_0 = 1 - P_F$. Finally, by using the Minimax criterion described in 3.1.3 we get the following formula which gives us the optimal solution to the detection problem

\[(C_{11} - C_{00}) + (C_{01} - C_{11})P_M - (C_{10} - C_{00})P_F = 0. \tag{3.22}\]

Moreover we have already assumed the minimum probability of error criterion, that is, $C_{00} = C_{11} = 0$, and $C_{01} = C_{10} = 1$. By applying these costs to the Minimax formula we get

\[P_M = P_F, \tag{3.23}\]

which is the last step to solve the fusion problem.

**Local miss ($P_{M_i}$) and false alarm ($P_{F_i}$) probabilities**  The local miss and false alarm probabilities need also to be established before applying the fusion rule for a specific system. For the most systems it is rational to assume that the probability of false or miss is the same for every mote, since those probabilities are depended on the operation of the sensor and the geometry of the system. In the case of the geometry, one could say that the mote should be calibrated in the beginning in order to have more reliable results. However, due to the stochastic nature of this characteristic, the calibration could be a difficult task and it would not provide trustworthy results. The same applies also to the case of the problematic sensor which could provide wrong results.

For these reasons a more complex solution could be applied in which each mote has a specific reliability level. In the beginning all nodes have a fixed miss and false alarm probability. In the case of $N$ nodes, where $N$ is supposed to be more or equal than 5, if less than the 20% of them have a different result than the majority, then their reliability decreases by 50%. If the same nodes present a result that is in agreement with all the motes in a later step, their reliability increases slower with 10% steps.

The fusion center is responsible for the calculation of the miss and false probabilities and informs the system and the user in a case of very low reliability results from a specific mote. In that way the system has memory and a problematic sensor, or a sensor that measures wrong values due to the geometry of the system can be ignored after a few steps since it cannot provide a reliable measurement. The system is auto-configured and the results are becoming more reliable in comparison to a system with fixed values for the miss and false alarm probabilities. Illustration of the algorithm for the calculation of the local estimates can be seen in figure 3.2.
3.1. DETECTION

Figure 3.2: Flowchart with the algorithm for the local estimates

Case of frequent events

In the case of frequent events, the prior probabilities cannot be estimated by assumption. Instead there are two other solutions that could yield reliable results that both are based on the Neyman-Pearson test as described in 3.1.3. The Neyman-Pearson test is suitable when the prior probabilities are unknown. In that case there is a constraint of the false alarm which is supposed to be smaller than a specified quantity.
CHAPTER 3. DETECTION AND ESTIMATION

Distributed Neyman-Pearson detection

The Neyman-Pearson test gives fixed values to $P_F$ and $P_D$ in order to make the decision in the hypothesis.

The probabilities $P_F$, $P_M$ and $P_D$ can be expressed in terms of $P_{F_i}$ and $P_{M_i}$ by defining

$$M_u = P(u|H_1) = \prod_{S_0} P_{M_i} \prod_{S_1} (1 - P_{M_k}),$$  \hspace{1cm} (3.24)

$$F_u = P(u|H_0) = \prod_{S_0} (1 - P_{F_i}) \prod_{S_1} P_{F_k},$$  \hspace{1cm} (3.25)

and

$$P_{iu} = P(u_i = i|u), \ i = 0, 1$$  \hspace{1cm} (3.26)

where $S_0$ is the set of all $i$ such that $u_i = 0$ and $S_1$ is the set of all $i$ such that $u_i = 1$.

Then we may express $P_F$ and $P_M$ as

$$P_M = \sum_u P_{0u} M_u,$$  \hspace{1cm} (3.27)

and

$$P_F = \sum_u P_{1u} F_u.$$  \hspace{1cm} (3.28)

The Lagrange multiplier method is employed in order to solve the constraint optimization problem. The objective function $F$ is

$$F = P_M + \Gamma (P_F - \alpha),$$  \hspace{1cm} (3.29)

where $\Gamma$ is the Lagrange multiplier. We can expand the above formula

$$F = \Gamma (1 - \alpha) + \sum_u P_{0u} [P(u|H_1) - \Gamma P(u|H_0)],$$  \hspace{1cm} (3.30)

$F$ is minimized if we choose the decision rule

$$P_{0u} = P(u_0 = 0|u) = \begin{cases} 0, & \text{if } P(u|H_1) - \Gamma P(u|H_0) > 0, \\ 1, & \text{otherwise} \end{cases}$$  \hspace{1cm} (3.31)

which can be rewritten as

$$\begin{align*}
  p(u|H_1) & > \Gamma, \\
  p(u|H_0) & < \Gamma, \\
  u_0 &= 0
\end{align*}$$  \hspace{1cm} (3.32)
By expanding the above formula, the fusion rule is given by

\[ \sum_{j=1}^{N} \left[ u_j \log \frac{1 - P_{M_j}}{P_{F_j}} + (1 - u_j) \log \frac{P_{M_j}}{1 - P_{F_j}} \right] u_0 = 1 \]

\[ > \log \Gamma. \quad (3.33) \]

The fusion rule for the Neyman-Pearson test has the same structure as the fusion rule obtained in section 3.1.4 for the optimal data fusion except for now the prior probabilities are changed with a threshold \( \Gamma \) which is the Lagrange multiplier.

The decision rule at the local detector \( k \) is minimized when we set

\[ p(y_k|H_1) > \gamma, \quad u_k = 1 \]

\[ p(y_k|H_0) < t_k, \quad u_k = 0 \]

\[ (3.34) \]

where

\[ t_k = \frac{C^k_0}{C^k_1}, \quad (3.35) \]

and

\[ C^k_i = \sum_{u^k} P(u_0 = 0|U_k = 0, u^k) - P(u_0 = 0|u_k = 1, u^k)P(u^k|H_i). \quad (3.36) \]

This set of equations can be solved simultaneously under the constraint of \( P_F \) to yield the optimal solution.

**Neyman-Pearson with Q-function**

The same criterion can be used in a different way by utilizing the Q-function [PK05] [Sco]. In this method the system needs to know the variance \( \sigma^2 \) of the signal and, together with the \( P_F \) constraint, it is easy to calculate the \( \gamma \) threshold which separates the two different cases of existence or not of a signal.

The hypothesis test of the measurement \( x \) is

\[ H_0 : x \sim N(\mu_0, \sigma_0^2), \quad (3.37) \]

\[ H_1 : x \sim N(\mu_1, \sigma_1^2), \quad (3.38) \]

where \( \mu \) is the mean value of the distributions and \( \sigma^2 \) the variance.

Suppose we have a threshold test

\[ H_1 \]

\[ x > \gamma, \quad (3.39) \]

\[ H_0 \]

\[ x < \gamma, \]

\[ H_0 \]
where $\gamma \in \mathbb{R}$ is a free parameter. The false alarm and detection probabilities are

$$P_F = \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_0)^2}{\sigma_0^2}} \, dx = Q\left(\frac{\gamma - \mu_0}{\sigma_0}\right),$$  \hspace{1cm} (3.40)

$$P_D = \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{\sigma_1^2}} \, dx = Q\left(\frac{\gamma - \mu_1}{\sigma_1}\right),$$  \hspace{1cm} (3.41)

where $Q$ denotes the Q-function.

In that way we can easily find the optimal threshold without knowledge of the prior probabilities. Of course this technique requires to know the mean value and the variance of the probability density function. In figure 1 the two probability density functions of $H_0$ and $H_1$ are visible.

The first step to solve this problem is to estimate the mean value and the variance for every distribution. This can be done through calibration by sending known values to the fusion center. In that way it is possible to find the mean value and the variance since we already know the value that has been sent. After that it is easy to calculate the Q-function for each distribution and thus the threshold $\gamma$ of the system.

**Conclusions**

From the analysis above we can understand that the detection technique that is going to be used for the system is depended on the frequency of the events. In our particular system the frequency of the events is low, so all the techniques could be used in order to achieve the desired result.

Figure 3.3: Probability density functions of the two hypotheses and the $\gamma$ threshold
3.2. ESTIMATION

By examining the different methods on the theoretical point of view, it is obvious that the easiest method is the case of rare events, since we have already made the assumption on the prior probabilities which was a limiting factor. However, the system becomes less reliable since their is no prior knowledge on if the assumption can be close to the real ones.

On the other hand, in the case of more frequent events, the system has first to calibrate in order to estimate the variance and mean values of the probability density functions. Unfortunately that means more effort from the motes and from the fusion center which will affect the power efficiency of the system.

3.2 Estimation

In estimation problems the objective is to make decisions on the values of the parameters. The parameters $a$ reside in the parameter space. The observables are represented by $z$ and the decisions is made using an estimation rule $\hat{a}(z)$. The estimation error is $e = a - \hat{a}(z)$. A cost function can be assigned to the errors as $C(a, \hat{a})$, which is the cost for accepting $\hat{a}(z)$ for $a$. The most popular cost function for estimation problems is the squared error, $(a - \hat{a})^2$ and its vector analog, the squared Euclidean norm

$$\|a - \hat{\alpha}\|^2 = (a_1 - \hat{\alpha}_1)^2 + \cdots + (a_n - \hat{\alpha}_n)^2.$$  \hspace{1cm} (3.42)

By minimizing the expected cost

$$E[C(a, \hat{a})] = \int E[C(a, \hat{a})|z]f(z)dz,$$  \hspace{1cm} (3.43)

we can solve the Bayes estimation problem.

The minimization of the expected cost is a technique that can find in an easy way the desired result, however is not optimal. The Kalman filtering as we will see in the following subsection provides better results.

3.2.1 Kalman filtering

A powerful technique for estimation problems is the Kalman filtering [WB01]. In this technique one can reach an estimation of the true value in relatively few steps with an iteration algorithm. A simplification of the Kalman filter is the following equation

$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1},$$  \hspace{1cm} (3.44)

where $\hat{X}_k$ is the current estimation, $K_k$ is the Kalman gain, $Z_k$ is the measured value and $\hat{X}_{k-1}$ is the previous estimation. In that way it becomes possible to estimate
the true value of the measurement by knowing the prior estimated value.

The Kalman filter is trying to estimate the state \( x \in R^n \) by solving the linear stochastic differential equation

\[
x_k = Ax_{k-1} + Bu_k + w_{k-1},
\]

with a measurement \( z \in R^m \) that is

\[
z_k = Hx_k + Bu_k + v_k.
\]

The random variables \( w_k \) and \( v_k \) represent the process and measurement noise and are assumed to be independent, white and with normal probability distributions \( N(0, Q) \) and \( N(0, R) \) respectively.

The \( n \times n \) matrix \( A \) relates the state at the previous time step \( k-1 \) to the state at the current step \( k \), in the absence of either a driving function or process noise. The \( n \times l \) matrix \( B \) relates the optional control input \( u \in R^l \) to the state \( x \). The \( m \times n \) matrix \( H \) relates the state to the measurement \( z_k \).

**Kalman filter algorithm**

The Kalman filter estimates a process by using a form of feedback control. The equations for the Kalman filter are divided in to two sets: *time update* and *measurement update* equations. The time update equations (predictor) are responsible for projecting forward in time the current state and error covariance estimates, in order to obtain the *a priori* estimates for the next step. On the other hand, the measurement update equations (corrector) are responsible for the feedback so as to obtain an improved *a posteriori* estimate.

The time update equations are

\[
\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k,
\]

and

\[
P_k^- = AP_{k-1}A^T + Q,
\]

The measurement update equations are

\[
K_k = P_k^-H^T(HP_k^-H^T + R)^{-1},
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),
\]

and

\[
P_k = (I - K_kH)P_k^-.
\]

After each iteration the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This feature makes the Kalman filter to have memory of the previous states and recursively to condition the current estimate based on all of the past measurements.
3.2. ESTIMATION

3.2.2 Sequential measurements from one sensor

In order to have a distributed estimation in sequential measurements from one sensor it is possible to use the time and measurement update steps of the Kalman filter. Thus the Kalman filter can be seen to be a combination of estimators. This also forms an alternative proof of the optimality of the Kalman filter in the minimum mean squared sense under the stated assumptions.

Furthermore, it is important to assume that the sensors are able to transmit information to the central node at every time step \( k \).

Consider a random variable evolving in time as

\[
x(k + 1) = Ax(k) + w(k),
\]

(3.52)

the measurements from the sensors have the form

\[
z(k) = Hx(k) + v(k),
\]

(3.53)

where \( v(k) \) is again white zero mean Gaussian noise with covariance matrix \( R \). We wish to obtain an estimate of \( x(k) \) given all the measurements \( z(0), z(1), \ldots, z(k) \) [XGF13].

We can combine the local estimates to obtain a global estimate. Let \( \hat{x}(k|k - 1) \) be the estimate of \( x(k - 1) \) based on \( Y \) and \( P(k - 1|k - 1) \) be the corresponding error covariance. Then the estimate of \( x(k) \) given \( Y \) is given by

\[
\hat{x}(k|k - 1) = A\hat{x}(k - 1|k - 1),
\]

(3.54)

with the error covariance

\[
P(k|k - 1) = AP(k - 1|k - 1)A^T + Q.
\]

(3.55)

Thus the estimate of \( x(k) \) is given by the combination of local estimates

\[
P^{-1}(k|k)\hat{x}(k|k) = P^{-1}(k|k - 1)\hat{x}(k|k - 1) + H^T R^{-1}z(k),
\]

(3.56)

and

\[
P^{-1}(k|k) = P^{-1}(k|k - 1) + H^T R^{-1}H.
\]

(3.57)

In that way it is possible to estimate the state by using sequential measurements from one sensor.

3.2.3 Dynamic sensor fusion

In the case of multiple sensors that generate measurements about a random variable that is evolving in time, we can use a technique that utilizes the Kalman filter for combining sequential measurements.
There are two possible solutions for this scope. The centralized Kalman filter, in which the central node (gateway) implements a Kalman filter, and the distributed Kalman filter, in which each node sends a local estimate and thus helps a lot to reduce the calculations complexity in the central node. It is obvious that the centralized Kalman filter should be avoided because the central node needs to handle matrix operations that increase in size as the number of sensors increases.

One could say that the best choice is to use a distributed Kalman filter so as to split the computations complexity with the help of each node and to reduce calculations in the gateway. However that means that the nodes will work more and thus will need more power which is always limited in wireless sensor network applications.

**Distributed Kalman filtering**

Using the results from subsection 3.2.2, the global error covariance matrix and the estimate are given in terms of the local covariances and estimates by

\[
P^{-1}(k|k) = P^{-1}(k|k - 1) + \sum_{i=1}^{N} (P^{-1}_i(k|k) - P^{-1}_i(k|k - 1)), \quad (3.58)
\]

and

\[
P^{-1}(k|k)\hat{x}(k|k) = P^{-1}(k|k-1)\hat{x}(k|k-1) + \sum_{i=1}^{N} (P^{-1}_i(k|k)\hat{x}_i(k|k) - P^{-1}_i(k|k-1)\hat{x}_i(k|k-1)). \quad (3.59)
\]

Based on these results we can now choose between two different ways for the dynamic sensor fusion depending on the calculations complexity and transmission expenses that we are able to do from the node side.

**More effort from the gateway** In the first way the nodes send only their local estimate \(\hat{x}_i(k|k)\). The gateway combines the estimates from each sensor using the above equations. By using the time update equation \(\hat{x}(k|k-1) = A\hat{x}(k-1|k-1)\) the gateway can calculate all of the missing terms of the equations. Furthermore, all the covariances can be calculated without any data from the nodes.

**More effort from the nodes** On the other hand, each sensor could transmit the quantity

\[
(P^{-1}_i(k|k)\hat{x}_i(k|k) - P^{-1}_i(k|k-1)\hat{x}_i(k|k-1)) + z_i(k). \quad (3.60)
\]

In that way the fusion center will just add all of those quantities sent by all the nodes, in order to calculate easily the term \(P^{-1}(k|k)\hat{x}(k|k)\). The covariances don’t
depend on the data and can be calculated to the fusion center with little effort. The simplicity of the calculation of the global estimate comes from the observation that

$$P^{-1}(k|k-1)\hat{x}(k|k-1) = \sum_{i=1}^{N} z_i(k).$$ (3.61)

Of course the second way is easier but it comes with the price of more energy consumption from the node side.

### 3.3 Discussion

As it is obvious from sections 3.1 and 3.2, in detection there is a hypothesis and a decision rule. If the hypothesis is approved then, according to the fusion rule, the system comes to a decision among a known and finite set of choices. On the other hand in estimation the case is to determine and estimate a parameter. In the unique case that the decision rule in detection has infinite choices, then detection and estimation come to an intersection. However, detection problems usually are binary and between the cases of existence or not of a signal.

That being said, in a building automation system where we care about the appearance or not of water there is no need to send to the fusion center the exact value of the amount of water. It is enough to know that the water level has surpassed a threshold or not and to warn the user about this state. On the other hand, it is of significant importance to know the exact value in the node so as to compare with the greatest reliability the threshold with the measured value. So, a distributed detection could be applied to the fusion center and an estimation of the measured value to the nodes could solve the problem efficiently and with great reliability. Nevertheless the exact design of the system should be defined before such decisions could be made.

The distributed Kalman filtering is not appropriate in our situation because it comes to a global estimation for all of the sensors. In that way if only one sensor has a positive result and all the others are in a normal state, the fusion could provoke a wrong decision. On the other hand the usage of the technique 'Sequential measurements from one sensor' yields to a reliable estimation of the node state that can easily be compared to a threshold in order to make a decision. In this system a detection method should be utilized in order to set the appropriate threshold that is compared with the estimated parameter. A data fusion with an OR fusion rule could provide the appropriate result in which every sensor has its own power to alarm the whole system.

In this point we should mention that in the dynamic sensor fusion we make the assumption that the sensors are able to transmit information to the central node at
every time step. However, we know that wireless communications do not use the most reliable medium, so we should definitely take under consideration the case that multiple steps are not received by the gateway. That could mean that the whole procedure could not continue properly for each step and cause more problems than not using at all any estimation technique. Probably it could be better if the nodes calculate their local estimates and transmit those to the gateway without fusion. In that way at least we could know an estimate of the true value for each node and be sure that there is not any problem to the fusion center.

Finally, from the power point of view we should mention that in detection if the probabilities are static and not continuously updated, then the threshold at the nodes can be calculated only once since there are not any changes in the next steps of the operation. In that way we can save a lot of energy from the node side. However the estimation technique is continuously updated to a new estimation of the parameter so there is more power consumption. The same can be also said about the time-to-alarm metric, since detection can give a faster notification to the system that the estimation due to the less calculations. Finally, the results in estimation are accurate and precise compared to the fuzzy results of the detection. Nonetheless the accuracy and precision of the system is a matter of design of the system according to the needs of the system itself and the final user.
Chapter 4

System: Software and Hardware

One of the main goals of this thesis is to implement a wireless sensor network and to be able to interact with it. In this chapter the hardware and software that has been used for the implementation of the network is described. Furthermore there is a description of the graphical user interface (GUI) which gives the user the opportunity to supervise the operation of the network.

4.1 Hardware

The hardware of the system consists of three parts. The nodes that collect data from the environment, a gateway that receives the data from the nodes and a computer that receives the collected data from the gateway and present them to the user. The last part should be system independent, so there is no any reference on it, however there is a discussion on the programs that run as a GUI on it.

4.1.1 Nodes

The most important part and structural unit of a WSN is a node. A node is usually equipped with sensors and an antenna in order to send a collection of data to the gateway. In this thesis we used the telosB mote platform for the operation of the WSN. Some of the main features and characteristics of this platform are

- IEEE 802.15.4 Compliant
- 250 kbps, High Data Rate Radio
- TI MSP430 Micro-controller with 10kB RAM
- Integrated Onboard Antenna operating at 2.4GHz
CHAPTER 4. SYSTEM: SOFTWARE AND HARDWARE

<table>
<thead>
<tr>
<th>Operating Mode</th>
<th>Current Draw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>1.8 mA</td>
</tr>
<tr>
<td>Sleep</td>
<td>5.1 µA</td>
</tr>
<tr>
<td>Radio ON, RX</td>
<td>23 mA</td>
</tr>
<tr>
<td>Radio ON, Idle</td>
<td>21 µA</td>
</tr>
<tr>
<td>Radio ON, Sleep</td>
<td>1 µA</td>
</tr>
</tbody>
</table>

Table 4.1: Current draw

- Data Collection and Programming via USB Interface
- Open-source Operating System
- Integrated Temperature, Light and Humidity Sensor
- Outdoor Range up to 100m and indoor up to 30m

Power consumption

Node power consumption is of outmost importance in a WSN, so it is useful to know the current draw from the selected node. As we can see from table 4.1 the telosB mote platform has low current consumption.

4.1.2 Sensors

Another important part of a WSN is the sensors. In our situation we used two kind of sensors for the experiments, a humidity sensor and a light sensor.

Humidity Sensor

The humidity sensor on the telosB platform is a Sensirion SHT11. The humidity sensor range is from 0 until 100% RH. Furthermore the resolution is 0.03% RH and its accuracy is ± 3.5% RH. This characteristics make the sensor among the most accurate sensors for a low price humidity sensor of the market.

Light Sensor

The visible light sensor which is on the telosB platform is called Hamamatsu S1087. Its range is from 320 nm to 730 nm which includes all the visible light spectrum. However there is also a Visible to IR sensor on board that extends the range of the spectrum until 1100 nm. In figure 4.1 we can see the photosensitivity according to the wavelength for both of the light sensors.
4.2. SOFTWARE

There are three different devices that had to be programmed in order to get the final result. The nodes, the gateway and the computer. The first two are programmed with a dedicated for WSN operating system called TinyOS and the computer uses a Java program for the processing and the presentation of the results.

4.1.3 Gateway

A gateway collects the data from the nodes and either sends them to another device for processing or it processes them itself. In our situation a telosB mote platform was used as a gateway and since it has not the required processing power it is sending the data to a computer through a serial port.

In the beginning of the thesis there was a plan to use a dedicated and autonomous device as a gateway that it could process the received data and present them to the user either through an HTML-page or a screen installed on top of the device. However, this is a more expensive and complex solution and it was not preferred.

Figure 4.1: Photo Sensitivity of the Hamamatsu S1078 (Visible) and S1078-01 (Visible to IR) Light sensors
4.2.1 TinyOS

TinyOS is a free and open source operating system and platform created to serve WSNs. It is written in the nesC programming language as a set of cooperating tasks and processes.

TinyOS is designed to support the concurrency intensive operations required by networked sensors with minimal hardware requirements. It uses the Active Message Communication model for building non-blocking applications and higher Networking capabilities like Multihop ad hoc routing.

TinyOS was used in this thesis for the programming of the MSP430 Microcontroller of the nodes and the gateway as it is explained on the following sections.

Node code

Each node is programmed with a TinyOS program that takes advantage of the humidity or light sensors of the telosB mote platform and gathers information about the environment periodically. The measurement from the sensor is processed with a Kalman filter, a decision is taken according the presence or not of the event and the value is sent over the air to the gateway. For the communication between the node and the gateway it is used the Active Message Communication model. The measurement from the sensor is taken every 5 seconds when a timer reaches a given value.

Each node is programmed with the same code but during the programming it gets a different identification number that is also sent together in every packet to distinguish the messages. Furthermore, the nodes are programmed with a specified team identification number, so as not to get confused from other nearby networks.

Gateway code

The code of the gateway is more complex that the one from the nodes, mainly due to the additional work to send the data to the serial port. Usually for the programming of the gateway it is used an example TinyOS program called Basestation, but in our situation this is not the case.

The TinyOS serial stack is implemented to work using Active Messages so that the received packets can be easily forwarded to the PC using the Serial Active Message Communication model that provide the same high-level interface as its radio counterpart. The TinyOS serial stack has layers for packet formatting, error checking, and a read/write buffer.

However, for our project, we wanted to simplify the PC side so we did not have to use the TinyOS JNI libraries to receive TinyOS Serial Active Message packets. Instead a custom simple program is used that receives the Active Message data
from the nodes and forward them as bytes directly to the serial port by using the UartStream component for accessing the serial port.

In the same time, the gateway is also doing the same operation of a simple node by taking measurements from its sensors. Nevertheless, this is not needed for the actual system implementation.

Finally, the gateway in getting the RSSI values from its radio chip in order to calculate the distance between the node and itself. The RSSI value is then used for the presentation of the nodes in the graphical user interface.

### 4.2.2 Graphical User Interface

Another important part of the system is the graphical user interface (GUI). The reason of the importance is that it is the only part of the system that the user can see and interact with, so it should operate correctly and to show every needed information for the supervision of the network.

For the programming of the part that receives the data sent from the gateway through the serial port and the data fusion it was used Java. The selection of this language was intended because it can run in almost every operating system. For the presentation of the data and the actual GUI it was used the Processing programming language.

Processing is an open source programming language and integrated development environment (IDE). The language builds on the Java language, but uses a simplified syntax and graphics programming model. In figure 4.2 an instance of the GUI is shown.

From figure 4.2 we can see that the interface is intuitive and can be easily handled from a user. Each node is shown in the GUI as a circle with the gateway in the middle and the color of each circle declares the state of the environment. If the global decision from the fusion rule is $u_0 = 1$ then the gateway circle becomes red and enables the event. The distance of the node from the gateway is determined by the RSSI value acquired from the radio chip.

There are buttons to control the network (Start, Stop), a button to close the program (Quit) and other buttons to choose properties, to see the graph of the measurements etc. In the bottom of the GUI there is a terminal that informs the user for various events.

If there is a problem detected from the WSN, then the GUI informs the user both by showing the problem on the terminal and by sending a tweet at the user’s Twitter account. Furthermore, it sends a signal to a device that makes an actuation in order to try to solve the problem automatically before the user’s action. For the Twitter notification is was used the Twitter4J Java library that handles the Twitter API.
CHAPTER 4. SYSTEM: SOFTWARE AND HARDWARE

One of the most important features of this program is that it can dynamically add and remove nodes, so in case of bad reception or low battery the user can be informed about which node had the problem.

4.3 Graphical User Interface tutorial

A graphical user interface is important because it is the only part that connects the WSN with the final user. For this reason it should be intuitive, easy to use and to show all the needed information efficiently. In this section there is a tutorial on how to use the program using all of its features.

4.3.1 General Description

As it is shown in figure 4.3, in the initial window there are seven buttons at the left and one knob at the right side of the GUI. The knob is responsible for the setting

![Figure 4.2: Graphical User Interface](image-url)
of the threshold. The buttons are explained in the following list:

- **Start**, which starts the operation of the WSN
- **Stop**, which pauses the operation of the WSN
- **Quit**, which quits the program
- **Properties**, which opens the properties window
- **Test Twitter**, which tests if Twitter is set properly
- **Graph**, which shows a graph with the measurements for the last 30 minutes for every node
• **Legend**, which explains the meaning of the node’s characteristics

At the bottom of the GUI there is a message box which informs the user for various actions or options. In a future version of the GUI the message box could be a command line in which the advanced user can control the WSN faster.

In order to start the operation of the WSN, the user should first place the gateway in the USB port. In the case where the Start button is pressed and the gateway is not in place, the message box informs the user that it cannot start. If the button Start is pressed and the user wants to pause the operation, it is possible to press the Stop button. This is making possible to pause the operation which can continue by simple pressing again Start.

In the properties window shown in figure 4.4 there are several options that the user can tweak. To begin with, in the case where the operating system cannot detect the right port of the gateway, the user has the ability to choose it from a drop-down list. There is also a button Reload, that updates the list in the case that there is such a need. Furthermore, there are three switches that control different parts of the WSN. The first one is the rotation switch from which the user can choose whether the graphical representation of the nodes is rotating or not. The second switch disables the actuation and the third one disables the Twitter notifications. Finally by pressing the OK button it is possible to save the chosen options.

As it is said earlier, the graph button is responsible for the presentation of the measurements for every node. On the right corner of the window there is a drop-down list from which the user can choose a node in order to see its measurements. Figure 4.5 shows the window with the graph of node with id 0.

Finally by pressing the Legend button, the user can see the meaning of the node’s characteristics. A figure of this window is shown if figure 4.6.
4.3. GRAPHICAL USER INTERFACE TUTORIAL

4.3.2 Operation

If the gateway is placed in the USB port and the user presses the Start button, the WSN is starting its operation. The nodes are presented in the GUI as circles. In the beginning if there is not any node in range, the GUI presents only the gateway as a circle in the middle of the window. Since the program is dynamically adjusting to the node population, a message and a new circle is presented every time there is a detection of a new node.

Furthermore, a node is disappearing if it does not send any data to the gateway for a fixed amount of time. The user is informed about this action and in that way it is possible to understand if there is a problem with a certain node, for example if its battery is low or if the node is out of range.

The color of the node depends on the local decision and the color of the gateway on the global decision. If a node detects an event then it becomes immediately red, otherwise it is cyan. In figure 4.7 it is shown that the node with id 4 has detected an event and it is red, but the other nodes are cyan because there is not any event in their area. The global decision is shown by the color of the gateway which is green.

In the case where the data fusion rule decides that there is an event, the gateway circle in the middle becomes red too. In figure 4.8 it is shown the case where the global decision becomes $u_i = 1$. If the actuation and Twitter notification are enabled, then the GUI will act immediately in order to prevent a disaster and to inform the user. The notification sent to the user is shown in figure 4.9.

Moreover, the size of the circle shows closeness to the threshold. The reason of this characteristic is that if the node measurement is very close to the threshold,
then the user should have a hint that there might be a problem with the specific node.

Also, the alpha level of the node color shows its reliability. The more reliable the node, the more intense is its color and the user understands if the measurements are reliable or not. Finally, the physical distance between the node and the gateway is determined by the RSSI value for this node.

### 4.3.3 Conclusion

From all the above it is shown that the user can intuitively understand the state of the WSN by looking the graphical representation of the nodes. If there is a need for more detailed information, then there is the graph button that shows the exact value of the final measurements. Nevertheless it should not be needed since the graphical representation shows all the important information in a very simple way.
4.3. GRAPHICAL USER INTERFACE TUTORIAL

Figure 4.7: One node has detected an event but the global decision is $u_i = 0$

Figure 4.8: The global decision is $u_i = 1$ and the gateway becomes red
Warning! Water leakage detected! Check immediately! \#wsn 149:154:147:86:47

Figure 4.9: Twitter notification example
Chapter 5

Simulations and Experiments

In this chapter the experiments and simulations that were conducted during the thesis are presented. The main goal of this is to show the abilities of the proposed WSN according to the different methods used to increase the reliability of the final result. In the beginning there is an analysis on the Kalman filtering which is applied on the nodes and then on the data fusion techniques. Finally, there is a discussion on the results in section 5.3.

5.1 Estimation

As it is already mentioned in previous chapters, estimation is used in order to reduce the error and eliminate the noise from a measurement. In that way it is achieved a higher level of reliability from every noisy sensor like, for example, the light sensor.

In this experiment the light sensor was sampled every second in a steady condition and the node was forwarding through the serial port the measurements both with and without the Kalman filtering so as to see the actual difference of the results and thus the performance of the technique. Figure 5.1 shows the result of this procedure. It is obvious that the processed measurements have considerably reduced the noise caused by the sensor’s low precision.

It is also worth-mentioning that the variance of the data from the unprocessed data in this particular experiment is 11.8 in contrast to 0.18 which is for the filtered data. That means that the set of measurements in the case of the unprocessed data is much more spread out than the set with the filtered data due to the noise.

5.1.1 Kalman filtering performance on the final decision

It is apparent that Kalman filtering contributes a lot for the making of a reliable local decision and a reliable local decision leads to a reliable global decision. However
in a system where power efficiency matters and the less number of cycles made by the microprocessor the better, one could say that estimation is not playing a major role in the reliability of the whole system. This is actually the reason for which there is the need for data fusion in the first place, which gathers information from many noisy or in general low reliability sources in order to reach a global decision which should be more reliable.

However, it is beneficial to make a comparison between the fusion of local decisions that were made after Kalman filtering and the fusion of the "low reliability" local decisions. Thus, a metric of the performance of the Kalman filter and a WSN designer can compare the pros and cons of having this feature.

In order to measure the performance of the Kalman filtering on the final decision of the data fusion, eight sets of one hundred experiments were contacted using five nodes

1. Without Kalman filter when there is not an event \( (H_0) \)

2. Without Kalman filter when there is an event \( (H_1) \)
3. With Kalman filter when there is not an event ($H_0$)

4. With Kalman filter when there is an event ($H_1$)

In each of the four sets above there were two subsets of experiments one without and one with intentionally produced noise. Furthermore, for each of these eight sets we got one hundred sets of five local decisions (one from each node). Each of these local decision sets give a global decision which is depended on the fusion rule.

In table 5.1 it is shown the percentage of every result the experiments gave. For instance, in the experiment where the nodes used Kalman filtering, there was intentional produced noise and there was not any phenomenon ($H_0$), in 36% of the experiments none of the nodes noticed an event, in the other 43% one node noticed an event and in the rest 21% two nodes detected an event incorrectly.

On the other hand, in the similar case of the system where there is noise, there is not any event ($H_0$), but there is no Kalman filtering, the results are very different. As an example, in this case four out of five nodes detected incorrectly an event in the 14% of the experiments. This result proves that the Kalman filter provides more reliability in the final decision, since no matter which fusion rule is being used, if four out of five nodes agree on the existence of an event the global decision will be $u_0 = 1$.

According to the results of table 5.1 it is possible to find the probabilities

- Decide there is no event when there is no event, $P(H_0|H_0)$
- Decide there is event when there is no event, $P(H_1|H_0) = P_F$
CHAPTER 5. SIMULATIONS AND EXPERIMENTS

<table>
<thead>
<tr>
<th></th>
<th>Without Kalman</th>
<th>With Kalman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No noise</td>
<td>With noise</td>
</tr>
<tr>
<td>( P(H_0</td>
<td>H_0) )</td>
<td>1</td>
</tr>
<tr>
<td>( P(H_1</td>
<td>H_0) )</td>
<td>0</td>
</tr>
<tr>
<td>( P(H_0</td>
<td>H_1) )</td>
<td>0</td>
</tr>
<tr>
<td>( P(H_1</td>
<td>H_1) )</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.2: Probabilities that show performance of the Kalman filter

- Decide there is no event when there is event, \( P(H_0|H_1) = P_M \)
- Decide there is event when there is event, \( P(H_1|H_1) = P_D \)

These probabilities give the performance of the network and in that way it is possible to see the effects of the Kalman filter. Obviously, the fusion rule plays a major role in this purpose, since the global decision is calculated according to it. In this case the simple data fusion rule will be used which is actually a majority rule for such a small number of nodes.

In table 5.2 it is shown the probability for every case of the table 5.1 to happen. It is clear that in the case where there is a Kalman filter, the results are correct even if there is intentionally produced noise. However, in the case of the noisy environment and without Kalman, there is 25% probability to have a false alarm.

5.2 Detection and data fusion

In the detection and data fusion there are three different simulations. In the first simulation the data fusion rule is applied with stable local false alarm and local miss probabilities. As it is mentioned in section 3.1.4, since the state of the system is stochastic there is not any way to determine the local false alarm and local miss probabilities \( P_F \) and \( P_M \). So, having in mind that the system consists of identical sensors, these probabilities have a fixed value, at least in the beginning, which is the same with the prior probabilities.

However, the fixed values cause the fusion rule to become a majority rule where the global decision is \( u_0 = 1 \) if \( N/2 + 1 \) nodes have \( u_i = 1 \) as their local decision. Obviously this is not a flexible scheme, because there could be many more combinations of nodes that imply the detection of an event, but the system is unable to
make this decision. For this reason there is a suggestion of two flexible implementations of the fusion rule that adjust their local probabilities \( P_{Fi} \) and \( P_{Mi} \) according to their reliability and the use of knowledge coming from the neighboring nodes to reach a decision faster.

### 5.2.1 Setup of the simulation

To make a comparison between the different implementations, the simulations should be contacted in the same environment. For this reason the fusion rule was simulated with the same sequence of data received by the sensors. The purpose was to create all the possible sets of local decisions for a fixed number of nodes. For this simulation a fixed number of 10 nodes was used which results into \( 2^{10} = 1024 \) different local decision sets from \( u = \{0,0,0,0,0,0,0,0,0,0\} \) to \( u = \{1,1,1,1,1,1,1,1,1,1\} \).

### 5.2.2 Simple data fusion

The simple case of the data fusion is where the local false alarm and local miss probabilities \( P_{Fi} \) and \( P_{Mi} \) are stable during the whole operation of the network. This simulation can result only to one final result and it does not depend on the sequence of the local decision sets. The number of times that the system detects an event is every time the number of the local decisions are more than \( N/2 \), here 5, and in this situation is 386 times out of 1024.

### 5.2.3 Data fusion with adjustable \( P_{Fi} \) and \( P_{Mi} \)

In this case \( P_{Fi} \) and \( P_{Mi} \) become adjustable according to their reputations, which means their agreement with the majority. If a very small minority of sensors decides that there is an event but the vast majority disagrees, then the system decides to reduce the reliability of the minority nodes. In this way a broken sensor or a sensor with wrong measurements can lose its reliability by increasing its \( P_{Fi} \) and \( P_{Mi} \). When these values reach 0.5 which is the top value that can be given by the system, then the sensor is neglected by the fusion rule because \( \log((1 - P_{Mi})/P_{Fi}) = 0 \) if \( P_{Fi} = P_{Mi} = 0.5 \) (See 3.1.2).

Furthermore, this experiment depends on the sequence of the local decision sets, but the initial values of \( P_{Fi} \) and \( P_{Mi} \) is not important because it will be adjusted automatically in a few cycles to the suitable values.

The variations of the \( P_{Fi} \) and \( P_{Mi} \) for \( i = 1, \ldots, N \) are presented in figure 5.2. We can see that many sensors lose their reliability for many times according to their local decision. In this point it should be mentioned that this figure only tries to indicate that there are variations on the \( P_{Fi} \) and nothing more.
The total number of times that the system detected an event has been reduced to 285, which shows that the system did not take under consideration some nodes with low reliability in contrast with the experiment in 5.2.2. However it is important to mention that some of the sets detected an event and produced a $u_0 = 1$ without half of the nodes having a local decision $u_i = 1$. This shows that the fusion rule is not just a majority rule as in 5.2.2.

### 5.2.4 Data fusion using knowledge from the neighbors

One interesting case of data fusion is when the neighbors’ knowledge is considered for the adjustment of the reliability of each node. In this situation the nodes gain reliability when their neighbors’ local decision agree with them. The more neighbors agree with a specific node, the bigger is the increment in its reliability. Again this experiment depends on the sequence of the local decision sets and this is the reason that the sequence was exactly the same with the other experiments.

In this point it should be mentioned that this method is a more complicated version of the implementation described in 5.2.3. It is interesting that the number of
5.3. DISCUSSION

Data Fusion Mode | Detected events (out of $2^N$) | Minimum number of nodes for $u_0 = 1$
---|---|---
Simple | 386 | 6
With adjustments | 285 | 5
With neighbors | 304 | 3

Table 5.3: Data fusion methods for $N = 10$

detections increased to 304 in this situation in contrast to 5.2.3. This was expected and the reason is that many nodes that before had low reliability, this time had neighbors in agreement with them and thus they caused an event. For instance, a triplet of neighbor nodes can produce an event even if it is minority. In table 5.3 there is a summary of the different methods.

5.3 Discussion

According to the results of the previous sections it is obvious that there is an increase in the reliability of the wireless sensor network thanks to the estimation and detection techniques. In section 5.1 it is shown that the Kalman filter produces results with less noise which is very important for the local decision of the nodes. In section 5.1.1 and more specifically in table 5.1, according to the experiments it is shown that the Kalman filter indeed helps the data fusion rule to give more reliable results in a noisy environment. These results suggest that estimation is an important factor for the wireless sensor networks’ reliability.

Furthermore, in table 5.3 it is shown that using adjustable $P_{F_i}$ and $P_{M_i}$ in the data fusion rule, it is possible to achieve detection even with three nodes out of ten, if the reliability of some other nodes is low. This result could never be achieved in a simple data fusion rule. All these show that the designer of a wireless sensor network has many choices in order to create a reliable WSN.
Chapter 6

Proposal for a smart home WSN

The purpose of this chapter is to suggest possible implementations of a wireless sensor network for building automation to the future WSN designer. After the analysis of the previous chapters, there are some options that can determine the performance of the WSN, according to the application, that will be described in the following sections.

6.1 Hardware

It is obvious that the hardware plays an important role in the planning of a wireless sensor network. In the beginning the designer should first consider a compromise between reliability, security, power efficiency and hardware complexity\(^1\). Some of these factors can limit a lot the performance of the system. For instance, if the designer choose to make a highly secured WSN, then the system will be less power efficient and it will be needed more complex hardware from the node side.

However, as it is stated many times before in this text, BAS have to be reliable and secure. The nodes should communicate with each other and the gateway with protocols that provide security and reliability. This means that the hardware should be complex enough in order to be able to handle this kind of features.

A platform like the one used in this thesis, telosB, can provide everything to the designer in order to make a secure and reliable BAS. However, if double way communication is not important in the application, then simpler platforms with limited resources could be used. Furthermore, the designer should be able to define the cost of the information which depends upon the application. In this way, it is possible to decide easier on the complexity of the security and thus the hardware.

Regarding power efficiency which is a very important aspect of a WSN, a designer should be very careful and try to find the needs both of the WSN and the customer.

\(^1\)Cost is a very important factor too but in this proposal it will not be considered.
In the case where maintenance is difficult or even impossible the designer should find ways that either make the WSN work for a rational period of time in respect to the application or to make a WSN that is able to scavenge power from its environment or from the gateway. An example of a technique that scavenges power from the gateway is RFID. Its features and restrictions are explained in detail in section 2.4.

To sum up, the hardware of a WSN should be the main concern of the designer in the beginning of her or his work. In this thesis the designer chose the telosB platform as it could provide all the needed features in order to have higher reliability. There are more details about this choice in section 4.1.

6.2 Software

After the choice of the hardware platform, the next task is to choose the operating system that runs on the hardware of the WSN. There are many noteworthy operating systems like TinyOS or ContikiOS. As it is explained in section 4.2, TinyOS was used in this thesis because it is lightweight, robust, simple and it has a large community on the internet.

Nodes and Gateway were programmed with TinyOS programs that used several features of this operating system. The communication between the nodes was implemented with the Active Message model which is very easy to use and provide higher networking capabilities. In the case where the designer wants to build a mesh network (section 2.1), the Active Message communication model can provide all the required methods to do it.

Furthermore, security can be achieved by using either the CC2420 radio chip encryption as explained in 2.6.4 or by implementing a security algorithm like AES or RSA, 2.6.2. There is also the choice of scrambling the data, which is a simple technique to achieve encryption explained in section 2.6.3.

One very important feature that the designer could add to the WSN is localization. This can be achieved by using RSSI as it is explained in section 2.7. However, it is important to know that the nodes should have as less code as possible. Many features are important but the designer should make a careful decision in order to keep the node code simple, functional and power efficient.

Finally, the WSN should have a way to present its results. Depending on the type of the gateway there are different choices. In the case where the gateway is simple and has to be connected to another machine, a GUI in a cross-platform language should be implemented that presents the results to the user. In the case where a complex gateway is used, then it could have a dedicated interface on the gateway device. Moreover, one very interesting way is to create a HTTP-server on the gateway that presents the data on a web-page. An example of a cross-platform GUI that was implemented in this thesis is explained in section 4.2 and in more
6.3 Detection and Estimation

Detection and estimation help a WSN to become more reliable. Both techniques were used in this thesis, since building automation systems should be reliable. For the detection, the designer should know about the nature of the events that the system is trying to detect. For example, section 3.1.3 provide suggestion for the case that the prior probabilities are uncertain. In the case where the events are rare, like in this thesis, section 3.1.4 provides suggestions that could help the designer. On the other hand there are suggestions for the system that detects frequent events in section 3.1.4. The data fusion technique in section 3.1.2 provides an optimal solution to the detection problem.

Estimation is the other method that increases the reliability of the WSN. The problem with this technique is that it is too complex and requires many calculations from the side of the node. However, in section 3.2.3 it is suggested a method that needs less calculation from the node side and more effort from the gateway. Of course if there is not any limitation with the power, it is certainly beneficial to use Kalman filtering together with the data fusion rule. In that way both the local decision and thus the global decision become more reliable. Furthermore, we should consider that there could be a state where all the sensors receive a simultaneous instant noise-spike that the Kalman filter could eliminate. Nevertheless, this is a very rare situation.
Chapter 7

Conclusion

7.1 Results

After the implementation of the WSN and the testing, there are some noteworthy results. To begin with, Kalman filter increased the performance of the system. The local sensor measurements became more reliable and the experiment in table 5.2 shows that Kalman filtering is important for the correct final decision.

Furthermore, from the same table we have seen that the optimal data fusion rule that it was used makes a \( u_0 = 1 \) decision only when there is an event. Moreover, the data fusion using knowledge from the neighbors and with adjustable \( P_{F_i} \) and \( P_{M_i} \) contributes to a more efficient detection that takes under consideration the history of the system. In that way problematic sensors can be neglected and deleted from the system.

Finally, the graphical user interface connects everything into a simple and intuitive environment that informs the user with Twitter notifications when there is an event and takes action with actuators.

To sum up

- Kalman filter increases reliability of the local sensor measurements.

- Detection and data fusion makes a decision only when there is certainty.

- Graphical user interface connects everything into a simple and intuitive environment that informs the user using Twitter and takes action through actuators.

However, there are still some problems which are described in the following section.
CHAPTER 7. CONCLUSION

7.2 Critique

As every project ever made, this thesis work has also some problems that should be mentioned in order to be able to know what is the future work that should be done. To begin with, the data fusion rule used for the detection part of the thesis could be more complex. A solution described in Chapter 3 with the distributed Neyman-Pearson detection seems the most appropriate.

Furthermore, a distributed Kalman filter described in 3.2 could provide a global estimation for all the sensors. This result could be better for some kind of applications. However, it is not perfect for the application this project is focusing.

Another issue that could be better is the implementation of the experiments. It could be much better if there was the opportunity to make an experiment with at least 10 sensors in order to test the functionality of the proposed techniques. However, simulations using Matlab helped in this direction. Moreover, the sensors located on the telosB platform were not so accurate and precise, but this helped to determine the performance of the Kalman filter. Unfortunately, due to this problem it was not possible to make exactly the same experiment for the same environment with the same noise.

Finally, power efficiency from the node side and security are two issues that have not tested in the final implementation. In the following list there is a summary of the issues that this critique is mentioning about

- More complex detection rule
- Experiments with real sensors (at least 10 sensors for the experiment needed)
- Distributed Kalman
- Use more precise and accurate sensors
- Work on power efficiency
- Security with low requirements

Nevertheless, these problems can be solved in the future. In the following section there is a list with improvements that will make the designed wireless sensor network more complete.

7.3 Future work and improvements

There are many aspects of the WSN that need improvement. In this section there are some suggestions for the most important ones. Firstly, the detection and data fusion rule used could be a more complex and thus more efficient one. Although the
7.3. FUTURE WORK AND IMPROVEMENTS

The technique used in this thesis is adjusting to the state of the environment, it would be better if there was a technique that models the actual application.

About the localization part of the WSN, as it is already stated in section 2.7, there is the need for positioning and refinement. In that way it will be possible to know the exact position of every node and thus increase the functionality of the WSN.

To continue with, the WSN’s current topology is the Star network. However, an advanced WSN should have a mesh network topology. In that way it becomes possible for every node to reach the gateway even in the cases where distances are very long for the star network. Of course in order to achieve this there should be an appropriate MAC and routing work.

Moreover, the current WSN has not any security implemented. Although many tests were conducted with the security from the radio chip CC2420 (section 2.6.4), the final implementation of the WSN has not any security due to the power efficiency. A few other ways were investigated, like the scrambling in section 2.6.3, but there was not enough time in order to apply it on the system.

Finally, the gateway used for the experiments was a telosB platform like the ones used as nodes. It would be much better if there was the possibility for a dedicated gateway that can handle higher data bit-rates. Also, it could be possible to create an HTTP-server on the gateway that can publish the results of the WSN on the internet. In that way there is not any need for a PC that is connected to the gateway and the system becomes more independent.

In the following list there are the improvements as explained in the previous paragraphs

- More complex detection techniques
- Positioning and refinement for the localization
- Mesh network topology instead of star
- MAC and routing for the mesh network
- Security with low requirements
- Dedicated gateway that can handle higher data bit-rates and it can be independent.

Of course there are many other improvements that could make the designed WSN more complete, but the most important features according to the author’s judgement are included.
Bibliography


