Faculdade de Engenharia da Universidade do Porto

Civionics - An Interdisciplinary and Emerging Area

João Miguel Figueiredo Ferreira

Provisional Version

Master Thesis
Electrical and Computers Engineering
Major Automation

Supervisor: Prof. Doutor Adriano da Silva Carvalho

July 2011

Abstract

Civionics is the combination of electronic engineering with civil engineering; it is an emerging area, which the main application is the use of electronic systems for structural health monitoring. Although the use of SHM systems presents benefits on long term, the initial cost is still too high. On this dissertation the study of new approaches is made in order to validate cheaper but reliable alternatives.

Wireless sensor networks have shown promising results in a lot of areas, therefore the application of these platforms for structural health monitoring may represent a path to follow. This dissertation gives an overview about the usage of these devices for civil structures monitoring and describes the developing process of a solution of this kind.

It is important to understand that it is not the aim of this document the presentation of a final product, but the validation of this new concept. Thus, an extensive study of different solutions to implement wireless sensor networks is presented. In order to validate some of the technologies proposed, a prototype system is developed and tested.

This document shows, once again, that systems based on wireless networks can achieve promising results. To conclude, it is pointed out the direction that future projects should follow.

Resumo

Civionics é a combinação de engenharia electrónica com engenharia civil. Representa uma área emergente cuja principal aplicação é a utilização de sistemas electrónicos para monitorização do estado de estruturas civis. Embora a utilização de sistemas deste tipo apresente benefícios a longo prazo, o custo inicial é ainda demasiado elevado. Nesta dissertação o estudo de novas soluções para monitorização de estruturas é realizado a fim de validar alternativas mais baratas, mas com resultados igualmente bons.

Redes de sensores que utilizam comunicações sem fios têm mostrado resultados promissores em muitas áreas, portanto, a aplicação destas plataformas para monitorização de estruturas pode representar um caminho a seguir. Esta dissertação apresenta uma visão geral sobre o uso desses dispositivos para monitorização de estruturas civis e descreve o processo de desenvolvimento de uma solução deste tipo.

É importante compreender que não é o objectivo deste documento a apresentação de um produto final, mas a validação deste novo conceito. Portanto, um estudo de diferentes soluções baseadas em comunicações sem fios é apresentado. A fim de validar algumas das tecnologias propostas, um protótipo de sistema é desenvolvido e testado.

Este documento mostra, mais uma vez, que os sistemas baseados em comunicações sem fios podem alcançar resultados promissores. Para concluir, neste documento são apresentadas várias sugestões sobre a direcção que o desenvolvimento de sistemas deste tipo deve tomar.

Agradecimentos

Quero agradecer à minha família por todo o apoio demonstrado. Todos os jantares fora de horas e conversas esclarecedoras.

Ao meu orientador, o Prof. Doutor Adriano da Silva Carvalho pelos conhecimentos transmitidos e disponibilidade mostrada ao longo do desenvolvimento deste trabalho.

Ao Prof. Doutor Joaquim Figueiras pela disponibilidade e compreensão mostrada ao longo do projecto. Também ao Eng. Augusto Faria pela transmissão de conhecimentos sobre estruturas e ajuda nos ensaios realizados em laboratório. A todos os técnicos do LABEST que sempre se mostraram disponíveis para ajudar em qualquer questão de teor mais prático.

Quero deixar também um muito obrigado ao meu colega de projecto Pedro Moreira pelo esforço e empenho demonstrado no desenvolvimento deste projecto. Não poderia deixar de referir também todos os colegas presentes no laboratório I105 pela solidariedade e momentos de lazer proporcionados. A todos eles, um muitíssimo obrigado.

Por fim mas não menos importante à Carina Gaspar por todo o apoio mostrado nos momentos mais críticos deste trabalho.

Contents

List of Figures

List of Tables

Symbols and Acronyms

ACK	*Acknowledge*
ADC	*Analog-to-Digital Converter*
ALU	*Arithmetic and Logic Unit*
API	*Application Programming Interface*
APL	*Application Layer*
CA	*Collision Avoidance*
CAN	*Controller Area Network*
CANH	*CAN High*
CANL	*CAN Low*
CAP	*Contention Access Period*
CF	*Compact Flash*
CFP	*Contention Free Period*
CRC	*Cyclic Redundancy Check*
CSMA	*Carrier Sense Multiple Access*
DAC	*Digital-to-Analogue Converter*
DAS	*Data Acquisition System*
DLA	*Dynamic Load Allowance*
DLC	*Data Length Code*
DSP	*Digital Signal Processor*
DSSS	*Direct Sequence Spread Spectrum*
EOF	*End of Frame*
FCC	*Federal Communications Commission*
FDMA	*Frequency Division Medium Access*
FFD	*Full Function Device*
FHSS	*Frequency Hopping Spread Spectrum*
FIFO	*First In First Out*
HMI	*Human-Machine Interaction*
I/O	*Inputs/Outputs*
I^2C	*Inter-Integrated Circuit*
IC	*Integrated Circuit*

IDE	*Integrated Development Environment*
IEEE	*Institute of Electrical Engineering*
IP	*Internet Protocol*
ISM	*Industrial, Scientific and Medical*
ISO	*International Organization for Standardization*
LAN	*Local Area Network*
LLC	*Logical Link Control*
MAC	*Medium Access Control*
MEMS	*Microelectromechanical Systems*
MOb	*Message Object*
NDE	*Non-Destructive Evaluation*
NRZ	*Non Return to Zero*
NWK	*Network Layer*
OSI	*Open Systems Interconnection*
PAN	Personal Area Network
PC	*Personal Computer*
PCB	*Printed Circuit Board*
PHY	*Physical layer*
PLL	Phase-Locked Loop
PTC	*Positive Temperature Coefficient*
RAM	*Random Access Memory*
RF	Radio Frequency
RFD	*Reduced Function Device*
RIMS	*Remote Intelligent Monitoring System*
ROM	*Read Only Memory*
RTC	*Real Time Clock*
RTD	*Resistance Temperature Detector*
SCL	*Serial Clock*
SD	*Secure Digital*
SDA	*Serial Data*
SHM	*Structural Health Monitoring*
SOF	*Start of Frame*
SPI	*Serial Peripheral Interface*
TCP	*Transmission Control Protocol*
TDMA	*Time Division Medium Access*
UDP	*User Datagram Protocol*
USART	*Universal Synchronous Asynchronous Receiver/Transmitter*
UWB	*Ultra Wideband*
WAN	*Wire Area Network*
WPAN	*Wireless Personal Area Network*

WSN	*Wireless Sensor Network*
ZC	*ZigBee Coordinator*
ZED	*ZigBee End Device*
ZR	*ZigBee Router*

Chapter 1

Introduction

Civil structures are the backbone of our society. We depend on them to keep up the lifestyle that we are used to. This statement can be easily supported if we think about the transportation network that we rely on. The need of roads, bridges and other structures date from the early years of civilization. Nowadays these infrastructures are directly related to the economic success of a nation and are an important parameter to rank the life quality of a certain country [1].

The main goal of this document is to show and discuss ways to preserve and maintain these important constructions which are one of the main supports of the modern world.

1.1 Motivation

The importance of the structures that compose the transportation network is undeniable but as any other construction they need maintenance to keep working properly. Most of the infrastructures that we use nowadays were built about 50 years ago and have suffered from decades of bad maintenance and overuse, leading to the acceleration of their deterioration [2]. The lack of attention for the integrity of these structures resulted on an unsatisfactory condition of our infrastructures and there is now a need to replace or rehabilitate most of them. Also, most of the bridges and roads were constructed to support the demands of that time and it is known that these requirements have changed drastically. The increase of traffic density and the weight augmentation of the new cargo vehicles contributed to the appearance of new requirements [3]. Thus, the estimated lifetime of a structure, calculated by the time it was built does not apply anymore due to the change of the conditions that the infrastructure is subjected to.

The engineers are not only interested in keeping the old structures working. As the world and society progress the infrastructures should follow, new technologies, materials and methods of construction are being developed and there is a need to test them in a more accurately way.

The problems described previously are not new in anyway, and the engineers have relied on their experience to detect behaviours that could lead to structural failures. However with the increase of new structures with higher degrees of complexity, it is essential to find new ways of structural monitoring to understand their behaviour, which are more accurate and reliable. In addition, with the development of new sensors and data acquisition systems, civil engineers start to apply techniques based on electronics to extract reliable information from the structure behaviour. They found it advantageous to work together with electrical engineers to develop solutions that could monitor and even detect structural damage without human intervention. A new field of studies with the name Civionics was born.

In short, the expansion and development of new infrastructures and the need of keeping the old ones working with high safety levels, led to the search for new methods of structure monitoring. When data acquisition systems evolved, becoming relatively cheap and capable of providing accurate and reliable information about the integrity and behaviour of an infrastructure, engineers understood that this was the solution they were looking for. However, there are still requirements to be achieved. In this document it will be shown some of the new solutions and advances on the field of Civionics and structural health monitoring. The goal is always the same; to gather more and better data as well as trying to keep the costs acceptable.

1.2 Aim and Research Objectives

The construction sector is not always open to new ideas and most of the times prefer to use old and outdated methods that they can understand. There are additional costs when installing innovative monitoring systems in a structure and for these systems to be accepted they have to prove themselves worthy. This research will show the advantages and disadvantages of using wireless sensors to build a structural health monitoring system as well as provide a comparison between the different architectures that are used nowadays.

It will also be developed and implemented a wireless sensor system capable of measuring useful data for structural studies. This equipment will be tested and the results discussed giving the reader an understanding of the applicability of the solution proposed.

The objectives of this thesis can be gathered in the following list:

- Discuss and study the importance of Civionics and Structural Health Monitoring;
- Study and compare the various types of transducers and the technologies they use, with emphasis on MEMS sensors;
- Compare the different types of solutions available and discuss the use of systems based on wireless sensors;
- Design and implement a solution based on wireless sensors that can achieve the some of the requirements defined;
- Test and validation of the proposed design.

1.3 Work Methodology

It is important to note that this project complies two parts. This document only reports the work done on the wireless sensor network but a top level system based on a human-machine interface was also developed. In charge of this last part is my fellow colleague Pedro Moreira. The need of integration both parts led to a lot of shared work, especially in the protocol development like it is possible to notice further ahead.

In order to achieve all the goals proposed a work methodology was defined. First of all there was a need to understand what structural health monitoring stands for and what are the primary requirements of this kind of systems, e.g. what are the main parameters that have to be measured.

Afterwards a state of art revision was made in order to identify the technologies already proposed and their main characteristics. The review was focused primarily on architectures based on wireless sensors. The goal was to elaborate a document where all these equipments were described and discussed giving the reader an idea of the problems and limitations that this kind of systems have. At this point it was made a clear division between the different elements that make up a wireless based system and was also presented the most important technologies used in each of these subsystems.

The next step was to determine the requirements that the solution should fulfil. All the requirements were discussed in order to achieve an agreement between all the stakeholders.

As the study of various transducers used in structural health monitoring systems are one of the subjects of study, a discussion of several sensors was made to determine the differences between some of the most well known technologies. It was given special attention to sensors based on MEMS technology.

At this point the design of a solution began. The final architecture was developed and the construction/implementation phase started. The reader will find later on in this document all the justifications and decision made when choosing the hardware and the topology of the system.

The final goal is to validate the proposed solution, a number of tests were though in order to obtain accurate results. All experiments made are well documented and the conclusions obtained are discussed at the end of this document.

1.4 Outline of the Document

The outline of the document follows the methodology already described. Therefore next chapter starts by introducing the background studies. It is discussed various types of structural health monitoring systems and the components of a wireless smart sensor platform. It is also made an overview of the wireless sensors available in the market.

Chapter three will present the system requirements which represent the start point for the development of a solution. They are all discussed and justified.

Another objective of this research is the study of the sensors used for SHM. Chapter four will discuss this matter taking special attention to the microelectromechanical systems (MEMS). It is also presented here the chosen wireless sensor platform and the type of networks used to integrate all the system.

Chapter five is dedicated to the implementation process. It starts by the hardware developed explaining all the circuits constructed. Following is the protocol implementation on the different units. This chapter concludes describing the overall functioning of the system.

Chapter six shows the tests made to the solution proposed and present the results obtained. The discussion around these results is also made given the reader an idea of what was accomplished.

Last chapter represents the conclusion of this project. It defines also possible future work.

Chapter 2

Background Studies

This chapter presents the research made before the start of the project. It starts by giving the reader a definition of Structural Health Monitoring (SHM) and explains the different types of existing systems.

Afterwards it is presented the state of art. The main focus of this subchapter is the discussion of wireless sensor platforms already developed and the presentation of various technologies used for the different modules.

The conclusion of this chapter includes a discussion of the various problems encountered when trying to develop a SHM system using wireless sensors.

2.1 Structural Health Monitoring

The concept of Structural Health Monitoring is related to the process of implementing a damage recognition system for aerospace, civil and mechanical infrastructures [4]. The aim is to give a diagnosis of the "state" of the structure at every moment during the lifetime of the infrastructure. Other goal is to detect the appearance of damage, thus a definition for this phenomenon is needed. For the purpose of this research, damage is defined as a violation of the designed characteristics. In other words, when the behaviour of the structure is a lot different from the specified in the design it is possible to say that the structure is damaged. The concept of "lot" depends on the infrastructure in cause and has to be defined by specialists.

Although SHM can be seen as an improved way to make Non-Destructive Evaluation (NDE), it is much more than that; it involves the integration of sensors, smart materials, data transmission and processing ability inside the structure. It is so integrated that can be considered part of the structure in a way that can even change the initial design [5].

The SHM system should be able to provide useful and reliable information about the condition and integrity of the infrastructure at any given time, in this way it can help to prevent catastrophic situations, allow for short-term verification of innovative designs and improve the maintenance effectiveness [6].

It has already been given a swift overview about the meaning of Structural Health Monitoring; however it is important to know that sometimes a SHM system does not integrate all these features.

These systems can be used in a wide range of structure types, from aerospace, mechanical devices and civil structures. In this document SHM will always be related to civil structures with the main focus being bridge structures.

Next, it is discussed different categories and classifications based on the complexity of the system and its ability to detect damage.

2.1.1 Classification of different types of Structural Health Monitoring Systems

There are a wide variety of tests that can be made to infer about the integrity of a civil structure. The reason for having this variety is because it is impossible to evaluate all the characteristics of the structure (that ultimately define its state) with only one type of test. Thus, the SHM system has to be adapted to the type of test desired in order to maximize its efficiency. The next image shows the division of SHM systems according to the type of test and expected information.

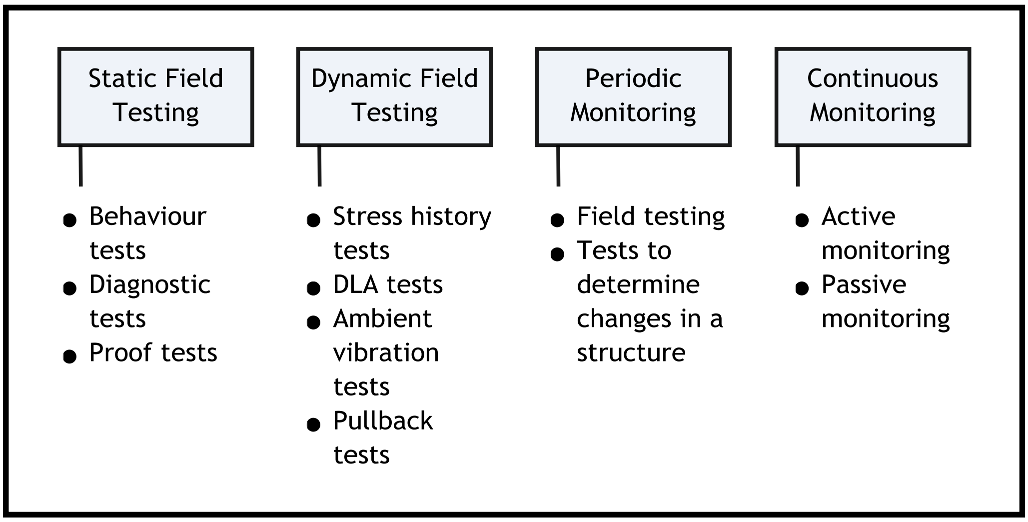


Figure 2.1 - Subsystems of a Structural Health Monitoring (adapted from [6])

Static Field Testing is based on the use of static loads, they are the easiest to perform and for that reason the most common. These tests infer about the ability of the structure to withstand loads as well as provide data on the deformation that suffers under the influence of a certain weight. The information provided enables the evaluation of the stiffness of the construction and the assessment of its short-term safety. The wide use of these methods is also due to the easy interpretation of results, however dynamic effects such as shock, vibration and resonance are not studied in such trials [6].

Within the category of Static Field Testing are subcategories that represent different types of tests, which are possible with static charges. The Behaviour Tests are used to study the mechanical behaviour of the structure as to validate analytical methods employed to describe it. The test loads must be within the parameters defined in the design for the normal use of the structure.

As for the Diagnostic Tests, they are performed to study the response of a given component of the structure in order to understand its importance and interaction with the rest of the elements. In this way it is possible to assess where does the structure need to be strengthened. The type of load applied is the same of the one used on the behaviour tests.

To complete this category there are the Proof Tests. These are the most and dangerous tests of this category. The load applied exceeds the design parameters and have to be increased slowly while the structure is monitored. If the load is to heavy it can lead to permanent damage and for this reason these tests have to be conducted with extreme caution. The goal is to assess the structure's capability to withstand loads greater than the nominal.

The second category mentioned, Dynamic Field Testing, tries to assess the dynamic behaviour of the structure. Since the structure under analysis in this research is a bridge, the methodology presented here is directed to such construction. The load applied in these tests is usually a vehicle that crosses a bump constructed on the bridge dynamically exciting the structure. The size of the bump, the weight of the vehicle and its speed can be changed in order to get different ranges of excitation. These tests are performed to get information about the resonance frequency of the structure allowing the comparison with the frequency response given by mathematical models.

In this category are presented four new subcategories. Stress Tests History aim to indentify the stress that certain parts of the structure are subjected to. Usually the components under investigation are the ones critical to the survival of the building. The data obtained can be used to calculate the lifetime of the construction before failures due to fatigue start occurring.

Dynamic Load Allowance Tests or DLA Tests are used to determine the dynamic amplification factor. This factor is the result of moving loads, when the load is not static its motion affects the behaviour of the structure leading to a different response to a certain weight. Performing these tests allow the calculation of the amplification factor being than possible to determine the maximum weight that the bridge can in reality handle [7].

Another way to obtain the resonance frequency of a construction is by using natural phenomenon such as wind to stimulate the structure. Tests using this strategy are called Ambient Vibration Tests. This area is still in its initial development phase and there are still many problems to overcome. The main struggle is to control the exact level of excitation that the bridge is subjected too, also it is important to monitor the surrounding environment with a great level of precision to be able to extract useful data from these trials. The correct placement of the sensors in this kind of method is more crucial than in others, which leads to the need of an expert that has a vast knowledge in the field of structures [8].

The last group of tests on this category are the Pullback Tests. They are usually applied to bridges when the wind is not enough to excite the bridge laterally. It is necessary to build a system that pulls the bridge in the lateral direction and then drop it, the information provided by these tests is the same provided by Ambient Vibration Tests. The need of an additional structure to excite the construction under test is a restriction that makes this kind of method underutilized.

The last two categories presented in figure 2.1 are not characterized by the type of load used but by the periodicity of the measurements. Periodic Monitoring refers to tests that collect data only at certain time intervals. This type of tests is related to field observations which are tests performed on the local. This kind of trials is the most common and they are particularly effective when there is a need to understand how changes in the structure have affected its performance. Static Field Testing usually use periodic monitoring as there is no need to be constantly collecting data when the structure is not under investigation.

Finally, it is presented the Continuous Monitoring category, as the name suggests monitoring takes place continuously. Due to the high complexity and cost of this type of SHM it has only been used recently [5]. Usually it is only applied to constructions of great value or that are in high risk environments. The continuous monitoring systems fall into two categories, the most common ones acquire information and store it so that it can be interpreted later, the most modern and sophisticated send data as soon as it is received to central post in order to be analyzed in real time.

The need to classify SHM systems in categories according to the information desired and the type of test performed is understandable but there is another approach to categorize these systems. This one is based on the damage identification process, the capability of the system to detect damage and to what extent.

The damage identification process is usually structured into the following levels [1]:

- Damage detection, where the presence of damage is detected;
- Damage location, systems that can determine where is the damage located;
- Damage typification or quantization, where the type and the quantity of damage is inferred;
- Damage consequences, where the severity and consequence of the damage is assessed.

An increase on the detail of the damage characterization leads to a need of a more sophisticated system which means higher costs (Figure 2.2). For this reason it is of the best interest to determine what kind of information is really interesting for a specific application/structure [9].

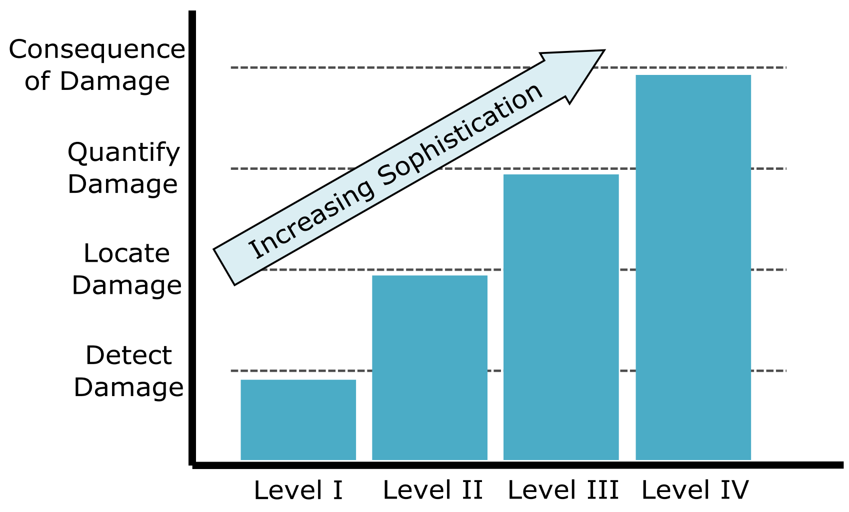


Figure 2.2 - Classification of different systems relative to their performance (adapted from [3])

2.1.2 Composition of Structural Health Monitoring Systems

Ideally a SHM system should provide information about significant damage in a structure when commanded. This information can be transmitted by a local area network or sent to a remote outpost to be processed and interpreted. In any case a wide range of equipment has to be used to build a system capable of extract data from a group of sensors, process, transmit and report the information gathered to the end user. Because a SHM system is composed of such a large variety of devices, it can be divided in subsystems depending on the role of each one. As it was already said there are different types of systems for different applications so the structure here presented has to be seen as a generalization of what is used by most of the SHM systems [3].

Typically a modern SHM system will consist of six common components, namely:

- Acquisition of data, sensory system;
- Communication of information;
- Intelligent processing and data analysis;
- Storage of processed data;
- Damage detection algorithms;
- Interface with the user, retrieval of information.

The next image shows how these subsystems interact and allows a better understanding of this concept. Notice that this is only a typical architecture, not all the SHM systems have all these modules and some other have extra equipment installed to fulfil some specific requirements.

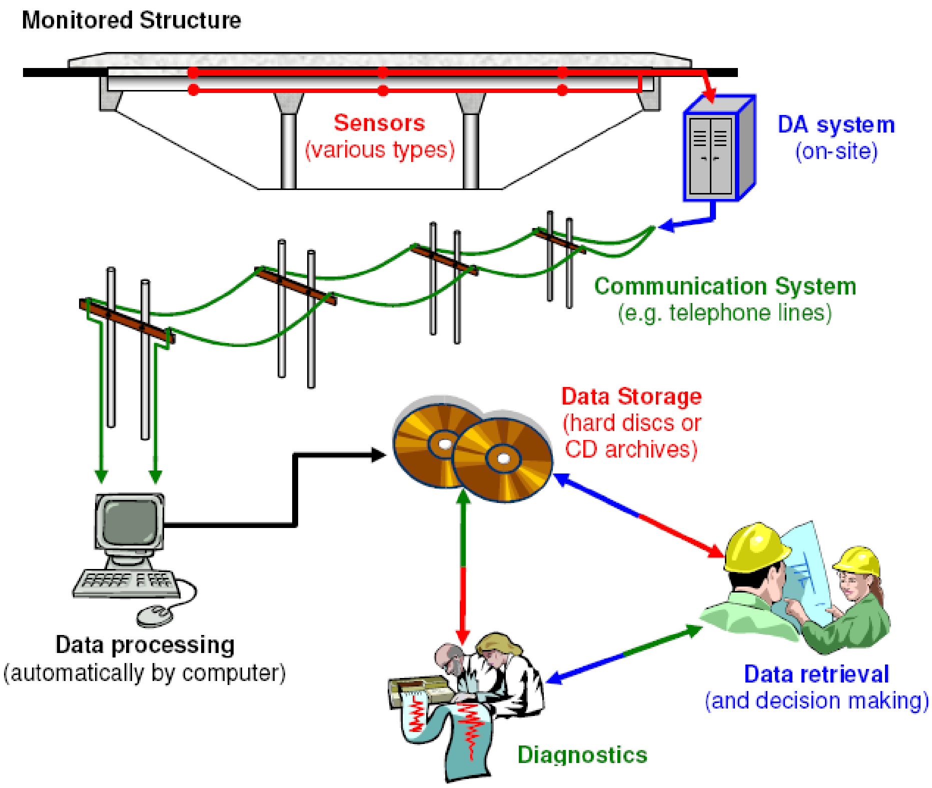


Figure 2.3 - Typical SHM system architecture [3]

The first module presented in figure 2.3 is the sensory system. This is one of the most important parts of a SHM system. If the sensor net does not work properly you cannot extract any reliable information about the structure. Typically these sensors are resistive strain gauges, vibrating strings, inclinometers, temperature sensors, accelerometers, linear displacement sensors and lately fiber-optic sensors. Once again the sensors used vary from application to application and depend mainly on the measures required by the owner of the structure and the existing conditions. Correct selection of sensors is essential to the success of any SHM system. It is required to take special attention to the following factors:

- Resolution, related with the precision required;
- Bandwidth, depends on the frequency of the phenomenon one wants to measure;
- Power consumption, crucial for wireless sensor networks;
- Reliability;
- Robustness;
- Durability;
- Physical size and difficulty of installation;
- Limitations in terms of signal transmission;
- Cost.

Another big issue when deploying sensors in a structure is the place where they are attached. Even high quality sensors cannot produce useful data if placed in the wrong places. There is a need to study the structure and realize its key points in order to find where it is worth to put these transducers.

The review and study of the sensory part of a SHM system is one of the aims of this research and have a dedicated chapter further ahead. At this point it is interesting only to understand its importance as well as some of the issues that the designer has to have in consideration.

Next it is presented the data acquisition module which is, combined with the sensory part, perhaps the most important part of the system and the main focus of this work. The data acquisition module is responsible for acquiring the signal provided by the sensor net and transform it into something more than an electrical magnitude. In the case of a resistive strain gauge sensor the electric signal that the acquisition module receives has to be turned into strain which is the measurement that has interest to the analyst.

The complexity of the data acquisition system (DAS) has been increasing and there are many issues to take into account when choosing the device that will perform this role. One of the biggest concerns is related to the data sampling rate. A well thought data acquisition algorithm has to capture the adequate amount of data. As one might expect, structures which are heavily instrumented will generate a large amount of data which can easily become unmanageable if the system is not set up efficiently. Data sampling rate should be high enough in order to get all the valuable information from the sensors but not so high that would overwhelm the rest of the system. Decisions regarding appropriate sampling rates should be based on the type of phenomenon that is supposed to be measured [3].

There is a great amount of characteristics that the designer must have into account when choosing the data acquisition system for a certain application. However it is possible to name a few factors that should always be matter of study:

- Resolution, usually referred to the number of bit of the analogue-to-digital converter;
- Sampling frequency, previously discussed;
- Processing power;
- Embedded communication module, important to transmit data to a remote post;
- Power consumption;
- Cost.

These two modules are the backbone of a SHM system and are usually integrated with the structure under evaluation. One issue that has to be addressed is the connection between these parts. If it is a wired solution the cables should be protected against electromagnetic noise and the data should be transmitted in current instead of voltage. This is even more critical when the structure has large proportions. It also helps to have a signal conditioning circuit to eliminate the existing noise on signal coming from the sensor. The conditioning can be made right at the output of the sensor, at the input of the DAS or at both. It is advisable to have digital signal conditioning performed by the data acquisition system computer core. The weight of the cables and their distribution in the structure has to be considered, if not they may affect the behaviour of the structure.

New technologies have been proposed to avoid that electromagnetic noise interfere with the output signal of sensors. Using fiber-optic sensors will eliminate most of the problems related to noise, however they bring some other problems. A slight discussion of these sensors is presented on chapter four.

The use of a distributed system where each sensor or small group of sensors has an independent data acquisition module that communicates through a wired or wireless network is also a solution. Wireless communications are evolving rapidly and it is expected they will be increasingly used for SHM of very large structures in the future. The application of these technologies in monitoring systems is the main focus of this dissertation.

Returning to the architecture shown in Figure 2.3, the communication system refers to the mechanism that transfers data from the site, where it is gathered, to a remote post where the data will be processed and analysed. This allows engineers/owners to monitor the structure remotely eliminating the need for site visits. The transfer of data can be done using telephone lines, internet or even wireless systems like cellular transmission [10].

Afterwards the information sent by the communication system to a remote post is processed. The main purpose of this part of the process is to eliminate irrelevant data. Superfluous information can be caused by digital noise that has to be removed or by the influence of environmental phenomena, usually temperature changes, which have to be compensated in order to obtain the right measures. It is also important to arrange the data in a way that it can be stored more efficiently without sacrificing its integrity. In more sophisticated systems, neural computing and artificial neural network techniques may be employed [3]. For example, on bridges with low to medium traffic only heavy trucks will generate changes in the infrastructure that are interesting to monitor. Using a good data management algorithm means that only relevant information is stored. In the given example the important information is related to the passage of a heavy truck. Modern systems with distributed processing are able to process data from the sensors on site, reducing the amount of information that has to be transmitted to the remote station [11].

Data storage is the next module in the list. The need of having a system that stores all the data collected is obvious. Dependent on the type of system and on its purpose the storage equipment can have different characteristics. One of the biggest concerns is the physical medium where data is stored that should allow information to keep uncorrupted for many years. Also, depending on the amount of sensors and the data sampling rates the storage device should have the capability to store all the relevant information. Most of the systems do not save raw data and only keep processed or analysed information reducing the amount of space needed. Unfortunately, this does not allow for reinterpretation at a later time.

The latter two subsystems shown in Figure 2.3 are related with the understanding of the behaviour of the structure. At this point data is analysed in order to diagnose any significant damage in the structure. Sophisticated systems are already making the diagnosis automatically, using complex algorithms, but typically this process requires the intervention of experts in the area.

It was shown the typical composition of a SHM system. Now this research will focus on the wireless sensors solution and will be mainly concerned with the sensory and data acquisition subsystems.

2.2 Structural Health Monitoring systems based on Wireless Sensors

The importance of structural monitoring in undeniable, these systems usually offer long term cost reductions as they enable to reduce the maintenance of a construction. However, the initial price is still pretty high. As it was discussed previously a SHM system is composed of sensors, data loggers, computers and connecting cables. These devices have to be purchase and installed, which leads to a substantially increase of the initial cost of the structure. Most of the SHM systems incorporate coaxial wires for communication between sensors in order to guarantee reliable measures. The installation of this cables is generally very expensive and labour-intensive: it was estimated that in tall buildings the use of this wires add an excess of $US5000 per sensing channel [12]. It was also determined that this cost increases exponentially with the increase of sensors number [2].

Recently, smart wireless sensors have been considered as an alternative for reducing the costs maintaining the SHM system reliable. This technology integrates the sensor itself with the data acquisition and communication system on a single platform. Since the communications between the numerous platforms is wireless it is possible to reduce the use of full-scale cables reducing the installation cost of the system. Furthermore the integration of computational modules allows the increase of the overall computational power enabling the utilization of distributed processing architectures [12].

These systems can be composed by several different modules but it is possible to define a typical structure of a wireless smart sensor platform. The next scheme shows how these platforms are integrated and the role of each of the elements.

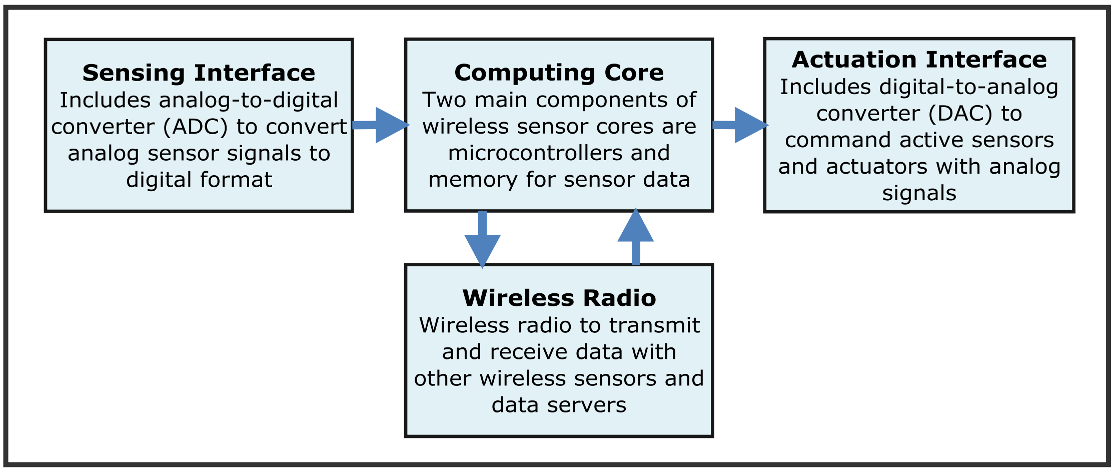


Figure 2.4 - Functional diagram of the various components of a Smart Sensor platform

2.2.1 Sensing Interface

Smart sensors must contain an interface to which sensing transducers can be connected. The sensing interface is mainly responsible for converting the analogue signal coming from the sensor output into a digital representation that can be processed by digital electronics. Typically this interface consists of anti-aliasing filters, current to voltage converters and other circuits capable of conditioning the signal, which is then transmitted to the analogue-to-digital converter (ADC). The ADC is perhaps the most important component of the sensing interface, and its choice has to be made with regard to the system requirements. The main factors when selecting this component are the resolution (number of bits) and the sampling frequency. These characteristics are highly important because they limit the quality of information sent to the computing core and from it to the rest of the system.

There are a wide range of ADCs on the market but for SHM applications the choice usually falls into those that have resolutions between 8-bits and 16-bits and sampling frequencies between 500 Hz and 500 kHz. In theory higher resolutions and sampling rates lead to more accurate readings but in practise they will generate a lot of superfluous information. A resolution of 16-bits is more than enough to acquire signals that have changes of a few mV and a sampling rate of 500 kHz is sufficient to gather data that allows the understanding of the structure behaviour [13]. It is also important to understand that an improvement of these characteristics means an increase of energy consumption which is a critical restriction in a wireless system.

Some smart sensors platforms support other protocols to connect sensors, usually I^2C or SPI. These communication standards are described in more detail on chapter 4.

2.2.2 Computing Core

Once measured data have been acquired by the sensing interface, the computational core takes responsibility of the information. This module has different roles according to the architecture of the overall system but normally it defines where the data is stored, how it is processed and also control the acquisition and communication of this information. In order to perform these tasks the computing core is typically composed by a microcontroller. Usually the microcontroller has access to two types of storage: RAM (random access memory) where all the data required to process the information received is stored and ROM (read-only memory) used to store the firmware or other usual routines. While RAM is easily expanded, ROM is pretty much tied to the microcontroller chosen. There is a wide variety of controllers that can be incorporated in these systems and one major classifier is the size of their internal data bus (in bits). In SHM the choice often falls between the 8, 16, or 32-bits microcontrollers. The maximum clock frequency must also be taken into consideration. The speed of the clock is a direct limitation on how fast the microprocessor is able to execute a given program. It is true that larger data buses and faster clocks enable a higher processing throughput, however the improvement of these characteristics leads to an increase of energy consumption.

2.2.3 Wireless Radio

In order to have the capability to interact with other wireless sensors and to transfer data to remote workstations, a wireless transceiver is integrated on these platforms. This module should be able to transmit and receive data from other platforms, and like the computing core there is a wide variety of equipments in the market. Some of the main factors of choice are communication distance, radio frequency used and signal power. Wireless transceivers used in smart sensors usually operate in the ISM (Industrial Scientific and Medical) frequency bands, typically 900 MHz, 2.4 GHz and 5 GHz. Besides the physical limitations implied by the antenna's design and restricted power consumption, the signal power, which means the communication distance, is limited by legislation. The Federal Communications Commission (FCC) mandates the maximum power an antenna can output is 1W for the United States [13]. Some studies support that is more efficient to use various communication modules that transmit over shorter distances than a single module for long range transmissions [14].

Typically there are two types of wireless signals that can be sent through a selected radio band: signals that only use a single carrier frequency or that use multiple carriers. Narrow-band wireless transmission modulates all of the data upon a single carrier frequency. This strategy is prone to multipath effects and interference which diminish its performance. To enhance the reliability spread spectrum wireless transmission is preferred. Spread spectrum encodes data on a number of different frequencies within a frequency band which reduces the probability of interference. There are a number of methods for modulating data using this strategy, the most used are frequency-hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS) [15].

The need to integrate multiple platforms of different manufactures in one system led to the creation of specific standards for wireless communications. The Institute of Electrical Engineering (IEEE) has developed several protocols in the 2.4 GHz band. The 802.1x or Wi-Fi is mostly used on PC-based applications and allows high data rate transfer but on the downside the associated energy consumption is impractical for the use in smart sensors platforms. Bluetooth supported by the 802.15.1 standard is not as energy costly as Wi-Fi but as limit range that is sometimes too short for SHM systems. Developed especially for sensor networks the 802.15.4 protocol offers a good distance range keeping the power consumption to the minimum (Figure 2.5). Notice that all these standards only define the two lowest layers of the OSI model, Physical Layer (PHY) and Medium Access Layer (MAC) [16].

	IEEE 802.11 Wi-Fi	IEEE 802.15.1 Bluetooth	IEEE 802.15.4 ZigBee	400-900 MHz Proprietary
Applications	PC-Based Data Acquisition	PC Peripherals, Sensors	Wireless Sensors, Automation	Remote Monitoring, Control
Range	30 to 100 m	10 to 30 m	50 to 100 m	100 m to 1 km
Data Rate	54 to 540 Mbps	1 Mbps	250 kbps	20 to 150 kbps
Frequency	2.4 GHz, 5 GHz	2.4 GHz	900 MHz, 2.4 GHz	400 MHz, 900 MHz
Battery Life	Hours	Days	Years	Years
Security	Best	Good	Better	Good

Figure 2.5 - Wireless technologies survey [16]

In order to promote the use of wireless sensors some higher layers have been developed. The ZigBee Alliance has created a protocol based on the IEEE 802.15.4 standard which adds the Network Layer (NWK) and the Application Layer (APL) (Figure 2.6). Notice that ZigBee is an open protocol that anyone can have access to.

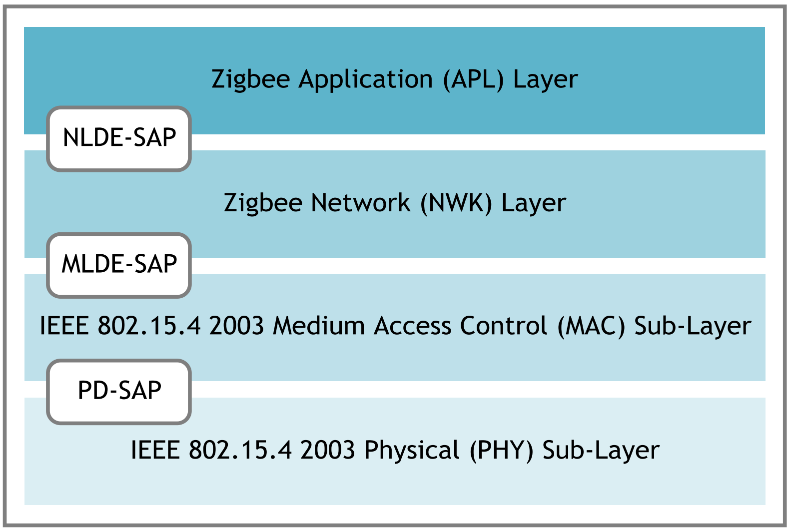


Figure 2.6 – The ZigBee stack

As it is based on the IEEE 802.15.4 lower layers this protocol uses the DSSS spread spectrum technique. In order to improve the robustness the IEEE standard also employ a technique usually known as frequency division medium access (FDMA). This means that it divides the 2.4 GHz band in 16 non-overlapping channels which are 5 MHz apart. As so, ZigBee devices can comfortably coexist with other devices using the 2.4 GHz band.

One specific property of the ZigBee protocol, implemented by the Network Layer, is the definition of three node types. ZigBee Coordinators are used to form the network, the ZigBee Routers can route packets through the network and ZigBee End-Device, which can sleep and go into low-power modes. Using the sleep ability allows the system to save a lot of energy, which is one of the major advantages of this protocol [17].

The ZigBee protocol is not the only one based on the IEEE 802.15.4 standard. One of the biggest competitors is the WirelessHART. This technology was developed in order to address some of the main concerns raised by the industry towards ZigBee. The first big difference is that WirelessHART defines only one type of node able to work as router or end-device, thus increasing the flexibility of the network. Also it uses time division multiple access (TDMA) to share the medium instead of the carrier sense multiple access with collision avoidance (CSMA/CA). The first one allows devices to increase their power savings because they only need to keep the radio on during the required timeslots. However this technique arises some problems related to different nodes synchronization [18].

These two protocols here discussed can support different network architectures like star, tree or mesh (Figure 2.7). However it is important to understand that one of the biggest advantages of having the network layer defined is that it enables the use of multi-hopping mesh architecture in an easy and interoperable way. The advantages of these different architectures will be discussed later in this chapter.

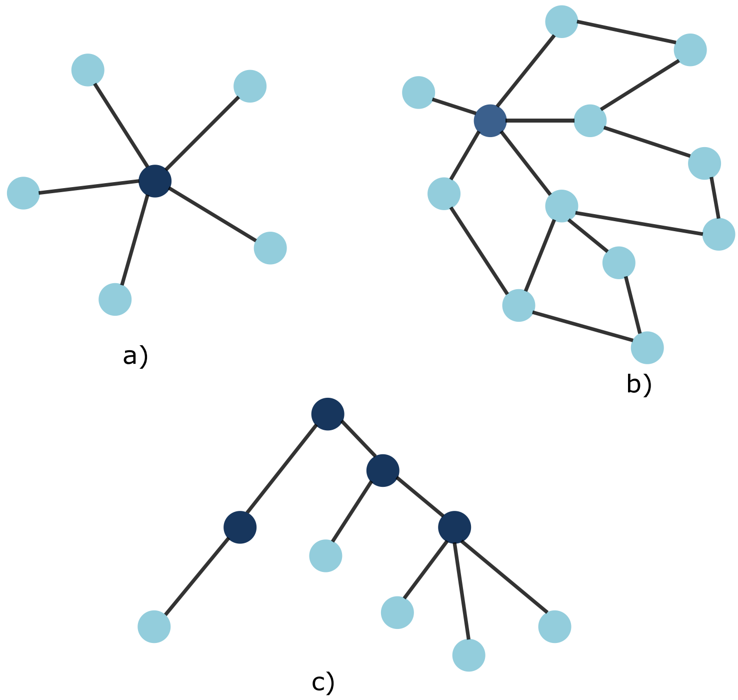


Figure 2.7 - Network architectures: a) Star topology, b) Mesh topology and c) Hierarchical tree topology

There are many more protocols developed on top of IEEE 802.15.4, it is even possible to only use the layers defined by this standard and built a solution from there. Wireless USB is another emerging technology for small and battery-operated devices. It uses the Ultra Wideband (UWB) which supports 480 Mbps over a distance of two meters. This protocol is not meant to function on a scale as large as ZigBee and it is more suitable for PC peripherals. Other solution developed by Zensys is the Z-Wave which is useful for home automation products. It operates in the 868 MHz band for Europe and the 908 MHz for the United States. Usually this technology has a data rate of 9 kbps which is rather inferior compared to the 250 kbps offered by ZigBee [17].

These protocols will be studied in a deeper manner throughout this document. However, the main focus will be ZigBee due to its expressivity in the smart sensors domain.

2.2.4 Actuation Interface

The last subsystem shown in Figure 2.4 is the actuation interface. It provides to a smart sensor the capability to interact directly with the structure where it is installed. Actuators and active sensors can both be commanded by this module which is usually composed by a digital-to-analogue converter (DAC). Actuators can be used to excite the structure in order to obtain better results but it is not always profitable energy wise. As to ADCs there are a lot of choices available in the market and it is necessary to understand the requirements of the actuators and the trade-off between functionality and power consumption [13].

2.2.5 Power Supply

Although not specified in Figure 2.4, the power supply is one of the major concerns of smart sensors design. In SHM systems it does not make sense to communicate all the data wirelessly when there is still the need to equip the structure with cables to supply energy. Therefore, most of the wireless smart sensor technologies are supplied by batteries or some kind of renewable energy source that can be produced on the local.

There are many types of batteries however the choice is usually lithium based batteries. Compared to the nickel-cadmium batteries, lithium batteries provide twice the energy density and allow more autonomy. They have a high cell voltage of 3.6 Volts which allows battery pack designs with only one cell. Their size and weight are also advantages compared to nickel-cadmium batteries and cause little harm when disposed. Despite its overall advantages, lithium has its flaws. It is fragile and needs a protection circuit to maintain safe operation, there are aging concerns when the battery is not in use and most of all it is expensive to produce.

The deep study of portable batteries is out of the scope for this research, yet a summary table is presented. Table 2.1 shows all the important characteristics that the designer has to have in account when choosing a battery.

Methods to increase the autonomy of wireless smart sensors are being developed. The use of renewable energies provided by photovoltaic panels, eolic generators or the utilization of the natural structure vibration are some of the examples [19].

	NiCd	NiMH	Lead Acid	Li-ion	Li-ion polymer	Reusable Alkaline
Gravimetric Energy Density(Wh/kg)	45-80	60-120	30-50	110-160	100-130	80 (initial)
Internal Resistance (includes peripheral circuits in mW)	100 to 200 6V pack	200 to 300 6V pack	<100 12V pack	150 to 250 7.2V pack	200 to 300 7.2V pack	200 to 2000 6v pack
Cycle Life (to 80% of initial capacity)	1500	300 to 500	200 to 300	500 to 1000	300 to 500	50 (to 50%)
Fast Charge Time	1h typical	2-4h	8-16h	2-4h	2-4h	2-3h
Overcharge Tolerance	moderate	low	high	very low	low	moderate
Self-discharge / Month (room temperature)	20%	30%	5%	10%	~10%	0.3%
Cell Voltage (nominal)	1.25V	1.25V	2V	3.6V	3.6V	1.5V
Load Current (C-rate)						
- Peak	20C	5C	5C	>2C	>2C	0.5C
- Best result	1C	0.5C	0.2C	1C	1C	0.2C
Operating Temperature (discharge only)	-40 to 60°C	-20 to 60°C	-20 to 60°C	-20 to 60°C	0 to 60°C	0 to 65°C
Maintenance Requirement	30 to 60 days	60 to 90 days	3 to 6 months	not req.	not req.	not req.
Typical Battery Cost (US$, reference only)	$50 (7.2V)	$60 (7.2V)	$25 (6V)	$100 (7.2V)	$100 (7.2V)	$5 (9V)
Cost per Cycle (US$)	$0.04	$0.12	$0.10	$0.14	$0.29	$0.10-0.50
Commercial Use Since	1950	1990	1970	1991	1999	1992

Table 2.1 - Characteristics of commonly used rechargeable batteries (adapted from [20])

Many solutions based on wireless smart sensors have been developed. Next chapter will present some of the most notorious. Keeping in mind the goal of this dissertation all the technologies discussed are related to structural health monitoring.

2.3 Solutions based on wireless sensor networks

The academic community fast realized the need to reduce costs associated with wired structural monitoring systems. A wide variety of researchers from different institutions developed different prototypes of wireless sensor platforms. Latter partnerships were made and some companies start to invest in this emerging technology. Almost every single solution integrates the subsystems previous discussed, thus this subchapter will give only an overview of the different approaches in a chronological order.

2.3.1 Academic point of view

One of the first solutions presented dates from 1998, Straser and Kiremidjian from Stanford University presented a system based on commercial off-the-shelf (COTS) components [21]. It uses a Motorola 68HC11 microprocessor, chosen for its large number of on-chip hardware peripherals and for its possibility of integrating high-level programming languages for embedding software. The wireless radio is a Proxim Proxlink MSU2 operating on the 902-928 MHz with a maximum range of 300 meters and a data rate of 19.2 kbps. Although the wireless sensor proposed does not give the proper attention to the power consumption it can be considered the first major step by the structural engineering community towards decentralized data processing and wireless SHM [13].

The next system comes from Lynch et al. [22] and was presented in 2001 (Table 2.2). Also from Stanford University, this solution emphasizes the need of powerful decentralized data processing system. The 8-bit Atmel AVR AT90S8515 enhanced RISC (Reduced Instruction Set Computer) microcontroller was the choice for this platform. Representing the continuation of Straser and Kiremidjian work, the idea was to increase the processing throughput without compromising the energy consumption.

Introducing a somewhat innovative concept Mitchell et al. (2002) [23] presented a solution based in a two-tier SHM architecture (Figure 2.8). The idea is to separate the monitoring system in wireless sensors and wireless data server (clusters). The communication between the sensors nodes and clusters is made recurring to the Ericsson Bluetooth wireless transceiver. It operates on the 2.4 GHz radio band and has a maximum reach of approximately 10 meters line of sight. In order to overcome the distance restriction a multihopping strategy is proposed. The Bluetooth radio consumes only 35 mW of electrical power. All the cluster nodes are equipped with a long-range radio which enables the transference of data between them. Furthermore this architecture allows the clusters to have powerful computing cores and it is also proposed the connection to the World Wide Web using cellular modems for long range communications [23].

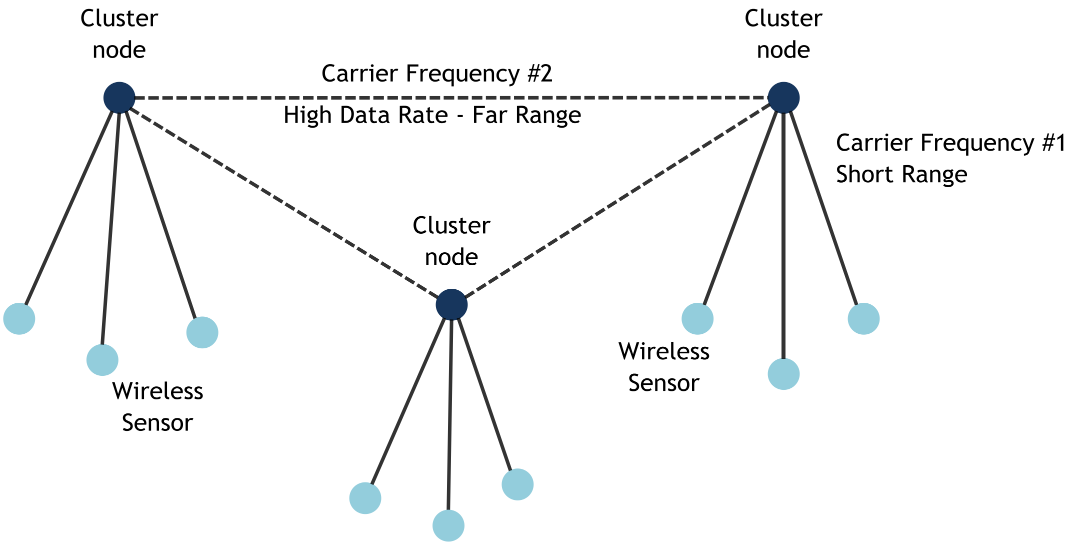


Figure 2.8 - Two-tier architecture proposed by Mitchell et al. (2002) [23]

	Straser and Kiremidjian (1998)	Lynch et al. (2002)	Mitchell et al. (2002)	Kottapalli et al. (2003)	Lynch et al. (2003)
DATA ACQUISTION SPECIFICATIONS					
A/D Channels	8	1		5	1
Sample Rate	240 Hz	100 kHz	20 MHz	20 MHz	100 kHz
A/D Resolution	16-bit	16-bit	8-bit	8-bit	16-bit
Digital Inputs	0	2		0	2
EMBEDDED COMPUTING SPECIFICATIONS					
Processor	Motorola 68HC11	Atmel AVR8815	Cygnal 8051	Microship PIC16F73	Atmel AT90S8515 AVR/ MPC555PowerPC
Bus Size	8-bit	8-bit	8-bit	8-bit	8-bit/32-bit
Clock Speed	2.1 MHz	4 MHz		20 MHz	4 MHz/20 MHz
Program Memory	16 kB	8 kB	2 kB	4 kB	8 kB/26 kB
Data Memory	32 kB	32 kB	128 kB	192 kB	512 kB/448 kB
WIRELESS CHANNEL SPECIFICATIONS					
Radio	Proxim Proxlink	Proxim RangeLan2	Ericsson Bluetooth	BlueChip RBF915	Proxim RangeLan2
Frequency Band	900 MHz	2.4 GHz	2.4 GHz	900 MHz	2.4 GHz
Wireless Standard			IEEE 802.15.1		
Spread Spectrum	Yes	Yes	Yes	Yes	Yes
Outdoor Range	300 m	300 m	10 m	500 m	300 m
Enclosed Range	150 m	150 m	10 m	200 m	150 m
Data Rate	19.2 kbps	1.6 Mbps		10 kbps	1.6 Mbps
FINAL ASSEMBLED UNIT ATTRIBUTES					
Dimensions	15x13x10 cm	10x10x5 cm	5x3.8x1.2 cm	10x5x1.5 cm	12x10x2 cm
Power Consumption			120 mW	100 mW	
Power Source	Battery (9v)	Battery (9V)	Battery	Battery (9V)	Battery (9V)

Table 2.2 - Summary of academic wireless sensing unit prototypes (1998 – 2003) (adapted from [13])

Kottapalli et al. (2003) [24] also used the two-tier network concept. The major accomplishment of their solution was to drastically reduce the power consumption. Wireless sensing units communicate with their corresponding site master using the BlueChip EVK915 which consumes only 36 mW when receiving and 150 mW when transmitting. Powered by alkaline AA batteries, this system can stand 18 months until the portable energy supply depletes [24].

	Aoki et al. (2004)	Mastroleon et al. (2004)	Sazonov et al. (2004)	Allen (2004) Farrar et al. (2005)	Wang et al. (2005)
DATA ACQUISITION SPECEFICATIONS					
A/D Channels		5	6	6	4
Sample Rate		480 Hz		200 kHz	100 kHz
A/D Resolution	10-bit	16-bit	12-bit	16-bit	16-bit
Digital Inputs		0	16		0
EMBEDDED COMPUTING SPECIFICATIONS					
Processor	RenesasH 8/4069F	Microchip PICmicro	Texas Instruments MSP430F1611	Intel Pentium/ Motorola	Atmel AVR ATMega128
Bus Size	8-bit	16-bit/8-bit	16-bit	16-bit	8-bit
Clock Speed	20 MHz			120/233 MHz	8 MHz
Program Memory	128 kB		16 MB	256 MB	128 kB
Data Memory	2 MB			Compact Flash	128 kB
WIRELESS CHANNEL SPECIFICATION					
Radio	Realtek RTL-8019AS	BlueChip RFB915B	Chipcon CC2420	Motorola neuRFon	Max-stream 9XCite
Frequency Band		900 MHz	2.4 GHz	2.4 GHz	900 MHz
Wireless Standard		IEEE 802.15.1	IEEE 802.15.4	IEEE 802.15.4	
Spread Spectrum		Yes	Yes	Yes	Yes
Outdoor Range	50 m	200-300 m	75 m	9.1 m	300 m
Enclosed Range	50 m			9.1 m	100 m
Data Rate		19.2 kbps	250 kbps	230 kbps	38.4 kbps
FINAL ASSEMBLED ATTRIBUTES					
Dimensions	30x6x8 cm	8x8x2 cm			10x6.5x4 cm
Power Consumption			120 mW	6 W	100 mW
Power Source					Five AA batteries (7.5V)

Table 2.3 - Summary of academic wireless sensing unit prototypes (2004 – 2005) (adapted from [13])

While these two last proposals focus on reducing the power consumption, Lynch et al. [25] presented an upgrade to their first system focusing on the processing power. This time a dualprocessor configuration was proposed. It is known that microprocessors with high computational throughput consume more energy, so Lynch et al. [25] overcame this restriction using a two processor design where the low power 8-bit Atmel AVR AT90S8515 is utilized to overall unit control and real-time data acquisition, and the 32-bit Motorola MPC555 PowerPC is in charge of data processing. This second core is not always running allowing low power mode operations, actually it is only turned on when there is data ready to be analysed. It contains 448 kB of ROM and 26 kB of RAM, along with a floating-point arithmetic and logic unit (ALU) which permits the execution of demanding damage detection routines [13].

The next wireless sensing unit prototype called Remote Intelligent Monitoring System (RIMS) is proposed by Aoki et al. (2003) [26]. Designed for the purpose of bridge and infrastructures monitoring, each hardware component is chosen in order to reduce the energetic consumption and size of the platform. One important feature presented in this design is the addition of a dedicated three-axis microelectromechanical system (MEMS) piezoresistive accelerometer (Microstone MA3-04). The study of sensors based on MEMS technology is part of the objectives of this dissertation and its discussion will be presented further ahead. It is important however to refer that these systems are low power consuming and really small (Figure 2.9).

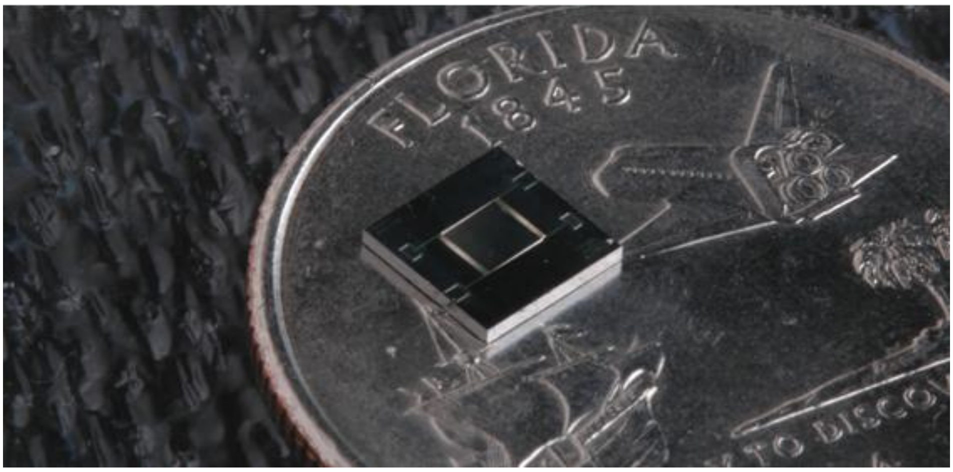


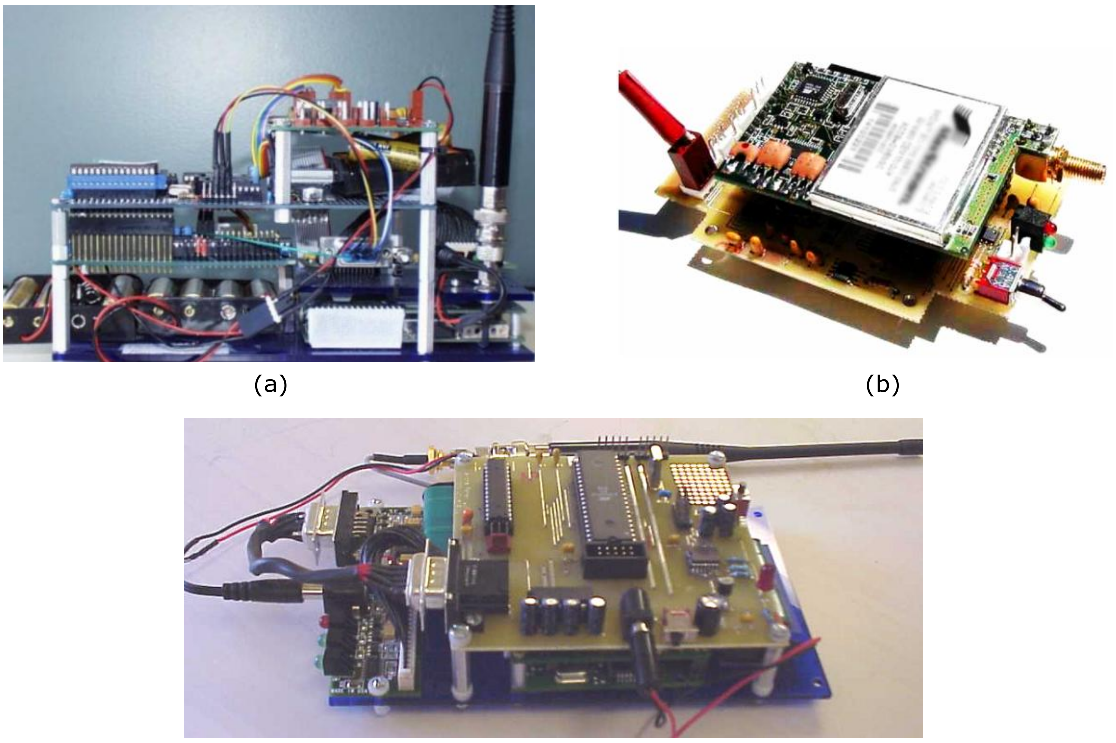
Figure 2.9 - The size of generic MEMS sensor

Extending upon the design proposed by Kottapalli et al. (2003) [24], Mastroleon et al. (2004) [27] have accomplished greater power efficiency by upgrading most of the unit's original components. The main change is the computational core which is changed by a PICmicro. Featuring a low power consumption and high computational performance this microcontroller is capable of achieving real-time processing and time synchronization by using multilevel priority interrupts and a phase-locked loop (PLL) synchronization mechanism [27]. Also the Mircrochip PICmicro enables the dynamic switch between six power modes, thus providing an ultralow power consumption platform.

The first prototype using the IEEE 802.15.4 standard was proposed by Sazonov et al. (2004) [28]. The wireless transceiver used is the Chipcon CC2420 which operates in the 2.4 GHz radio spectrum with a data rate of 250 kbps. It has a range of 10 to 70 meters, yet it only consumes 60 mW when receiving and 52 mW when transmitting.

With a different strategy in mind Allen (2004) [29] and Farrar et al. (2005) [30] proposed a system that emphasizes the computational power providing a wireless sensor design that was capable to perform a broad array of damage detection algorithms. In close collaboration with Motorola Labs they designed a platform that enables seamless interaction with DIAMMOND II, an existing damage detection package written in Java. It was selected a standard PC-104 SBC with a 133 MHz Pentium processor, 256 MB of RAM and a 512 MB Compact Flash (CF) card serving as hard drive. The interface with sensors integrates the DSP56858 digital signal processor (DSP) that is used to for treat the data from the analogue-to-digital converter. All these processing power led to a unit volume of 1750 cm^3 that consumes 6 W of power.

The last academic prototype here presented comes from Wang et al. (2005) [31]. Using the latest commercially available embedded system components they proposed a wireless sensing platform that has multitasking capabilities. This system is able to sample measurement data simultaneous to wirelessly transmitting data to other units. The most attractive feature is the wireless radio. The MaxStream 9XCite wireless modem operates on the 900 MHz band and is capable of data rates up to 38.4 kbps. The communication range is 300 meters in line-of-sight and consumes 250 mW when transmitting and 150 mW receiving. Also MaxStream modem presents an idle mode that consumes only 5 mW making this unit really attractive to periodic monitoring systems.



(a) (b)

(c)

Figure 2.10 - Pictures of some wireless sensing units: (a) Unit from Straser et al. (1998) [21]; (b) Unit from Wang et al. (2005) [31]; (c) Unit from Lynch et al. (2003) [25]

2.3.2 Commercial solutions

A number of commercial wireless sensor platforms have emerged in the recent years that are well suitable for SHM applications. Using this type of solutions can be extremely advantageous because they feature an immediate out-of-the-box operation and good technical support. Many researchers use these commercial solutions to explore their theories and develop add-ons to make the unit more suitable for the application in question [13].

Like for academic prototypes there are too many commercial platforms to discuss them all in this chapter. An overview of the most common systems is made with special attention to the Mote wireless sensor platforms. The popularity of Motes is mostly due to the open source architecture both at hardware and software level. They were initially developed at the University of California-Berkeley and subsequently commercialized by Crossbow [32]. The first prototype developed is called WeC, was produced in 1999 and after commercialized as the Rene Mote by Crossbow. It integrates 8-bit Atmel AT90LS8535 AVR microcontroller which has an internal eight-channel, 10-bit ADC. In order to communicate with the other sensors it has a RF Monolithics TR 1000 wireless radio integrated. This transceiver employs amplitude modulation and operates on the 916 MHz frequency.

Rene Mote was after updated to produce the Rene2 platform which presents the same design with a new microcontroller. The choice recalled on the Atmel ATega163L that has a larger internal memory including 16 kB of ROM and 1kB of RAM. Some studies have been made using this platform to interface with two types of MEMS accelerometers, Analog Devices ADXL202 and Silicon Devices SD-1221. It was first concluded that two sensing channels cannot be sampled at the same time resulting in a relative 30 µs offset between samples. It was found that this offset has a negative impact on the embedded software used to calculate cross-correlation coefficients for sensor signals with high-frequency content. The internal memory featured was also a problem not allowing large buffers of sensor data to be stored. As result only on-the-fly data interrogation was possible to implement [33]. Other issues were raised related to the communications reliability when the single-channel RF Monolithics TR 1000 was tested [34].

In 2002 Crossbow released MICA Mote wireless sensor. The successor of the Rene2 is based on the 8-bit Atmel ATmega103L microcontroller which incorporates 128 kB of flash ROM and 4 kB of RAM. This unit was designed to work with the TinyOS embedded operating system. MICA still integrates the same wireless transceiver of its predecessor. Critical hardware issues were solved. However, time synchronization across a large number of platforms is still not precise enough. Tests identified errors of 7 ms on a system using MICA Motes [13].

The low reliability offered by the singe-channel RF Monolithics TR 1000 was other problem in need of attention. MICA2 was presented in 2003 and brought a new wireless radio with him. The Chipcon transceiver operates on the 900 MHz band and presents a frequency modulation (FM) with high noise immunity. The carrier frequency can be changed via software allowing FHSS encoding techniques. MICA2 was later upgraded to be in compliance with the IEEE 802.15.4 standard. A lot of studies using MICA2 Motes for SHM were made, there are still some issues that need to be addressed but all the researchers consent that wireless sensor solutions for structure monitoring is the path to follow [13].

	UC Berkeley-Crossbow WeC (1999)	UC Berkeley-Crossbow Rene (2000)	UC Berkeley-Crossbow MICA (2002)	UC Berkeley-Crossbow MICA2 (2003)	Intel iMote Kling (2003)
DATA ACQUISITION SPECEFICATIONS					
A/D Channels	8	8	8	8	
Sample Rate	1 kHz	1 kHz	1 kHz	1 kHz	
A/D Resolution	10-bit	10-bit	10-bit	10-bit	
Digital Inputs					
EMBEDDED COMPUTING SPECIFICATIONS					
Processor	Atmel AT90LS8535	Atmel Atmega163L	Atmel ATmega103L	Atmel ATmega128L	Zeevo ARM7TDMI
Bus Size	8-bit	8-bit	8-bit	8-bit	32-bit
Clock Speed	4 MHz	4 MHz	4 MHz	7.383 MHz	12 MHz
Program Memory	8 kB	16 kB	128 kB	128 kB	64 kB
Data Memory	32 kB	32 kB	512 kB	512 kB	512 kB
WIRELESS CHANNEL SPECIFICATION					
Radio	TR 1000	TR 1000	TR 1000	Chipcon CC1000	Wireless BT Zeevo
Frequency Band	868/916 MHz	868/916 MHz	868/916 MHz	315, 433, or 868/916 MHz	2.4 GHz
Wireless Standard					IEEE 802.15.1
Spread Spectrum	No	No	No	Yes (Software)	Yes
Outdoor Range					
Enclosed Range					
Data Rate	10 kbps	10 kbps	40 kbps	38.4 kbps	600 kbps
FINAL ASSEMBLED ATTRIBUTES					
Dimensions	2.5x2.5x1.3 cm	8x8x2 cm			10x6.5x4 cm
Power Consumption	575 mAh	2850 mAh	2850 mAh	1000 mAh	100 mW
Power Source	Coin Cell	Battery (3V)	Battery (3V)	Coin Cell	Battery

Table 2.4 - Summary of commercial wireless sensing unit prototypes (1999 – 2003) (adapted from [13])

A new generation of Mote platforms was introduced in 2003. It was the result of the collaboration between the University of California-Berkeley and the Intel Research Berkeley Laboratory. This new hardware conception proposes only a computational core and wireless transceiver. They recognized that specific sensing applications need specific sensing interface, thus iMote features a highly modular construction that allows separated sensing interfaces to be snapped onto its platform [13]. The iMote computational core is the 32-bit

ARM7TDMI microcontroller running at 12 MHz, which provides four times greater computational power than the one integrated in the MICA Mote. It exhibits 64 kB of RAM for data storage and 512 kB of ROM for running TinyOS. The wireless communication module is the 2.4 GHz Zeevo Bluetooth radio which is integrated with the microcontroller in a single integrated circuit. In order to supply energy to the mote it is used two Panasonic Lithium CR2 3V batteries. Studies by Spencer et al. (2004) [35] have reported the iMote as a powerful tool for future wireless SHM systems.

Following the MICA Mote concept TelosB was presented by Crossbow in 2005. The main update is the wireless module. It has an integrated transceiver that work in compliance with the IEEE 802.15.4 standard. As for the computational core the Texas Instruments MSP430 with a 16-bit bus is integrated. In regard to MICA unit this platform exhibits a powerful processor without increasing the power consumption [36]. Tmote Sky is the successor of Telos. It was presented by Moteiv in 2005 and includes increased performance, functionality and expansion. With TinyOS help, Tmote proportionate emerging wireless protocols and is integrated in the open source software movement [37]. Both these units have not yet been deeply studied for SHM applications. They are only referenced in this document for historical reasons.

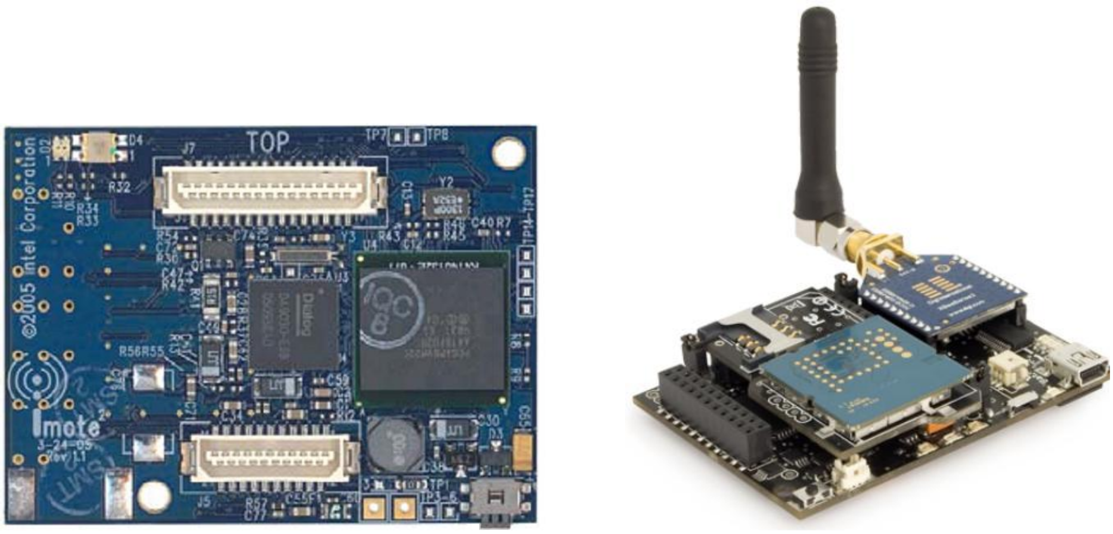
In 2007 Crossbow began selling the iMote2 (Figure 2.11). This platform is considered one of the most powerful solutions on the market. It integrates the Intel PXA271 XScale processor that can be configured to operate in different clock frequencies being the lowest 13 MHz and the highest 416 MHz. The ability to change the speed at which the computational core works is an important feature for energy optimization. This equipment supports several operating systems including TinyOS 1.1, TinyOS 2.0, SOS and Linux. iMote2 is considered one of the most advanced platforms on the wireless sensor field [38].

The last platform discussed in this overview is the Waspmote (Figure 2.11). It was presented by Libelium in 2009 and brought with him a lot of new features. The major characteristic is the 0.7 μA that it consumes when in sleeping mode. Also Libelium took the modular concept to a new level. Waspmote can be integrated with a lot of different wireless communication technologies. They offer a wide variety of XBee modules that support numerous standards like IEEE 802.15.4, IEEE 802.15.1 and ZigBee. Some tests showed an outdoor range of 12 Km using the 900 MHz radio and 7 km with the 2.4 GHz radio [39]. Also Libelium presents a lot of interface sensor units that can be snapped into the Waspmote without any hardware changes [40].

As it was showed a variety of solutions are available to integrate wireless sensor networks (Table 2.2, 2.3, 2.4 and 2.5). At the same time that these platforms become more powerful at processing level and more autonomous they start to present a viable solution for structural health monitoring systems. The price of these devices is quite reduced compared to wired solutions, for example Crossbow sells the iMote2 for €100(US) a unit [41]. The reduced cost is also one of the reasons for the success of wireless sensor systems in the research community.

	UC Berkeley-Crossbow TelosB (2005)	Moteiv Tmote Sky (2005)	Crossbow IMote2 (2007)	Libelium Waspmote (2009)
DATA ACQUISITION SPECEFICATIONS				
A/D Channels	8	8		7
Sample Rate				15 kHz
A/D Resolution	12-bit			10-bit
Digital Inputs				
EMBEDDED COMPUTING SPECIFICATIONS				
Processor	TI MSP430	TI MSP430 F1611	Intel PXA271 XScale	Atmel ATmega1281
Bus Size	16-bit	16-bit	32-bit	8-bit
Clock Speed	4 MHz	32 kHz	13 MHz to 416 MHz	8 MHz
Program Memory	48 kB	16 kB	256 kB SRAM	8 kB SRAM and 4 kB EEPROM
Data Memory	10 kB	10 kB	32 MB SDRAM and 32 MB flash	128 kB flash
WIRELESS CHANNEL SPECIFICATION				
Radio	Integrated IEEE 802.15.4 radio	Chipcon CC2420	Chipcon CC2420	XBee modules
Frequency Band	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz, 900 MHz
Wireless Standard	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4/ IEEE802.15.1
Spread Spectrum	Yes	Yes	Yes	Yes
Outdoor Range	75 to 100 m	125 m		7 km
Enclosed Range	20 to 30 m	50 m		
Data Rate	250 kbps	250 kbps	250 kbps	Depend on the radio
FINAL ASSEMBLED ATTRIBUTES				
Dimensions	6.5x3.1x6 cm		4.8x3.6x0.75 cm	7.35x5.1x1.3 cm
Power Consumption				
Power Source	2X AA batteries	2X AA batteries	3X AAA alkaline batteries	Lithium battery (3.3V)

Table 2.5 - Summary of commercial wireless sensing unit prototypes (2005 – 2009)



(a) (b)

Figure 2.11 - Wireless sensor platforms: (a) Intel IMote2.0; (b) Libelium Waspmote

2.4 Embedded software and middleware

Systems based on wireless sensor networks do not rely only on hardware platforms. It is mandatory to have a good embedded operating system and high quality middleware services in order to achieve good results. There are several software solutions already developed and this sub-chapter discuss some of them.

2.4.1 Embedded operating systems

One of the most commonly used operating systems for SHM is TinyOS (www.tinyOS.net). It is open-source, has a large community and many successful implementations on wireless sensors. It utilizes a component-based architecture which is suitable for the use in smart sensors due to their extreme memory constraints.

This operating system employs non-blocking I/O, which means that it has only one memory stack. TinyOS support two types of executions: regular tasks and hardware event handlers. Tasks work in a FIFO manner, the order they are implemented is the order that they follow. Hardware event handlers on the other hand can pre-empt the execution of a task [42]. Due to being an event-driven operating system it has limitations at temporal level. One of the known issues is the uncertainty of the delay when start sensing. This phenomenon has to be address in order to synchronize sensing is to be achieved [43].

Applications for TinyOS are written in nesC which is based on C-language. Although it is used widely for wireless sensor networks, TinyOS is a challenging platform for non-programmers to develop network control and application software.

In the same line there is Mantis OS (Multimodal Networks of In-situ Sensors). It is a multithread embedded operating system for wireless sensor network platforms. It was developed for general purpose hardware and allows the performance of complex tasks even when they are time-sensitive [44].

Another operating system that has big impact is LiteOS. It is dedicated to sensor networks and was developed by the University of Illinois. The main feature of this operating system is the use of an object-oriented programming environment called LiteC which is based on C++ [45].

These are the main operating systems used for wireless sensor networks systems. Some hardware units can even handle some light version of Linux but the goal is always the same, provide a reliable base for developing applications on smart sensors.

2.4.2 Middleware Services

Before any damage detection routine can be implemented it is necessary to ensure certain features on the network itself. The existence of temporal synchronization inside the system is critical. It was already mentioned that tests with MICA Mote showed that interrogation algorithms present erroneous results when the data is not synchronized. Middleware services connect hardware to software and have to assure that the measurements made by the hardware components arrive to the damage detection algorithms in the best conditions.

There are a lot of studies on time synchronization protocols. However on this first approach the deeply study of these mechanisms is left out. One of the most complete overviews of these protocols is made by Sundararaman et al. (2005) [46].

2.5 Conclusion

The background studies here presented showed the importance of SHM systems. First it is discussed the structural health monitoring concept showing the main features and components. It is also mentioned the different types of tests performed in order to know the condition of a structure.

Wireless sensor networks are given as a solution to reduce the initial costs of SHM systems. The concept of these platforms is presented and all the subsystems discussed. The state of art of smart sensor platforms is made in the sub-chapter 2.3. It is presented the distinction between academic and commercial solutions and the most important devices were discussed.

After finishing this chapter it is possible to understand the advantages of using wireless sensor networks in SHM systems.

Chapter 3

SHM System Architecture

This chapter presents some of the most important requirements that an SHM system should fulfil. Defining the requirements is extremely important in order to manage the development process and understand what type of solution is needed. As this dissertation does not address any specific case of study, the requirements described here are suitable to most of SHM systems.

With all the requirements defined a conceptual design is proposed. This design does not take into account technologies and equipments availability, but cares only about how the final system should perform.

It is important to highlight that the goal of this dissertation is to validate the use of different technologies on SHM systems and not the construction of a final product. The development process is iterative and this research represents the first approach. Therefore it is possible that some needs are not fully addressed.

3.1 Requirements Analysis

The analysis and specification of requirements has two main objectives. First it is useful to understand well the needs that justify the development of the system. Secondly it produces a specification sheet where all the main features of the product are described. These features should be able to satisfy the needs previously defined. The specification serves then as guideline to the rest of the development process.

In order to determine these requirements it is necessary to enquire the various stakeholders on the project. On this project the major stakeholders are the Civil Engineers that work on the civil structures laboratory of oPorto's University (LABEST). Thus, most of the requirements here discussed are based on usage scenarios developed in partnership with these researchers.

Although there are many ways to classify requirements, in this document they are divided in functional and non-functional. This division tends to emphasize the separation between tasks that the system should be able to accomplish (functional requirements) and the properties and constraints that the system should respect (non-functional requirements).

3.2 Functional Requirements

The functional requirements of a product arise from the study and analysis of different needs and goals. As already mentioned the main stakeholders of this research are civil engineers, therefore the requirements described are based on their necessities.

In order to express these requirements in a methodical manner they were given an identifier flag and a priority classification. The requirements with "mandatory" tag are the most important and have to be accomplished. From that point on and with decreased importance there are the "desirable", "possibly useful" and "luxury" tags. The following table shows the main functional requirements defined.

ID	Requirement	Priority
Func-1	The system has to be able to acquire data from different types of sensors;	Mandatory
Func-2	The system has to be able to store the acquired data;	Mandatory
Func-3	All the information acquired has to be able to be transmitted to a remote control post;	Mandatory
Func-4	The system has to be able to show the information obtained on a human-machine interface;	Mandatory
Func-5	The system has to be able to make measurements when asked to;	Mandatory
Func-6	The user should be able to select the task that the system is to perform;	Mandatory
Func-7	The acquired data should be processed by the monitoring system in order to be presented as relevant information about the state of the structure;	Mandatory
Func-8	The system should have multiple measurements tasks like: • Monitor the structure for an interval of time; • Read the actual sensors values; • Activate and deactivate sensors; • Activate and deactivate actuators; • Acquire data continuously;	Mandatory
Func-9	Have an alarm history related to damage detection;	Desirable
Func-10	The system should be connected to a data base that can be accessed through the World Wide Web;	Possibly Useful
Func-11	The information presented on the human-machine interface should also be available on a website;	Luxury
Func-12	Besides monitoring the structure the system should be able to control actuators;	Luxury

Table 3.1 – Functional Requirements

The functional requirements analysis is not finished until all the requisites are explained and justified. Therefore all the items in Table 3.1 are carefully described in the next paragraphs.

Func-1: The system has to be able to acquire data from different types of sensors

As it has been explained in chapter 2 structural health monitoring systems have to be able to measure a lot of different phenomena. Therefore, the data acquisition module has to be able to extract data from a big variety of sensors. The most important measurements are usually acceleration and strain but it depends greatly on the type of structure and the purpose of the study. In addition, measures of velocity, displacement, temperature, humidity, wind velocity and wind direction are sometimes really useful to understand the behaviour of the construction [47].

On this project the measures chosen were acceleration, strain and temperature. Nevertheless the system has to be developed taking into consideration that there may be necessary to add some other types of sensors in the future.

Func-2: The system has to be able to store the acquired data

Another mandatory requirement arises from the need to store all the measurements made. The amount of the data to be saved depends on sampling frequency, number of sensors, resolution of the analogue-to-digital converter and time interval that the measurements last. For most of the phenomena Civil Engineers want to study sampling frequencies of about 10 Hz will be enough. However if dynamic vibration analysis is pretended this value can rise to 100 Hz easily. After some discussion it was decided that strain and temperature measurements could be done with sampling frequencies of 1 Hz or even individual readings and acceleration should be acquired with at least 100 Hz.

As the number of sensors and resolution of ADC were not yet defined, it was accorded a minimum value for storage capacity of 1Giga bytes. This allows the storage of 8 million measures with 10bits resolution.

Func-3: All the information acquired has to be able to be transmitted to a remote control post

This requirement comes from the need of a single station where all the information goes to. The main characteristic of the system affected by this requisite is the bandwidth of the communication module. If the traffic load of the network becomes too high data collision will start to occur. This will generate more traffic to a point that can even surpass the network's communication capacity.

To maintain reliability and efficiency of communication the transference of data should be well managed in order to optimize the usage of the bandwidth available.

Func-4: The system has to able to show the information obtained on a human-machine interface

As Civil Engineers are not experts in computer sciences a simple and functional interface has to be developed. This feature should allow an easy access to all the information gathered. The functionalities of this interface were not fully defined giving leeway for the design stage.

Func-5: The system has to be able to make measurements when asked

The user has to be able to start the measurement process when he wishes to. This may not be possible all the times, especially when the system is already running some kind of measurement task already. Thus, in for a first approach it was defined that the user has to be able to start any kind of task when the system is in idle mode.

Func-6: The user should be able to select the task that the system is to perform

The task selection has to be controlled by the user. This requirement fits as a mandatory functionality for the human-machine interface. There is no need for more justification as it is obvious that if the user is able to start a measurement task he should also be able to choose what type of task he wants the system to perform.

Func-7: The acquired data should be processed by the monitoring system in order to be presented with the correct units

The data acquisition system usually converts electrical signal from sensors to a value in bits related to its resolution. This output has to be converted to the real units that express the magnitude of the phenomenon that is being measured.

For future development there is also an interest in aggregate and process all the data in order to reduce traffic load on the network. This feature relies on algebraic operations such as averaging, numerical filtering, resampling, among others.

Func-8: The system should have multiple measurements tasks

The main tasks that the system should be able to perform are reading sensors during an interval of time and acquire the sensors value punctually. Monitor the structure during a period of time is important for dynamic tests. Dynamic phenomena can only be studied if the quantities in question are measured for some time. Being able to read punctually the sensor is also needed for monitoring static field tests. An example is the monitoring of cracks in infrastructures.

There is some interest on reading all the sensors in a continuously manner. However this feature would need an excessively large amount of storage capacity and bandwidth. At this first approach this feature has been neglected.

Other features like sensor calibration and traffic monitoring are also interesting and make SHM systems more attractive from a cost/benefit perspective.

Func-9: Have an alarm history related to damage detection

One of the tasks that the system should be able to perform automatically is the damage detection. There are different algorithms that output different types of information. As it was described in chapter 2, SHM systems can go from detecting damage to understand the influence of the damage on the integrity of the structure.

The study and implementation of these algorithms is not under the scope of this dissertation. For this reason it was determined that the system should alert the user when any measure exceed the threshold values defined.

Func-10: The system should be connected to a data base that can be access through the World Wide Web

The goal behind this requirement is to enable access to the information from any point of the world. Thus, it is possible to avoid the dislocation of qualified technicians to the site. This will reduce the cost and increase the speed of a project related to structural health monitoring.

Func-11: The information presented on the human-machine interface should also be available on a website

The feature here presented is not as important as the previous ones. Some structure owners would find interesting to be able to check the state of their constructions on a website. Also if the infrastructure under study is part of the transportation network there may be identities that would be interested in knowing if the construction is in comply with the applicable laws.

Func-12: Besides monitoring the structure the system should be able to control actuators

This requirement only makes sense for structures that have actuators installed. They can be used to excite the infrastructure in order to get better results about its dynamic behaviour. If the infrastructure belongs to transportation network it might have semaphores that can be controlled by the SHM system, which can close the road when damage is detected.

The main functional needs of LABEST researchers have been presented. However there may be some requirements that were not fully studied and may be addressed in a second iteration.

Next, the non-functional requirements are described. These should give a better idea about the direction of the project. Afterwards with all the requirements studied it will be possible to define the system’s architecture and determine which technologies should be tested and validated.

3.3 Non-functional Requirements

Other than the tasks the system should perform it is necessary to understand the conditions on which they are accomplished. One of the biggest problems that Civil Engineers face is the amount of cables used by traditional instrumentation systems. For this reason the biggest non-functional requirement (can be seen as a constraint) is the use of wireless sensors. Thus all the following requisites take into account that at least one part of the system should apply wireless communications.

In contrast with the last subchapter, non-functional requirements are first discussed and then a summary table is presented.

3.3.1 Scalable resource aware system

As it was described in chapter 2, it is possible that a SHM system might integrate a large amount of sensors in order to detect the location of damage. Structure damage is usually a local phenomenon and it is expected that readings from sensors near the damage contain more information about the disturbance than those remotely located. The variation between the damage zone and the safe zone can only be accurately assessed if sensors are distributed in a satisfactorily dense manner.

One of the advantages of using wireless sensor nodes is that such a vast number of transducers are economically possible; however, the limited resources provided by this type of nodes have to be well managed. In order to achieve a scalable system one has to take into consideration the power, bandwidth, memory and computational power that a wireless node offers [48].

3.3.2 Distributed data computing

In a system so complex where such a high amount of information has to be transmitted all the resources available have to be explored to a maximum. However it is presented here as a requirement, distributed computing, can also be seen as a solution. As it was already described smart sensors feature a computational core which means they are capable of some processing. This characteristic should be used in order to aggregate data, balance the power consumption among nodes and increase the flexibility of the system.

Data aggregation is important to reduce the amount of data that needs to be sent through the wireless network. It was already mentioned on the functional requirements analysis that the bandwidth should be sufficient to send all the data acquired to the remote post. However if the number of sensors increases it may not be possible. This is one of the reasons why data aggregation might be important.

Moreover without distributed processing some nodes would have to compute a lot of the acquired data on their own. With their limited computational power this would decrease the efficiency of the system. Also data processing consumes a high amount of power. Distributed processing will balance the power consumption of all the nodes [48].

3.3.3 Fault tolerant system

A system with a large amount of devices can suffer from nodes failure. If a smart sensor stops working the network should reconfigure itself in order to keep working. Thus only information acquired by the failing node is lost. At this stage this requirement is not mandatory; however, it should be address on the final system design.

Another source of failure is the malfunction of a sensor. In this case it is advisable to have an algorithm that detects faulty data and does not include it when determining the condition of the structure.

In a large network there is also the possibility of losing message packets. The system should be able to detect this problem and resend lost data. However, resending packets will increase the bandwidth use and power consumption. Hence, data loss tolerant algorithms are appealing [48].

3.3.4 Modal operation system

Multiple operation modes are attractive in order to facilitate power management. Wireless sensor nodes should be able to enter in sleep modes in order to save energy when no significant event to observe is expected. Transition between modes should be triggered or by alarms predefined or by user intervention.

3.3.5 Desirable characteristics for damage detection algorithms in civil infrastructures

Damage detection algorithms are out of the scope of this dissertation. Nevertheless it is important to enumerate some of their desirable characteristic for future studies.

A good damage detection algorithm should be able to work with data coming from different types of sensors measuring different physical quantities. It is expected that the more information the algorithm can relate, the more accurate its output is.

As it was explained before transmission of all the data acquire through the entire network is not desirable. However, sensor nodes should collaborate in local communities exchanging relevant information. Algorithms which can relate information that arrives from different sections of the structure are able to determine the extent of the damage more accurately.

3.3.6 Environmental constraints

Often wireless sensors nodes placed on structures are subjected to severe environmental conditions (e.g. wind, rain, sunshine, extreme temperatures and vibration). Therefore suitable packaging is advisable. If they are not robust enough, sensors exposed to this kind of conditions, may provide faulty data or even stop working.

In some structures the place where nodes are deployed should also be chosen carefully. The instrumentation system should not interfere with the normal behaviour of the structure. If this happens all the measurements should be analysed taking into consideration the monitoring system.

ID	Requirement	Priority
Operational-1	The system should use wireless sensor nodes;	Mandatory
Operational-2	The system should be scalable;	Mandatory
Operational-3	Data should be processed in a distributed fashion;	Mandatory
Operational-4	Low cost system	Mandatory
Operational-5	The system should be fault tolerant;	Desirable
Operational-6	Low power consume mode should be possible;	Desirable
Operational-7	Damage detection algorithms should be able to relate information about different kind of phenomena;	Possible Useful
Operational-8	Local sensor communities should be collaborative;	Possible Useful
Operational-9	Sensor nodes should be rugged in order bear severe environmental conditions;	Luxury
Operational-10	The instrumentation system should not interfere with the structure behaviour;	Luxury

Table 3.2 – Non-functional Requirements

This analysis may not be complete and some other requirements can be found in a second iteration. However, for now these will be the guidelines that should be followed. There are many difficulties that need to be addressed and technologies validation is an important part of this dissertation. It is imperative to emphasize once again that the main goal of this research is to test solutions presented in the literature as the future of SHM systems, mainly wireless sensor networks.

3.4 Proposed Architecture

A SHM system architecture designed to address many of the requirements described in the previous subchapter is proposed here. The overall mode of operation is initially described and the different subsystems are explained in more detail. At this point the used technologies are not discussed.

3.4.1 Global System Operation

The system is based on a hierarchical architecture with distributed processing divided by three control units. On a higher level there is the Remote Control Post, which is in charge of managing the information gathered and display it on the human-machine interface. It should also update the data base where the acquired data is stored. Furthermore it is from this point that orders are sent to the rest of the processing modules.

The intermediate level has two processing units. These are attached to gateways that make the gap between the processing plant network and the floor network. They are also responsible for coordinating the wireless communications with the low level units. Using wired communications for the top level enables an increase of bandwidth and reduces the power restrictions on these units. In order to take advantage of this fact data computing should be mainly realized by this subsystem.

Wireless sensor nodes that make part of the system lower level incorporate all the signal conditioning circuits to connect the sensors. Their task is to acquire information from the structure and transmit it to the intermediate level units wirelessly. Some data processing can be assigned to these nodes. However, it is important to remember that these units are supplied solely by batteries. Therefore power restrictions should be taken into consideration.

Next the functional diagram is presented in Figure 3.1 and the subsystems are discussed in more detail.

3.4.2 Network Architecture

The system proposed employs two types of networks. The upper level network is wired and should be able to transmit data over 500 meters. Deployment in big structures where sensor communities can be separated for some hundreds of meters should be possible. This network is expected to have higher bandwidth availability in order to transmit large quantity of information. It is possible to use wireless communications to transmit over big distances however this is not wise from an energy consumption point of view. The energy needed for this type of communications is excessively large for nodes powered by batteries [13]. Another approach would be installing more nodes and use multi hopping techniques. Still the use of a wired data transmission enables the validation and test of mixed network solutions.

The dotted lines in Figure 3.1 represent the wireless network. A star architecture is employed in order to simplify the implementation stage. Also with only three end nodes available it is almost impossible to use more elaborate architectures. Unlike the upper network, the communication distance should be restricted to less than 100 meters.

Inspecting the network topology as a whole it is possible to find a lot of similarities with the two-tiered model presented by Mitchell et al. (2002) [23]. The upper nodes are able to execute long-range communications and can have more energy dependent tasks allocated. The use of a wired network implies that these devices are supplied from the local grid. The end devices are only responsible for close range communications.

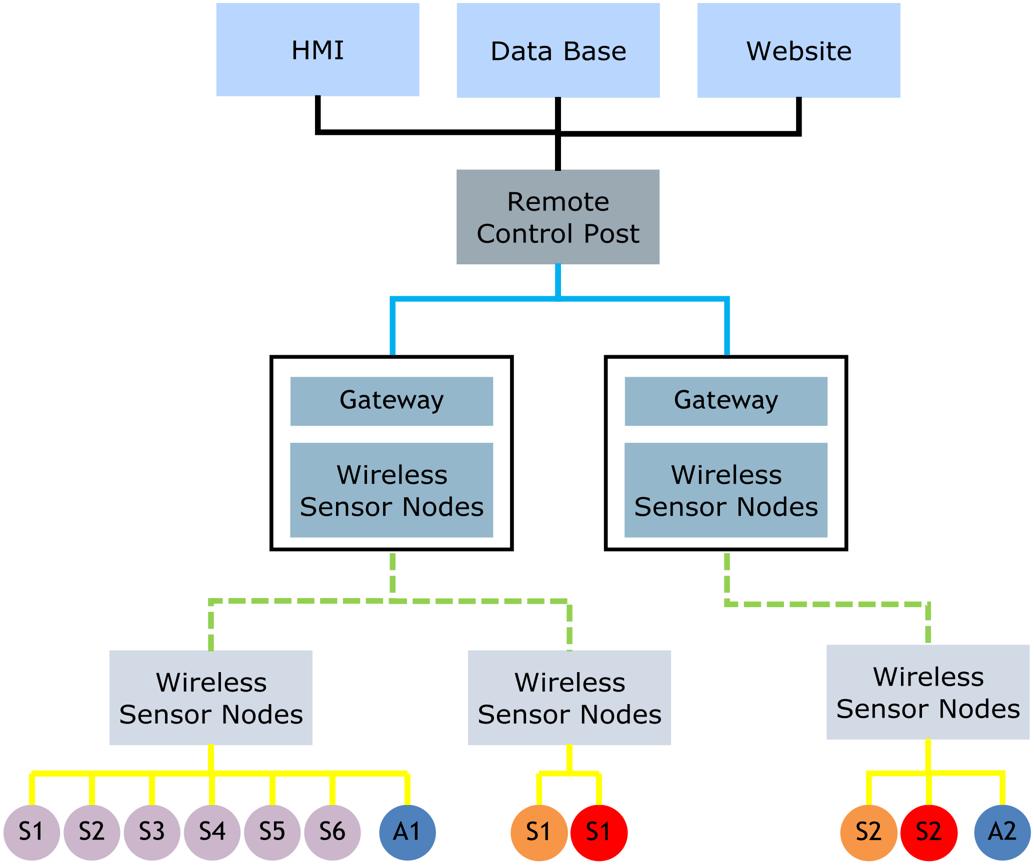


Figure 3.1 – Functional Diagram

3.4.3 Remote Control Post

Remote Control Post is the master processing unit of the architecture proposed. It is in charge of managing the interface and storage modules. When a user interacts with the system using the HMI or website, this unit is responsible to direct the received commands to the devices of lower hierarchy. From a communication perspective it is the system master.

On a first approach this module can be used to do some more intensive calculation like FFTs and digital filters. However for large quantities of data a distributed processing solution should be applied. Although it is not the objective of this analysis to discuss specific technologies, in order to clarify its characteristics one can compare this module to a common PC.

The Human-Machine Interface (HMI) is implemented on the remote control post and should enable the operator to control the entire system. The HMI should display in a clear way the overall condition of the structure, the points of the structure where damage is presented, a list of alarms, the results given by sensors, and the state of different actuators like semaphores.

The web server is also implemented in this unit; it should communicate internally with the interface. This is the main storage facility of the system. All the data received by the HMI should be transmitted to the data server in order to allow future analysis. The website is connected to the web server and should display all the relevant information about the integrity of the bridge.

As it was stated before the study of this high level unit is out of the scope of this document. Although an overall description of the technologies used to implement is presented in chapter 4.

3.4.4 Intermediate Level Nodes

These are the more complex nodes as they have to be able to make the bridge between the upper and lower level networks. Furthermore, they should coordinate the wireless communications and be able to process some of the data gathered.

From a global perspective these nodes work like slaves for the Remote Control Post and masters for the end devices.

In order to achieve a homogeneous configuration for the low level part of the system these nodes have the same characteristics of the end devices. This way it is possible to reassign different roles to all the devices improving the robustness to node failure. However, the use of the same hardware to manage the bridge between both networks and to execute the end modules tasks is not possible. Hence a second controller connected to the wireless sensor node is part of design. It is possible to see that distinction in Figure 3.1.

3.4.5 Lower Level Nodes

Wireless sensor nodes belonging to the lowest hierarchical stage of the system are responsible to acquire data from sensors and control actuators. As it was said before, these devices have the same hardware than the intermediate nodes. Yet, because some sensors need specific circuits to be connected to the existent data acquisition system, more hardware has to be added. Conditioning circuitry for each type of sensor has to be plugged to the original node platform. If this extra hardware composed mainly by conditioning circuitry and analogue filters is powered by the nodes batteries, energy consumption should be taken into consideration. Also, in this case, the controller device should be able to turn on and off these circuits as it is needed.

The main desirable characteristic of these platforms, other than low consumption, is having analogue-to-digital converters with enough resolution and sampling frequency to measure accurately the physical magnitude intended.

Taking into consideration the full architecture these end devices should always work as slaves in the Master-Slave cooperative model that is applied.

3.5 Conclusion

On this chapter, desirable characteristics for SHM system that employ wireless communications are described. The architecture used in this dissertation is also discussed from a functional level. As one can imagine the different communication protocols used and the need of connecting them together result in a complex system. One of the advantages of this architecture is the possibility of testing several technologies in order to validate some as possibilities for a second iteration of the development process.

Next chapter approaches technologies adopted and discuss some alternatives.

Chapter 4

Technologies Discussion

The study of different technologies applied to structure monitoring constitutes an important objective of this dissertation. Thus, the criteria and justification for technological choices made during the development stage are here presented.

This chapter is divided in sub-sections in representation of the modules previously described. Sensors sub-system is the first being presented followed by the wireless sensor node section. To conclude, a description of the communication protocols employed is made.

4.1 Most Common Sensors in SHM

A sensor is a device that is sensitive to a physical phenomenon and converts it into a signal which can be read by an instrument. In a monitoring system, sensors represent the interface between the physical world and the world of electrical devices [49].

As such, before starting to describe the different types of sensors it is necessary to define the physical quantities of interest. When monitoring civil structures it is possible to distinguish between two types of quantities. Responsible for the global characterization of the infrastructure there are global displacements, deflections, rotations and forces. On other hand giving information about local behaviour there are strain and local displacements. Although these are the traditional magnitudes of interest, recently vibration has appeared also as an important source of information about the dynamic behaviour [50].

In some cases, data regarding the environmental condition of the place where the structure is inserted is important. For example, measurements made during an ambient vibration test are only accountable if information about the environment is available. Also some sensors are affected by climacteric phenomena and their output has to be corrected according to the surroundings condition. Usually, weather stations are installed to monitor temperature, wind speed and wind direction.

The choice of sensors depends not only on the quantities to be measured. It is necessary to study the characteristics of these components in order to understand whether they are in fact appropriate for the job. Following are some of the most important sensor characteristics:

- *Transfer Function:* It shows the functional relationship between the physical input and the electrical output signal. This characteristic is usually represented by a mathematical equation that relate the input with the output;
- *Sensitivity:* Generally represents the ratio between changes in electrical output to small changes in the physical input. It may be expressed as the derivative of the transfer function in respect to the input. One can say that a sensor has high sensitivity if a small variation on its input result in a large output variation;
- *Span or Dynamic Range:* It defines the range of input physical signals that may be accurately converted to electrical signals. Usually the performance characteristics described in the data sheets are only applicable inside the sensor dynamic range;
- *Accuracy or Uncertainty:* Uncertainty is generally defined as the largest expected error between actual and ideal output signal. It is usually represented in percentage of the actual reading;
- *Nonlinearity or Linearity:* Defines the maximum deviation from a linear transfer function. There are several measures of this characteristic, the most common compares the actual function transfer with the “best straight line”;
- *Signal-to-noise Ratio (SNR):* All sensors output some noise in addition to the measurement signal. The SNR indicates the ratio between the signal power and the noise power corrupting the signal. This characteristic is usually described in the logarithmic decibel scale (dB).
- *Resolution:* It is defined as the minimum detectable input signal fluctuation. It describes the smallest increments of input, which can be sensed. Although, it is sometimes applied to analogue sensors, this characteristic is more important for sensors with digital outputs.
- *Bandwidth:* All sensors have a finite response time to an instantaneous change in the input signal. Bandwidth represents how fast the sensor can detect a change in the input. Usually it is represented by the upper and lower cut-off frequencies.
- *Thermal Drift:* Sensors usually exhibit different behaviours at different temperatures. This characteristic refers to the influence that temperature has in the sensor transfer function.

Some of these characteristics are used to discuss which sensor should be used for this application. It is important to note that sometimes manufactures do not give information about all these characteristics.

To conclude this introduction it should be reminded that only strain, acceleration and temperature are going to be measured. These are considered some of the most important physical quantities to determine the condition of the structure and are suitable to validate the design proposed.

4.1.1 Strain Measurement

The devices used for measuring strain are designated by strain gauges. For infrastructures monitoring, these sensors can be applied at the surface or embedded in the material. The first ones are suitable for masonry, wooden and steel structures. They can also be used in concrete although in this is case is preferable the use of embedded strain gauges. Embedding sensors is a process that can only be executed prior to the concreting stage. So if the aim of the system is to monitor a finished construction only surface application gauges can be used [50].

Speaking of a measuring point in strain measurements is unrealistic. This is because strain is actually measured across a segment. The segment length should be selected taking into consideration the characteristics of the structural material, specially its heterogeneity. In more homogeneous materials such as steel, smaller gauges can be adopted. In the case of most heterogeneous materials, like concrete, the length of the segment depends on the size of the aggregates used [47].

There is a wide variety of gauges based on both electrical signals and fiber-optics. In this section vibrating strings, resistive strain gauges and fiber-optics sensors are discussed.

4.1.1.1 Vibrating Strings

Vibrating strings sensors are based on the fact that there is a relationship between the vibration frequency of a string and the tension it is subjected to. Typically, a steel string with a diameter in the order of 0,3mm is installed in a capsule and fixed by its edges. The string is stretched due to the tension applied. A variation of this tension will represent a variation on the vibration frequency. In order to start oscillating the string has to be excited. This process is accomplished by the placement of a coil close to the string. When a current passes through the electrical coil a magnetic field is created which excite the string. The string vibration inside the magnetic field induces a current in the coil which is transmitted by a meshed cable. Therefore, the device used for the excitation process is also in charge of the acquisition of the output signal (Figure 4.1).

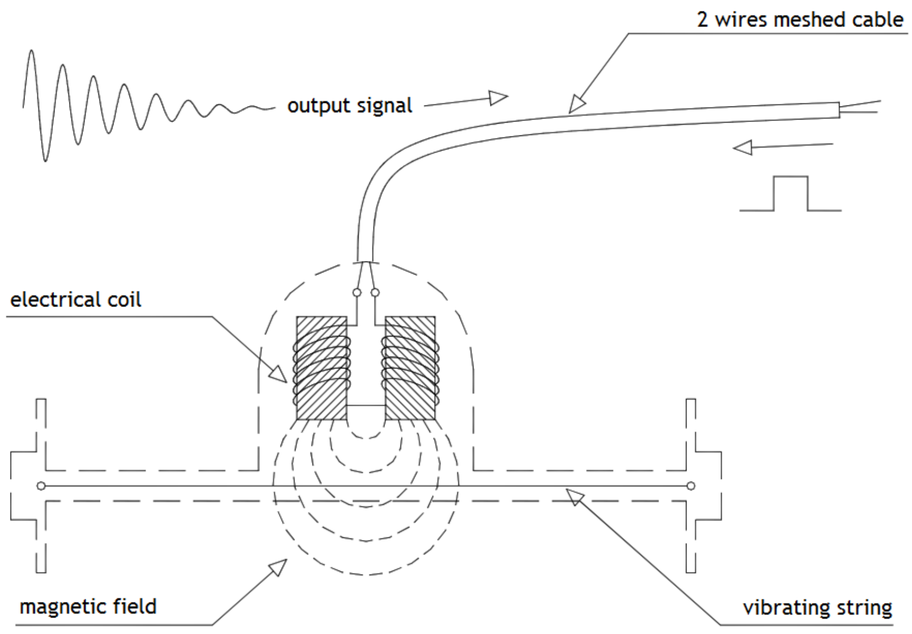


Figure 4.1- Operation diagram of a vibrating string strain sensor (adapted from [50])

The vibration frequency of the string can be converted to deformation by the following quadratic equation:

$$\Delta\varepsilon = \varepsilon_f - \varepsilon_i = K_v(f_f^2 - f_i^2) \tag{4.1}$$

In equation (4.1), f_i is the initial oscillation frequency and f_f the final one. Therefore, these sensors need calibration after they are installed because all the measures made will always be related to the initial state of the string. The K_v constant depends on the mechanical and geometrical characteristics of the sensor. This value should be supplied by the manufacturer. The strain variation is thereby given by the difference between the two different oscillation frequencies.

This type of sensors is quite bulky compared to the resistive strain gauges and their typical length can vary from 140 mm to 250 mm. Vibrating string gauges can detect extensions in the order of $1 \times 10^{-6}\ m/m$ and have a dynamic range of $3000 \times 10^{-6}\ m/m$. The signal attenuation along the cable that connects the sensor to the data acquisition system is quite small enabling the use of cables with 2000 m. This happens mostly because most of these sensors transmit the signal in current instead of voltage [50].

Temperature also affects these sensors; usually they present a 0.02% uncertainty for a maximum variation of 10 °C. For most of the measurements this inaccuracy is tolerable however in zones where big temperature variations occur this value should be monitored in order to be possible to offset the results afterwards.

The values here presented may vary from manufacturer to manufacturer; however, they can be considered the typical market values. Next image shows two types of vibrating string gauges, the left one is for surface mounting and the other for embedding in concrete.

Figure 4.2 - a) Vibrating string gauge for surface application, b) Vibrating string gauge for embedding in concrete

4.1.1.2 Resistive Strain Gauges

Another type of strain transducers that have high visibility in the field of structural monitoring are the resistive strain gauges. The basic principle behind their operation is the fact that electrical resistance of a material is sensitive to mechanical deformation. The most common materials used in this kind of sensor are semiconductors and metallic conductors (wire or foil). If in semiconductors gauges the way how strain changes the resistance seems to be well understood, there is still a long way to go before the same phenomenon is fully comprehended on metallic materials [51].

Before the discussion of several devices available in the market the definition of strain gauge sensitivity should be made. When a conductor is stretched in the axial direction, its length and cross-sectional area will change. As shown in Figure 4.3 the increase of length is accompanied by a decrease in the cross-sectional area, and vice versa. Also, the specific resistivity of the material may change. From the combination of these three characteristics results the electrical resistance of the conductor (equation

(4.3)). The strain sensitivity factor, or gauge factor, is given by the relationship between the resistance variation and the change in the length of the conductor.

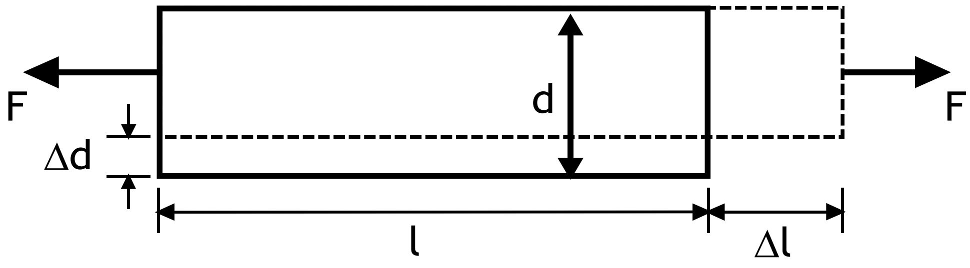


Figure 4.3 – Schematic diagram of stretched conductor

For a straight conductor of uniform cross section the strain sensitivity is expressed by the following equation [51].

$$K = \frac{\Delta R/R}{\Delta l/l} = \frac{\Delta R/R}{\varepsilon} \tag{4.2}$$

where:

- K = strain sensitivity factor of the conductor which is dimensionless and represents a physical property of the material;
- R = resistance in ohms;
- l = length in millimetres;
- $\Delta R, \Delta l$ = corresponding changes in resistance and length;
  $\varepsilon = \Delta l/l$ = strain along the conductor (dimensionless).

However and as Figure 4.3 shows the cross section of a metal wire is not constant. Furthermore the material resistivity may also vary with the tension applied which leads to a more complicated set of equations [51].

$$R = \frac{\rho l}{A} = \frac{\rho l}{\pi r^2} = \frac{4\rho l}{\pi D^2} \tag{4.3}$$

$$dR = \frac{4l}{\pi D^2} d\rho + \frac{4\rho}{\pi D^2} dl - \frac{8\rho l}{\pi D^3} dD \tag{4.4}$$

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dl}{l} - \frac{2dD}{D} \tag{4.5}$$

Then

$$K = \frac{dR/R}{dl/l} = \frac{d\rho/\rho}{dl/l} + 1 - \frac{2dD/D}{dl/l} \quad (4.6)$$

Since

$$\mu = -\frac{dD/D}{dl/l} \quad (4.7)$$

is defined as the Poisson's ratio (μ), comes

$$K = 1 + 2\mu + \frac{d\rho/\rho}{dl/l} \quad (4.8)$$

From the last group of equations, ρ represents the resistivity of the material, D the diameter of the conductor and l the length. As it has been stated these values depend on the material used. In general, metals have sensitivity factors between 2 and 4, while semiconductors present 150.

If the sensitivity gauge factor is considered a constant for all the measurements range, it can be assumed that resistive strain gauges vary linearly with strain. In many cases for civil structures monitoring this assumption can be made [47]. Next different types of resistive gauges are described in detail.

Wire Strain Gauges

One of the early wire gauges was the unbounded type. These instruments have the strain-sensitive wire mounted under tension on mechanical supports. A slight motion of the support indicates a change in strain hence a variation in the electrical resistance. Because they are not attached directly to the material under study some discrepancies may arise. The mechanical structure that supports the wire has to be accounted and sometimes its influence in the measure is not easy to determine.

Bonded wire strain gauges were a major improvement in this field. The difficulties raised by the mechanical support were eliminated by bonding a very fine strain-sensitive wire directly to the surface on which strain is to be measured. The filament has to be electrically insulated and the bonding perfect, which difficult the application of these sensors. Furthermore the wire used has to be ultra thin (0,0254 mm) thus, the shearing stress in the cement or other bonding agent will be considered too high to permit faithful measures [51].

Nowadays, the strain-sensitive wire is premounted on a suitable carrier that can be attached easily to the surface of the material in study. This way satisfactory installation, giving good and consistent results is possible in less time.

To get wire resistances in the order of some ohms it is necessary to use wire lengths of a couple of millimetres. As the desired gauge length is almost always less than the required length of the wire it is necessary to arrange it in some kind of grid. There are several grid configurations however the flat grid is probably the most used (Figure 4.4). The materials used for the strain-sensitive wire are usually Nichrome or Isoelastic alloys [51].

This type of gauges has been replaced by foil strain gauges. Considered the most common in SHM, these sensors are described in the following section.

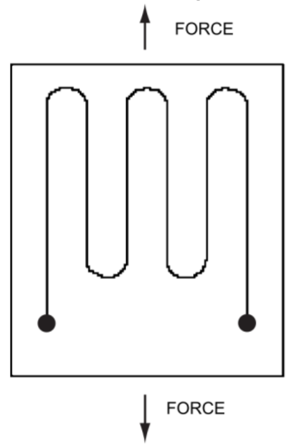


Figure 4.4 – Bounded wire strain gauge with a flat grid topology (adapted from [51])

Foil Strain Gauges

The foil gauge operates in a similar way to the wire gauge. However, the sensing element consists of very thin metal foil (about 0,00508 mm thick) instead of wire. These devices are produced by photo-etching techniques and use the same type of metals used in wire strain gauges. The most commonly used materials are: alloys of Copper-nickel (Constantan), Nickel-chromium (Nichrome), Nickel-iron and Platinum-tungsten [52]. Because the type of material used is one of the principal components, which determine the operating characteristics of a strain gauge, a brief discussion of the most important alloys is made.

Constantan is one of the oldest and still most widely used alloys for strain gauges. This situation is due to the fact that Constantan has the best overall combination of properties needed for most of the applications. It has an adequately high sensitivity gauge factor that is relatively insensitive to temperature changes and strain level. Its resistivity is high enough to achieve suitable resistance values in small grids. This material exhibits good durability and a relatively high elongation capability. Also self-temperature-compensated alloys based in Constantan are available. A negative point for this material is the continuous drift that it exhibits at temperatures above 65 ^{o}C (Figure 4.5). This characteristic should be taken into consideration when gauge stability is critical over big periods.

In order to measure large strains, over 50 000 με annealed Constantan is the grid material usually selected. This type of constantan is very ductile and allows gauge lengths of 3 mm or more, it can be strained to more than 20%. The problem arises when cyclic measures want to be made. This alloy exhibits some permanent resistance change which causes a zero strain resistance shift. Therefore, the sensor must be calibrated before each measure.

The Isoelastic alloy composed mainly by Nickel and Iron is preferable when purely dynamic strain measurements are to be made. In other words when it is not necessary to maintain a stable zero reference this alloy is advantageous. It has superior life time compared to Constantan and a higher gauge factor (approximately 3,2), which improves the sensor sensitivity and signal-to-noise ratio characteristic. The major problem is its thermal drift, which is rather high (about 145 με/^{o}C). Therefore, the use of this material is constraint to applications where high output is needed [53].

Nickel-chromium or modified Karma alloy is the last strain-sensitive material under discussion. It has an excellent stability and a good fatigue life. It is the preferred choice for accurate static strain measurements over long periods of time. One of its advantages is the thermal output curve exhibited (Figure 4.5). Like Constantan, modified Karma can be self-temperature compensated.

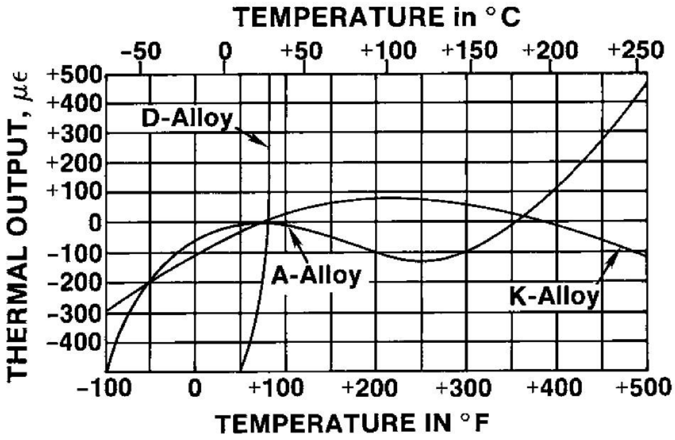


Figure 4.5 – Thermal output for different materials. A-Alloy: Constantan; D-Alloy: Isoelastic; K-Alloy: Modified Karma [53]

As it was stated before the metallurgical properties of certain strain gauges alloys (Constantan and modified Karma) can be processed in order to minimize the thermal output over a wide temperature range. However, in order to achieve good results these gauges have to be bonded to materials with thermal expansion coefficients for which they are intended. Therefore to choose a proper strain-sensitive alloy the type of material under test has to be considered.

Typically foil gauges exhibit a slightly higher gauge factor and lower transverse response than their equivalent in wire. Because they are thinner, they adapt more easily to the specimen surface, which makes them easier to install. Also their greater contact area permits better heat dissipation and reduces shearing stress in the bonding agent. This results in strain gauges that are solely subjected to the deformation of the material, which leads to longer life times [51].

Depending on the carrier, the alloy used and its metallurgical condition, foil gauges will measure with precision deformations in the range of 10 to 15 %. They are able to perform 10 million measures at strains of ± 1500 με. There are several different grid topologies available in the market and the carriers are usually made of paper, epoxy, phenolic, glass, reinforced resins and other plastics [51]. For SHM systems foil gauges with resistances of 350ohms or 120ohms are usually preferable. These values are also considered the typical for this type of sensors. Next figure shows the schematic of a typical foil strain gauge.

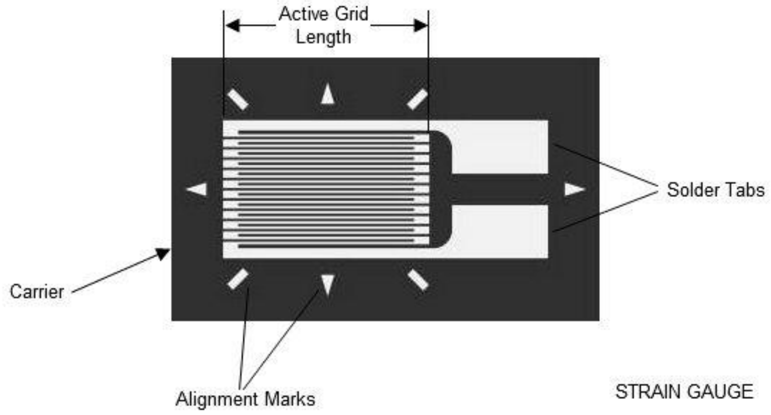


Figure 4.6 – Schematic of a typical foil gauge

As all the other resistive strain gauges, foil gauges can be applied on the surface or embedded. For concrete embedding the active grid length should be long enough to make sure that the strain measure is due to real deformation and not because some local material discontinuities [47].

Semiconductor Gauges

Semiconductor gauges are not as widespread as foil gauges however in certain applications they might be the right choice. These sensors use the piezoresistive effect present in certain semiconductor materials, such as silicon and germanium, to obtain greater sensitivity.

The main attraction of the semiconductors is the high gauge sensitivity. This means that is possible to precisely measure small strain variations, without the need of intermediate amplification. In contrast with these advantages semiconductors offer a whole lot of disadvantages. They exhibit a nonlinear transference function regarding their resistance and the strain to measure. Actually this function has a large number of variables involved that need to be monitored in order to understand the measured value. Another problem arises due to large gauge sensitivity. During the installation process the gauge resistance can suffer from minor changes, with a strain sensitivity of 150 this change will have a terrible effect on the output. Therefore, it is necessary to determine the gauge resistance following installation. Hence, if necessary, an appropriate correction can be made *a posteriori*. Temperature changes are also critical on this kind of gauges. Actually not only the material resistance changes drastically as also the gauge sensitivity vary with temperature.

Due to all this problems, it is not expected that semiconductor gauges will replace metal gauges for purpose of strain analysis. However in some specific cases these devices can work extremely well [51].

4.1.1.3 Strain Gauges Based on Fiber-optics

SHM systems have been increasingly used on Civil Engineering. Within this development sensors based on optical fiber represent an important breakthrough. This dissertation does not intend to study these instruments in great detail; however, with their increasing visibility in structure monitoring it is imperative to describe the main concepts. The main reason for ruling out this type of technologies from the start is because it is not easy or cheap to integrate optical fiber sensors on wireless sensor platforms.

As metal wire, optical fiber can be used as a transducer to measure strain, temperature, pressure and other quantities. This can be achieved by modifying a fiber so that the quantity to be measured affect the intensity, phase, polarization, wavelength or transit time of light that passes through the filament.

In structures monitoring the main type of sensors used are Bragg sensors and Fabry-Perot sensors. Bragg sensors rely on the relationship between the wavelength of the light signal and the strain that the optical fiber is subject to. While Fabry-Perot sensors are based on the change in phase of the light that goes by a strained fiber [54]. The main principles of these two phenomena are explained in the next paragraphs.

Bragg Sensors

These devices are based on optical fiber wavelength modulation, which changes according to the strain that is applied to the fiber.

Actually, a fiber Bragg grating can be used as an optical filter that reflects only certain wavelengths. This phenomenon is achieved by adding a periodic variation to the refractive index of the fiber core that generates a wavelength specific dielectric mirror. The mirrors created reflect only light with a wavelength that is proportional to the space between them [55]. The next diagram illustrates Bragg sensors operation mode (Figure 4.7).

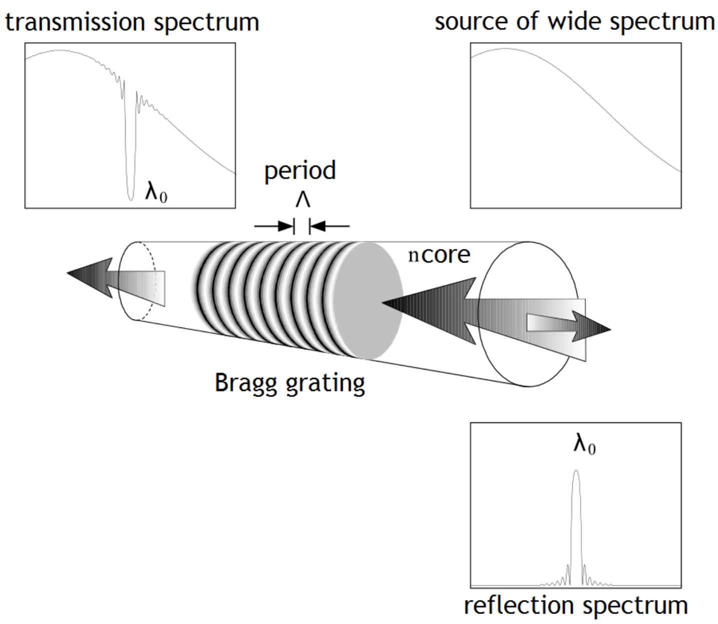


Figure 4.7 – Operation mode of Bragg sensors (adapted from [50])

For a normal incidence light beam the diffraction law established initially by Bragg can be given by the following equation:

$$\lambda_0 = 2 \times \eta \times \Lambda \tag{4.9}$$

where:

- λ_0 = reference value of the wavelength reflected;
- η = effective index of grating refraction;
- Λ = Bragg grating modulation period.

As Figure 4.7 shows the Bragg grating modulation period will change if mechanical tension is applied to the optical fiber. Also the effective index of grating refraction may change when deformation is present. Other than strain these variables are also sensitive to temperature variations.

Equation (4.10) describes the relationship between the wavelength reflected, the strain applied to optical fiber and temperature.

$$\frac{\Delta\lambda}{\lambda_0} = G_\varepsilon \times \Delta\varepsilon + G_T \times \Delta T \tag{4.10}$$

where:

- $\Delta\varepsilon$ = variation of axial deformation or strain;
- ΔT = temperature variation;
- G_ε = sensor's strain gain factor;
- G_T = sensor's temperature gain factor.

The strain gain factor for Bragg sensors depends on the optical fiber characteristics and place where it is installed. If only axial strain is taken into consideration, neglecting the cross-section deformation and assuming that the optical fiber is composed by a homogeneous and isotropic material, which is not sensitive to temperature variations, the strain gain factor is given by [50]:

$$G_\varepsilon = \frac{1}{\lambda_0}\left(\frac{\Delta\lambda}{\Delta\varepsilon}\right) = 1 - \frac{\eta_0^2}{2}\left(p_{12} - \upsilon(p_{11} + p_{12})\right) \tag{4.11}$$

where:

- υ = Poisson coefficient factor for optical fiber;
- p_{11} e p_{12} = values that characterize the sensitivity of the optical fiber to strain.

The most common values for $G_\varepsilon$vary usually between $0{,}7 \times 10^{-6}\mu\varepsilon^{-1}$ and $0{,}8 \times 10^{-6}\mu\varepsilon^{-1}$. The gauge factor for Bragg sensor can be calculated from the next mathematical expression.

$$S_\varepsilon = \frac{\Delta\lambda}{\Delta\varepsilon} = \lambda_0 \times G_\varepsilon \tag{4.12}$$

For Bragg sensors with a strain gain factor of $0.78 \times 10^{-6}\mu\varepsilon^{-1}$ the gauge factor is approximately 1pm/με for wavelengths of 1300nm. If the operating wavelength increases the sensor's sensitivity also increases. For example with the same strain gain factor Bragg sensors that operate in the 1550nm range exhibit a gauge factor of 1.21pm/με as for other that works in the 830nm region exhibit a 0.65pm/με.

A typical data acquisition system for Bragg sensors is shown in the next figure. The details of each subsystem are not discussed but the operation mode is easy to understand.

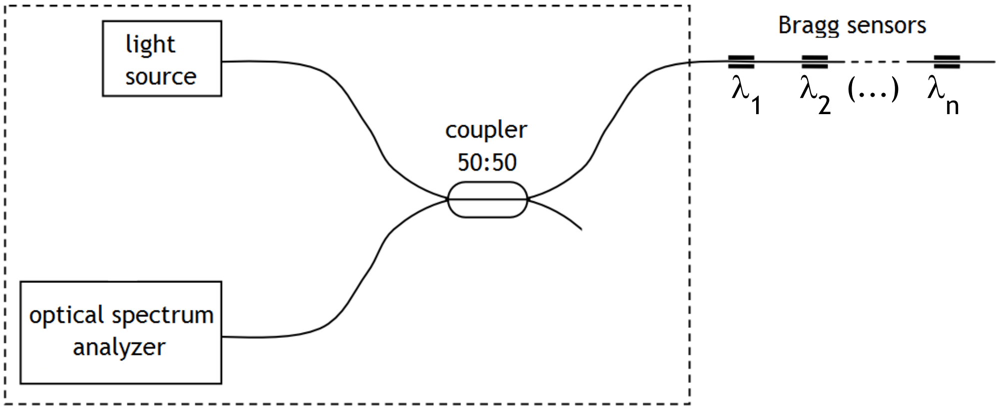


Figure 4.8 – Diagram of a typical instrumentation system based on Bragg sensors (adapted from [50])

On the system shown the light source is responsible for creating the light beam. The light signal goes to the Bragg sensors which reflect some part. The wavelength of the reflected signal is then analyzed by the optical spectrum analyzer.

Bragg sensors like other optical sensors present advantages in the field of signal transmission. In other words, the output of these transducers is not affected by electromagnetic noise unlikely electrical sensors. Another benefit of using Bragg sensors is that they can be made of the same type of optical fiber used for data transmission, which reduces its costs comparative to other optical fiber transducers. Also they exhibit a linear response to strain and temperature for a wide range of inputs. Last but not least, a number of sensors can be mounted in a single optical fiber due to the possibility of spatial multiplexing.

Fabry-Perot Sensors

Fabry-Perot sensors are based on light phase modulation and use optical interferometry. This technique combines two separate light waves together, superimposed them in order to extract information about their properties. This works because when two waves with the same frequency combine, the resulting pattern is determined by the phase difference between the two waves. If they are in phase, constructive interference will happen while if they are out of phase they will undergo destructive interference.

There are several interferometers but for this type of sensors the Fabry-Perot interferometer has the most interest. It is the only one that works with one fiber which makes its installation easy and immune to temperature variations [54]. Nowadays these sensors are extrinsic and have the configuration presented in Figure 4.9.

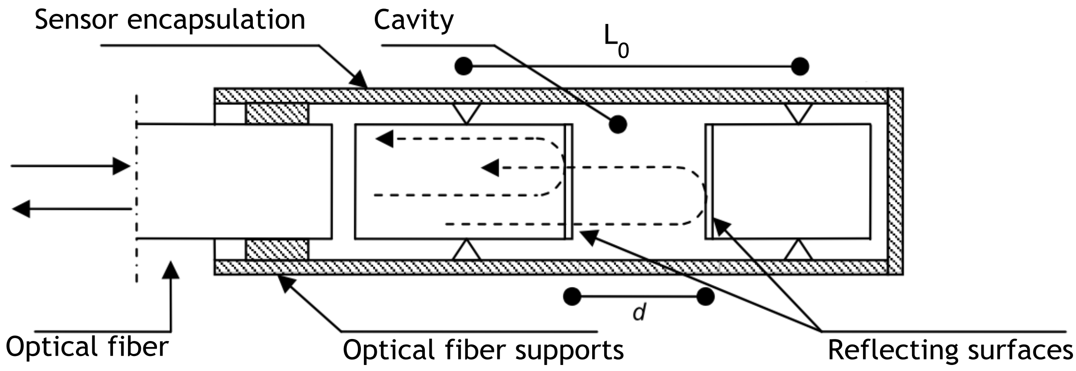


Figure 4.9 – Diagram of the Fabry-Perot interferometer [50]

For Fabry-Perot interferometers the difference of phases is function of the cavity length *d*. Strain is then obtained by dividing the variation of *d* with the reference length L_0. The optical fiber that transmits the optical signal to the data acquisition system is fixed to the sensor encapsulation but is physically separated, which makes it more robust and increases its durability.

The use of these sensors in structural monitoring has increased mainly due to the following advantages:

- Fabry-Perot strain sensors present low sensitivity to temperature changes;
- The fact they are extrinsic reduce the influence of glues used to install these instruments;
- The actual physical quantity measured is the length of *d* cavity which is an absolute parameter. The measure does not depend on the signal intensity of the output, which reduces the problems due to signal attenuation;
- The measure of L_0, from which the strain is calculated, is completely defined;

Fabry-Perot strain sensors, in relation to Bragg gratings, are advantageous when the material to be measured is subject to a large range of temperatures. Also, because they are extrinsic their application is easier and there are no concerns with glue properties. However spatial multiplexing that Bragg sensors feature is an enormous advantage when speaking of multiple sensors systems. Unlike Bragg sensors, Fabry-Perot transducers need a lot of extra equipment in order to achieve systems with such a large number of sensors. Thus, the cost per sensor is much lower for Bragg sensors.

The immunity to electromagnetic noise associated to the linear response with high precision that optical fiber sensors feature, makes them the future of strain measurements in structural health monitoring.

4.1.2 Selected Strain Gauge

For the proposed system presented in this document the strain gauge selected was the CEA-06-250UW 350 from VISHAY. This is a resistive foil strain gauge for general use. The strain sensitive material is Constantan that is encapsulated in polyimide with large rugged copper-coated tabs (Figure 4.10). It can be used for static and dynamical strain analysis and has a temperature range of -75^{0}C to + 175^{0}C.

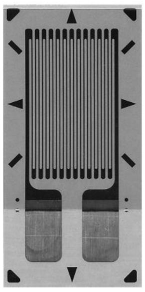


Figure 4.10 – Strain gauge CEA-06-250UW 350 from VISHAY

These gauges are self-temperature-compensated; the 06 value represents the thermal coefficient of steel which is the material where this device is to be applied. Other important characteristics features in this sensor are:

- Flat grid with linear pattern, the advantages of this feature has already been discussed;
- Resistance of 350 ohm with variations of ±0.3%. This is preferable to 120 because it generates less heat for the excitation voltage;
- Can measure strain variations of 5%, which is enough to test deformations on steel;
- Inexpensive considering optical fiber sensors;
- Good response in the low frequencies range. Frequency responses of less than 1Hz are typical for tall or long civil infrastructures.

Another measure of interest is acceleration. Next section describes the sensors used to acquire this physical magnitude, as some of the desirable characteristics they feature.

4.1.3 Acceleration Measurements

Next to strain gauges, accelerometers are the most commonly used sensors for SHM systems. They are mainly adopted on dynamic load tests in order to assess about the structure global frequency response. This characteristic is used to determine the dynamic behaviour of the construction which gives important information about the structure load-bearing capacity, remaining lifetime and allows the definition of proper maintenance intervals [1].

There are a wide variety of accelerometers for application on civil structures; however piezoelectric accelerometers are the most commonly employed. The study of accelerometers based on MEMS technologies is also important due to the increasing visibility of this type of sensors for other applications. As it was stated in Chapter 2, they exhibit interesting features in terms of cost and size that makes them tempting for integration in low cost SHM systems. Therefore, an analysis of these two types of devices is presented in this section.

Piezoelectric Accelerometers

These sensors rely on the piezoelectric effect of quartz or ceramic crystals to generate an electric output signal that is proportional to the applied force. The piezoelectric effect produces an accumulation of charged particles on the crystal in response to the applied mechanical stress (Figure 4.11).The accumulated charges are afterwards conditioned by transistors in order to produce a voltage that can be measured by the data acquisition system.

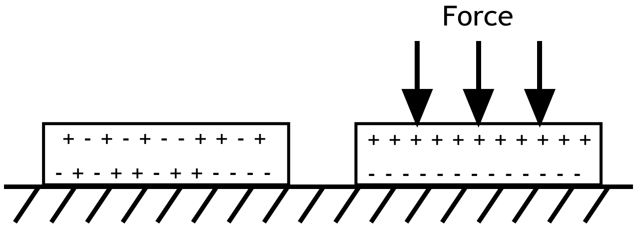


Figure 4.11 – Operation principle of piezoelectric accelerometers

The mechanical stress imposed to the crystal results from a seismic mass that is connected to it. When the accelerometer is subjected to movements this mass applies a force to the crystal that obeys to the second Newton's law: $= m \times a$. Therefore the electric response of these sensors, resulting from the applied force on the crystal, is proportional to the acceleration in a specific frequency range.

The mechanical configuration can have different topologies according to the measures of interest. These variations are related to the way that the seismic mass acts upon the piezoelectric material. The two most common topologies are Shear and Flexural Beam [49].

Shear topology is based on bonding piezoelectric crystals between a center post and seismic mass (Figure 4.12). Under acceleration the mass causes a shear stress to be applied on the sensitive material. This topology allows the isolation of the crystals from the sensor base which improves its thermal response and avoids errors due to base bending effects. Shear geometry is also characterized by its small size, minimizing the mass loading effects on the test structure.

Flexural mode configurations use a beam-shaped crystal that is supported by a carrier (Figure 4.12). These sensors offer low profile, light weight, excellent thermal stability and an economical price. Typically they are well suited for low-frequency and low-g-level applications.

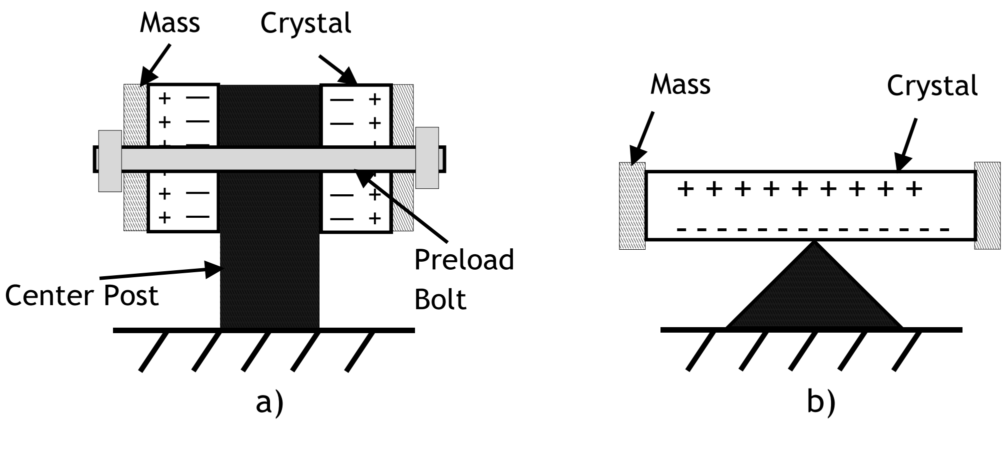


Figure 4.12 – Two design configuration for piezoelectric accelerometers: a) Shear design; b) Flexural Beam design

For validation purposes the piezoelectric accelerometer used was the PCB 352B10 (Figure 4.13).

Figure 4.13 – Piezoelectric accelerometer PCB 352B10

This sensor has already been used on acceleration measurements in infrastructures performed by LABEST researchers. Therefore the existing knowledge about this sensor performance makes it suitable to use as a reference. The next table presents its most important characteristics.

General characteristic	Performance exhibit
Sensitivity (± 10%)	10 mV/g
Measurement Range	± 500 g pk
Frequency Range (± 5%)	2 to 10000 Hz
Non-Linearity	≤1 %
Temperature Range (Operating)	-54 to +121 °C
Excitation Voltage	18 to 30 VDC
Constant Current Excitation	2 to 20 mA
Sensing Element	Ceramic
Sensing Geometry	Shear
Size	8.1 mm x 6.1 mm
Mounting	Adhesive

Table 4.1 – Characteristics of piezoelectric accelerometer PCB 350B10

Accelerometers based on MEMS

Micro-Electro-Mechanical Systems, or MEMS, is a technology that in the most general perspective can be defined as miniaturized mechanical and electro-mechanical elements that are constructed using the techniques of microfabrication. Their size usually varies from below one micron to several millimetres. The main advantage of this technology is that many elements can be built on the same wafer surface. The integration of different components in a single microchip is expected to be one of the most important technological breakthroughs of the future.

There are several types of MEMS devices, however in this document only microsensors are described. The possibility of integrating several components on a single chip enables the creation of small sensors that have all the required electronics attached. These are smaller, more functional and feature reduced manufacturing costs.

Microaccelerometers are one of the most used types of MEMS sensors. Typically they are composed by a mechanical microstructure that has a determined dynamic response to load variations. Accelerometers in specific are based on the cantilever principle in which an end mass displaces under an inertial force (Figure 4.14).

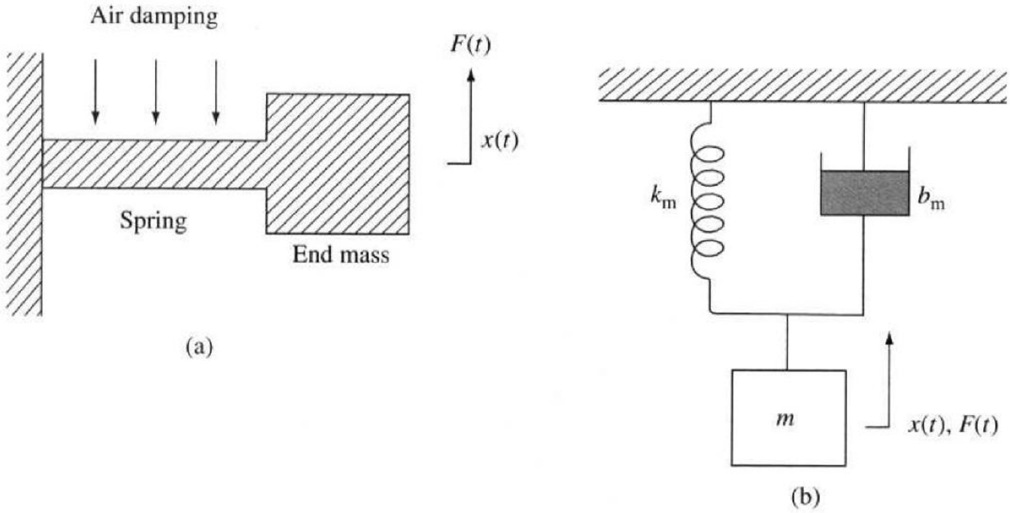


Figure 4.14 – (a) A simple microflexure comprising a heavy end or load mass and a spring like support; (b) the equivalent system of the microflexure [49]

Transducers used to convert these mechanical phenomena into electrical quantity are usually based on capacitive or resistive variations of a determined material. The following image shows a microflexure in which its end is coupled to a stationary sense electrode. These two plates separated by a dielectric operate has a capacitor.

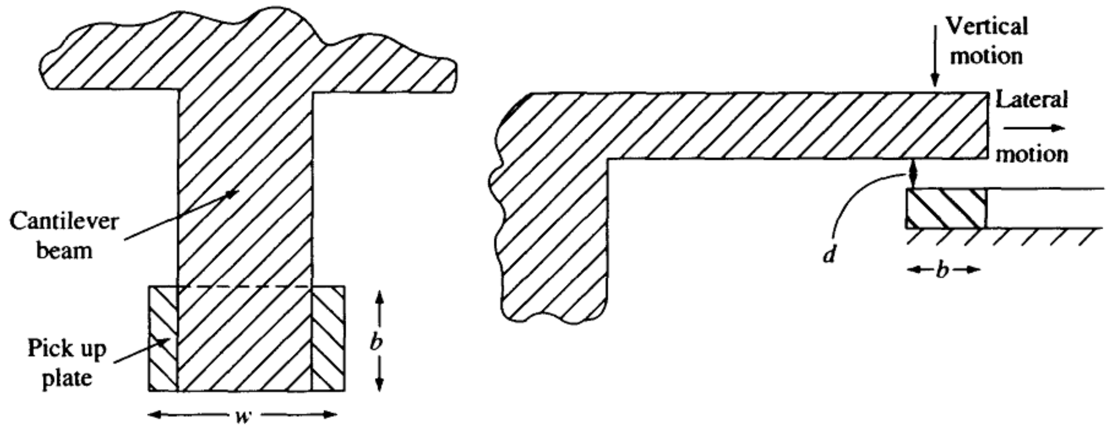


Figure 4.15 – Capacitive measurement of the deflection of a simple cantilever beam [49]

In this system a change in capacitance is related to changes in the plate separation d, area of overlap A, and dielectric permittivity ε. This relation can be expressed by the following equation [49]:

$$C = \frac{\varepsilon A}{d} \text{ thus } \frac{\delta C}{C} = \frac{\delta \varepsilon}{\varepsilon} + \frac{\delta A}{A} - \frac{\delta d}{d} \quad (4.13)$$

Many silicon mechanical microsensors use this principle to measure vertical deflection keeping A and ε constant. The area of overlap can be made relatively large and the gap size small, which allows integrated electronics to measure variations on the capacitance with acceptable sensitivity. Another advantage of using capacitive transducers derives from its high input impedance. Which means that little current is consumed making these devices suitable for integration in battery-based platforms as it is the case for this project.

The other important type of transducer used in accelerometers is based on a piezoresistor element (Figure 4.16).

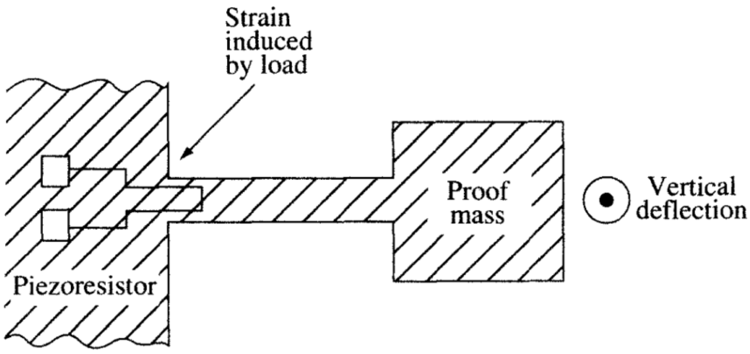


Figure 4.16 – Piezoresistive measurement of the deflection of a cantilever beam [49]

These elements can be made easily as a region of doped single-crystal silicon in a bulk-micromachined structure or as a doped polysilicon in a surface-micromachined structure. The gauge factor K_{gf} of a strain gauge defines its sensitivity and simply relates the change in the electrical resistance to the mechanical strain ε_m.

$$\frac{\Delta R}{R} = K_{gf} \times \varepsilon_m \qquad (4.14)$$

Piezoresistors can be made at very low cost and, like it was stated for semiconductor strain gauges, they have a strain gauge factor much higher than metals. However the same problems exist, their exact resistance is hard to control and the gauge factor is strongly dependent on both the doping level and the ambient temperature.

In order to test this type of sensors the ADXL203 from Analog Devices was employed. Respecting the MEMS philosophy this monolithic IC is a complete acceleration measurement system. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry. The output signals are analogue voltages proportional to acceleration. This device is capable of measuring both positive and negative accelerations in the range of at least ±1.7 g.

The sensor is a surface-micromachined polysilicon structure built on top of the silicon wafer. The mechanical structure applied is based on the cantilever beam previously explained. The deflection of the structure is measured by using a capacitor pickup that consists of independent fixed plates attached to the moving mass. The fixed plates are driven by square waves on phase opposition; acceleration deflects the beam and unbalances the differential capacitor resulting in an output square wave whose amplitude is proportional to acceleration. In order to determine the acceleration direction a circuit that employs phase demodulation techniques is applied. The output is then amplified and brought off-chip through a 32 kΩ resistor. Its bandwidth can be controlled afterwards by adding capacitors in order to filter unwanted frequencies (Figure 4.17).

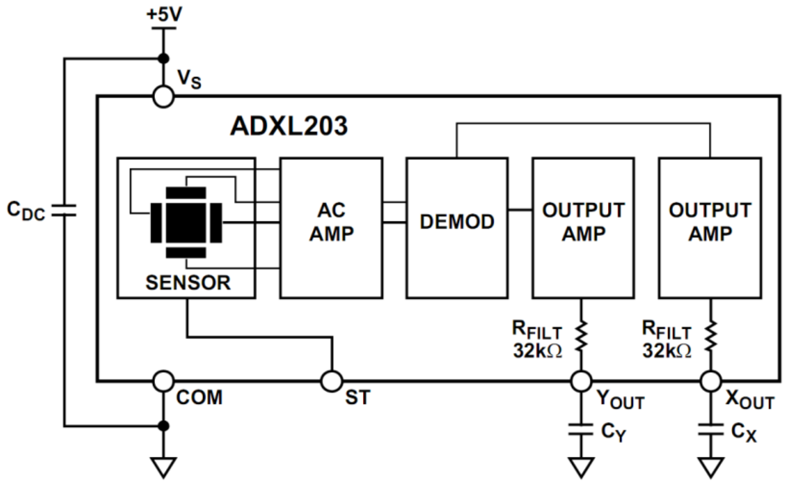


Figure 4.17 – Modules integrated on the ADXL203 chip

The ADXL203 does not incorporate temperature compensation circuitry; instead the fabrication process uses innovative design techniques that ensure high performance. There is essentially no quantization error or non-monotonic behaviour and temperature drift is very low (less than 10 mg over the -40°C to +125°C range).

This sensor is capable to perform acceleration measurements in two perpendicular axes, exhibits high precision and requires low power. The main characteristics of this device are presented in the next table.

General characteristic	Performance exhibit
Sensitivity (± 10%)	1000 mV/g (±0.3%)
Measurement Range	± 1.7 g
Frequency Range (± 5%)	0.5 to 2500 Hz
Non-Linearity	±0.2 %
Temperature Range (Operating)	-55 to +125 °C
Excitation Voltage	5 VDC
Constant Current Excitation	0.7 mA
Sensing Element	Capacitive Pickup
Sensing Geometry	Cantilever Beam
Size	5 mm x 5 mm x 2mm
Mounting	LCC package

Table 4.2 – Specification of the Analog Devices ADLX203 accelerometer

4.1.4 Temperature Measurements

As it has been stated before the environmental conditions can influence the output of many sensors used on structural health monitoring. Also, concrete behaviour may change when subjected to high temperatures. For both reasons the system developed in this dissertation should be able to integrate temperature sensors.

Traditional monitoring systems use temperature measures to compensate thermal effects in the read values, thus some sensors have already been validated for this use. The most commonly employed sensors are Resistive Thermal Devices (RTDs) and thermocouples.

Thermocouples

The basic principle of operation of this type of sensors relies on a phenomenon called the Seebeck effect. When two wires composed by dissimilar metals are joined together at both ends and one end is heated an electrical continuous current start to flow in the circuit. If the circuit net is open an electrical voltage appears, called the Seebeck voltage (Figure 4.18). This voltage is related to the junction temperature and the composition of the two metals.

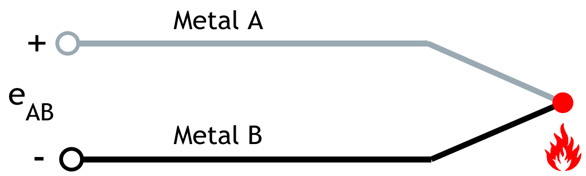


Figure 4.18 – The Seebeck effect, e_{AB} is the Seebeck voltage

One of the interesting properties of the Seebeck voltage is its linearity with the temperature for small ranges. When trying to measure the thermocouple voltage an interesting phenomenon occurs. The cables used to connect the measurement equipment create two new junctions. If the materials used for the thermocouple are constantan and copper, the connection made with copper creates a copper-to-copper junction that does not originate any thermal EMF, and a copper-to-constantan junction which adds an EMF (V_2) in opposition to V_1. The voltage reading is thereby proportional to the temperature difference between J_1 and J_2 (Figure 4.19).

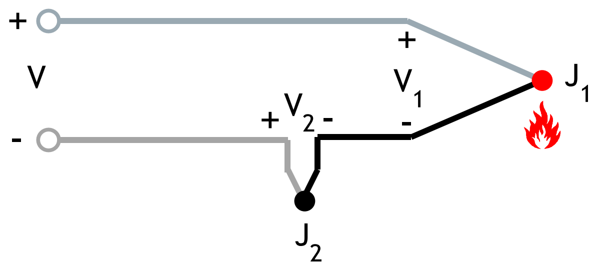


Figure 4.19 – The connection of measurement equipment creates a second junction (J_2)

There are many ways to impose or monitor the temperature in the junction J_2. The most common is to embed this junction in an isothermal block. This way the temperature at J_2 is always known. The output voltage is then given by the next equation:

$$V = (V_1 - V_2) = \alpha(T_1 - T_2) \tag{4.15}$$

where:

- $\alpha =$ Seebeck coefficient;
- T1 = temperature to be measured;
- T2 = reference temperature

If none of the materials used to build the thermocouple is copper, the two new junctions created when connecting the cables of the measuring system produce EMF. When this is the case both new junctions have to be at the same temperature in order to do not interfere with the reading.

There are several types of thermocouples. They are designated by letters that represent the combination of metals employed. The most widely use combinations are:

- K – Nickel-Chromium and Constantan;
- J – Iron and Constantan;
- K – Nickel-Chromium and Nickel-Aluminum;
- R, S – Platinum-Rhodium and Platinum;
- T – Copper and Constantan.

Depending on the combination used the Seebeck coefficient changes, giving more or less sensitivity to the sensor. The advantages of these sensors are: the reduce cost, their durability, robustness and the possibility of measuring high temperatures (1600 ^{o}C) [56]. Also they can be made really small which is good for some specific applications.

Resistive Thermal Devices (RTDs)

These sensors explore the fact that electrical resistance of some materials varies with temperature changes. Although they do not present the temperature range that thermocouples do, RTDs are more accurate and stable [56]. They are almost invariably made of platinum, however for some specific cases copper and nickel can be used. There are many different categories of RTDs. The most commonly used are described on the next paragraphs.

Film thermometers

This configuration has a layer of platinum on a substrate. The layer is usually extremely thin and can have different architectures. These are relatively low cost sensors due to the small amount of platinum used and have a fast response to temperature changes. One big disadvantage is the sensitivity to expansions of the substrate and platinum layer. If the sensor is subjected to variations of mechanical tension it may exhibits stability problems [56].

Wire-wound thermometers

Wire-wound thermometers can perform with better accuracy, especially if wide temperature ranges are to be measured. The helical coil of platinum provides better stability to strain variations. The main disadvantage of this sensor is the poor contact between the platinum and the point to be measured. This results in a slow thermal response.

Nowadays, this topology has been largely replaced by RTDs employing coil elements. This design has a coil that can expand freely over temperature changes which reduces strain-induced resistance variations. However, the use of this mechanical architecture may decrease the robustness of the sensor [56].

Compared to thermocouples these RTDs are more accurate, stable and present low drift. However for high temperatures it becomes increasingly difficult to prevent the platinum from becoming contaminated by impurities. These impurities change the response of the material to temperature variations and are almost impossible to control. Other disadvantages are the size of the sensor and the slow response to quick temperature changes.

In industry the most common RTD used is the PT100 (Figure 4.20). Its name derives from having a resistance of 100 ohms at 0 ^{o}C.

Figure 4.20 – PT100 sensor, the most common used on industrial environment

Integrated Circuit Temperature Sensor

The semiconductor temperature sensor is fabricated in a similar way as other common integrated circuits (ICs). Typically thousands of devices are formed on thin silicon wafers and then cut into individual chips. There are no generic types of IC temperature sensors, however a number of difference devices exist on the market [56].

Most of these sensors are based on the temperature and current characteristics of the transistor. If two identical transistors are operated at different but constant collector currents, then the difference in their base-emitter voltages is proportional to the absolute temperature of the transistors. Typically, the circuitry needed to convert the voltages measured to the actual temperature is also integrated on the IC. Therefore, these sensors commonly output the measured temperature in a digital form.

Integrated circuit temperature sensors share a number of advantages. They exhibit a good linear output with temperature, are relatively small and low cost. Although they present good accuracy if calibrated their repeatability is poor. This is mostly due to poor thermal contact with the outside surface. Another disadvantage is the temperature range; they are usually limited to -40 °C to 120 °C [56].

Like it was stated before, some of these sensors have integrated signal condition circuits. Therefore, they do not need external analogue-to-digital converters. Also many of these ICs include communication protocols with bus-type data acquisition systems.

The wireless sensor platform used on the system discussed in this dissertation employs an integrated circuit temperature sensor. Therefore a chance was presented to compare the results given by this type of devices to the traditionally used PT100.

The existing sensor is embedded on the RTC DS3231 from MAXIM. It represents the measured temperature as 10-bit code with a resolution of 0.25°C. It exhibits a range between -40°C and +85°C which is enough to acquire the environmental temperature. Has it was already stated the advantage of this sensor is the contact surface, being embedded in the RTC chip it will give the temperature of this IC rather than the ambient temperature.

4.2 Wireless Sensor Platform

On this section the Wireless Sensor Platform chosen is described and its main advantages discussed. The most important requirements concerning these devices are the ability to acquire data from several sensors and transmit the information gathered wirelessly to the system top-level units (Figure 3.1). These platforms should also exhibit other desirable characteristics that are discussed throughout this section.

In order to accomplish these features the Waspmote unit from Libelium was employed (Figure 4.21). This platform as already been tested for monitoring forests, greenhouses and cities with promising results [40]. The project here presented attempts to validate the use of Waspmote as part of a structural health monitoring system.

Figure 4.21 – The Libelium Wireless Sensor Platform Waspmote

As stated in chapter 2 these units usually integrate five different modules: sensing interface, computing core, wireless radio, actuating interface and portable power supply. Waspmote also exhibits these different subsystems. This device is, thus, presented following this architecture. However some of the subsystems can be aggregated in the same section for a better document organization.

First the general architecture of the Waspmote is presented and then the characteristics of the integrating modules are described in more detail.

4.2.1 General Architecture

Waspmote is based on a modular architecture. This way is possible to integrate different modules according to the application needs. Nevertheless this platform incorporates some core ICs that can't be changed. The main core elements are composed by: a microcontroller, a Real-Time Clock and a micro SD socket. Also sockets to connect the different modules are incorporated (Figure 4.23).

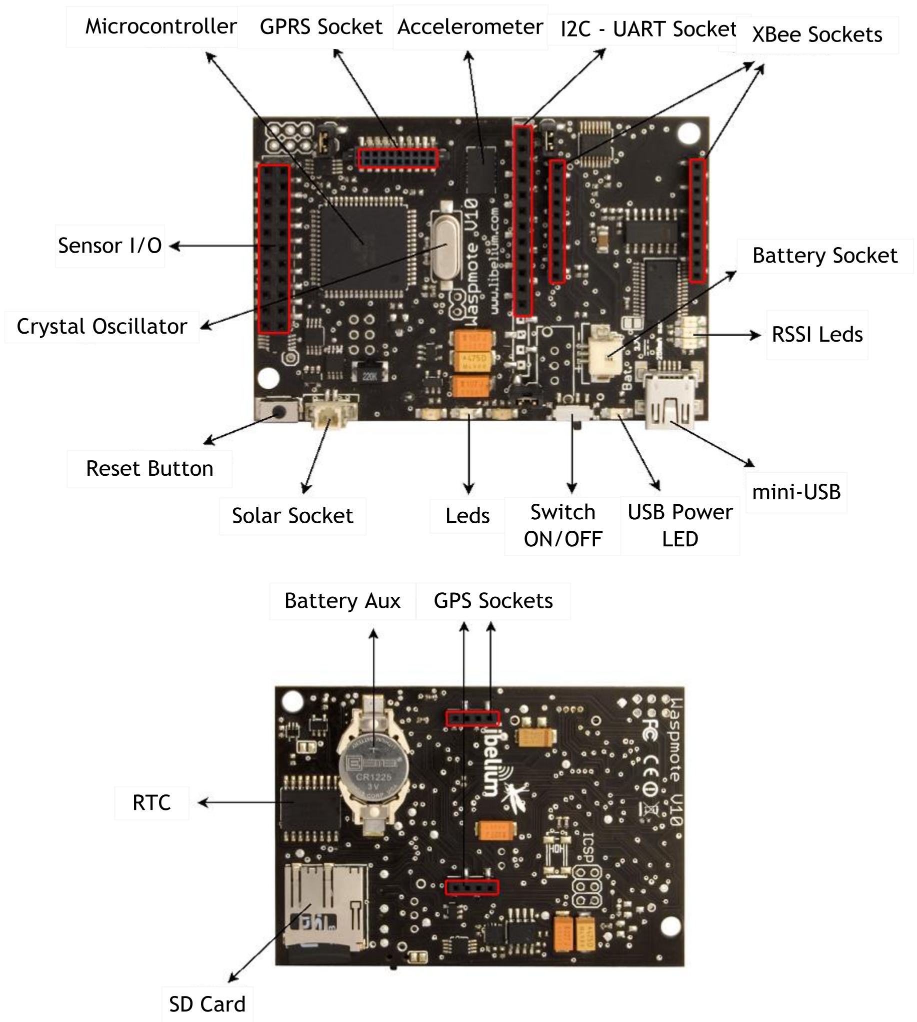


Figure 4.22 – Main Waspmote components and connectors

Waspmote platform weight 20gr and its dimensions are 73.5 x 51 x 13mm. These characteristics enable its easy installation in small hidden places. Therefore, they are suitable to apply in public structures without calling too much attention preventing the possibility of theft or vandalism. It operates in the temperature range of -20°C to +65°C.

The next block diagram shows the various subsystems available and how they are integrated. One interesting feature described by this diagram is the possibility of turning off some of the modules power supply increasing the overall energy efficiency. The “Modules power control” is commanded by the microcontroller, which means that is possible to manage the power supply connections via software.

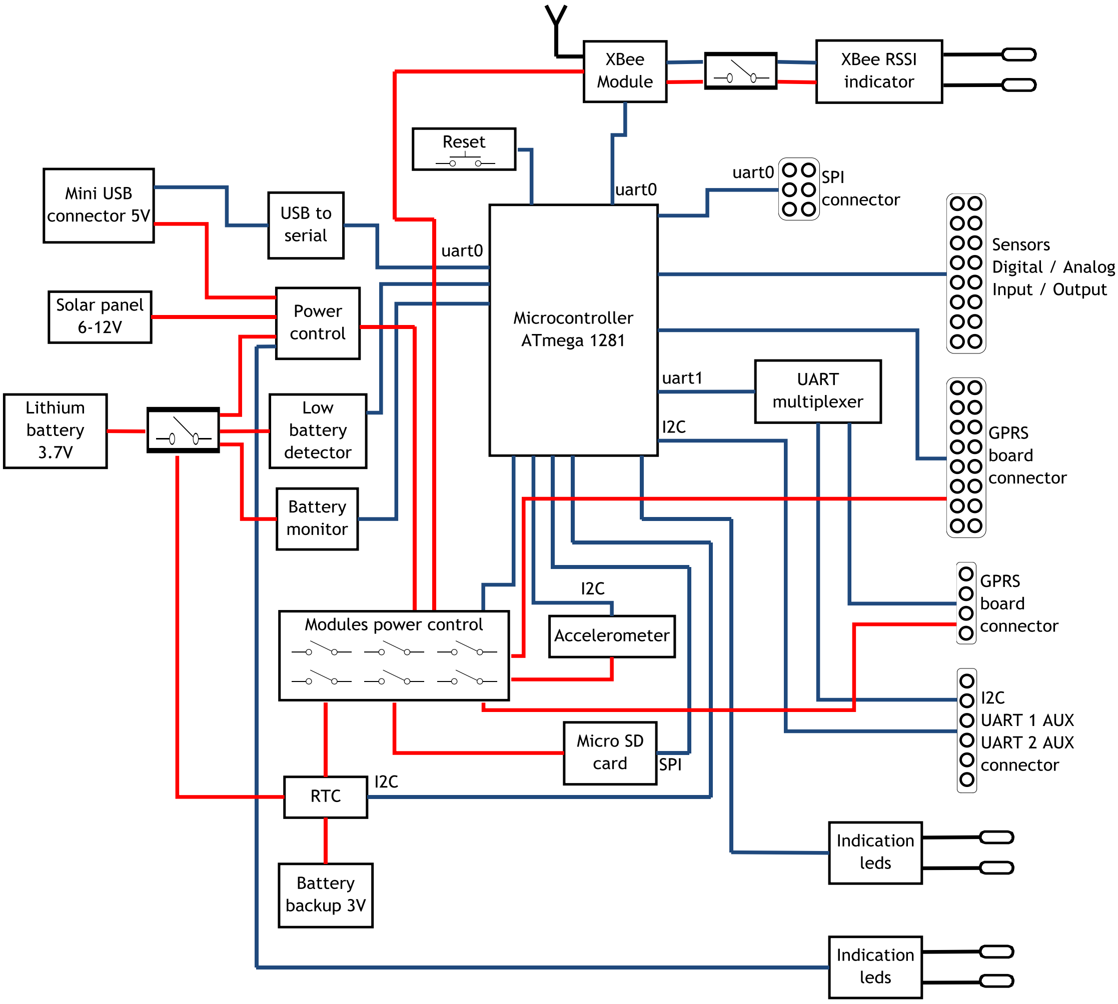


Figure 4.23 – Block diagram of Waspmote with red lines representing power signals and blue lines data signals

Different kinds of portable power sources can be integrated in this unit. The basic packet comes with a 3.7V battery with a capacity of 1250mAh but capacities of 1900mAh are also available. Waspmote has a control and safety circuit which makes sure the battery charge current is always adequate. In order to initiate the battery charging process the platform has to be connected through USB to a computer or any other adapter that powers the USB.

Renewable resources are other alternative. Waspmote can integrate solar panels up to 12V with maximum charging currents of 240mA. The current is once again controlled by the internal charging circuitry.

4.2.2 Computing Core and Sensing Interface

The computing core employed is the ATmega 1281 from Atmel Corporation. This is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. These architectures are able to execute powerful instructions in a single clock cycle which increases the speed of the processor enabling a better optimization of the energy consumption. The ATmega1281 is able to work with clock frequencies of 16 MHz; however, Libelium developers chose to employ an 8 MHz crystal oscillator. This is clearly an effort to keep the power consumption as low as possible. Also Atmel advises the use of supply voltages of at least 4.5V for speed frequencies higher than 8 MHz which in the Waspmote case would force the use of different kind of batteries (Figure 4.24).

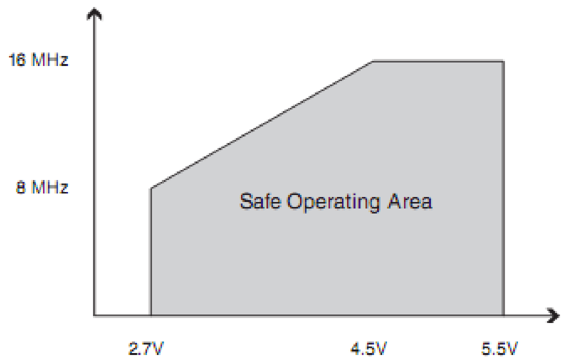


Figure 4.24 – Maximum frequency vs Vcc for ATmega1281

The ATmega1281 is capable of achieving throughputs approaching 1 MIPS per MHZ. It integrates 8KB of SRAM, 4KB of EEPROM and 128KB of flash memory. For a monitoring system the microcontroller storage capacity alone is not enough. Therefore Waspmote has an embedded micro SD card slot that can support cards up to 2GB. This module communicates with the ATmega1281 using the SPI bus. To give an idea of the capacity of information that can be stored, if one measure occupies 100 Bytes 20 million measurements can be kept in the SD card.

In order to interface with the exterior the ATmega1281 incorporates analogue-to-digital converters as well as some standard protocols to communicate with sensors. Actually, the sensing interface in the Waspmote platform is made by the peripherals that ATmega1281 feature.

Analogue-to-Digital Converter

The analogue-to-digital converter featured has a resolution of 10-bit and is from the SAR type. It is connected to a 16 channel analogue multiplexer which allows 16 single-ended voltage inputs. It also supports 14 differential input channels from which 4 are equipped with a programmable gain stage providing amplification steps of 0dB, 20 dB or 46 dB before the ADC conversion. Although the ADC on ATmega 1281 provides all these input channels Waspmote only has 7 single-ended inputs available which limits the number of sensors that can be connected. The reference voltage value for the inputs is 0V (GND) and the maximum value is 3.3V which corresponds to the microcontroller's general supply voltage.

The successive approximation circuitry that integrates the ADC requires an input clock frequency between 50 kHz and 200 kHz. Thus, the microcontroller has a prescaler module that can generate an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The number of ADC clocks needed to convert an analogue signal to digital is dependent on various factors. Analogue circuitry, for example, takes some time to be initialized; therefore the first conversation of the ADC takes more cycles than the ones after. Also the sample and hold circuit takes time to actuate, which decreases even more the sampling frequency, all these properties are accounted in the next timing diagram (Figure 4.25).

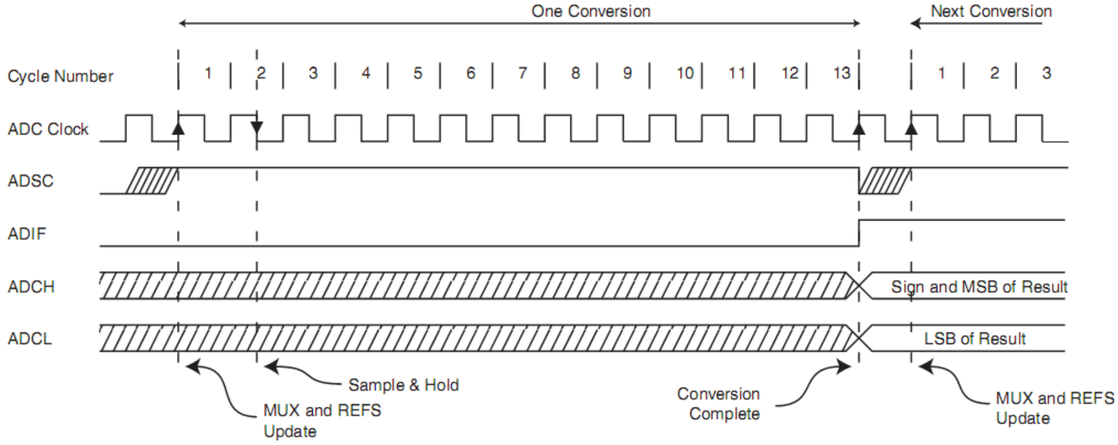


Figure 4.25 – ADC timing diagram of a single conversion that is not the first one [57]

The diagram shown is for single conversion mode but this ADC has also free-running mode and auto-trigger mode. Free-running mode automatically starts another conversion after the last one is finished. The auto-trigger mode initializes the conversion when some event is detected. Event handlers are managed by the microcontroller. Next table exhibit the number of cycles needed to performance these different processes, which allow the calculation of the maximum sampling frequency of the ADC. However when using different channels the time that the multiplexer needs to change the input has to be accounted as well.

Condition	Sample & Hold (cycles from start of conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5	13/14

Table 4.3 – Number of cycles of the different operations of the ADC

Using the values in Table 4.3 and an ADC clock of 200 kHz, the maximum sampling frequency for one channel considering that is not the first conversion is calculated by the next mathematical expression:

$$\frac{200\ \times 10^3}{(1.5+13)} \cong 13.793\ kHz \tag{4.16}$$

The calculated sampling frequency is close to the one that the manufacture provides on the component datasheet (Table 4.4).

Other important characteristics of the analogue-to-digital converter are its accuracy and non-linearity. The manufacturer states that this device features 1 LSB integral non-linearity and ±2 LSB absolute accuracy. Both these values should be taken into consideration when calculating the relative error of the measure.

All these characteristics are summed up on the following table.

Characteristics	Performance Exhibited by the ADC Used
Resolution	10-bit
Integral Non-linearity (INL)	1 LSB
Absolute Accuracy	±2 LSB
Conversion Time	13 – 260 µs
Sampling Frequency	76.9 kSPS (15 kSPS at maximum resolution)
Number of Single Ended Input Channels	16 (Multiplexed)
Number of Differential Input Channels	14 (Multiplexed)
ADC Input Voltage Range	0 - V_{cc}

Table 4.4 – Characteristics of the ADC integrated on the ATmega1281 [57]

Other Sensor Interface Protocols

Other than an analogue-to-digital converter the ATmega1281 offers the possibility of using I^2C and SPI to communicate with sensors. I^2C protocol was launched by Phillips and it is a multi-master bidirectional single ended computer bus that is used to attach low-speed peripherals to an embedded system (Figure 4.26). Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate either as a transmitter or receiver. They can also behave like masters or slaves. Units working as master are responsible to initiate the transfer, generate clock signals and terminate the transfer. Being a multi-master protocol I^2C provides an arbitration algorithm to ensure that only one master can control the bus at a time. Slave units are the ones addressed by masters they can also work as transmitter or receiver. In normal mode this protocol works with 8-bit packets that can be transmitted at data rates up to 100kbit/s.

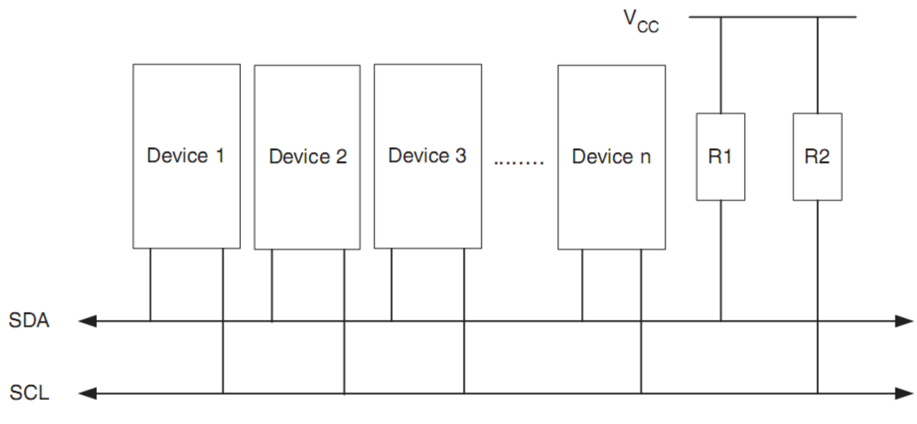


Figure 4.26 – I^2C bus architecture

The ATmega1281 support both master and slave operation and enables the interconnection with up to 128 different devices. Waspmote more than provide connections to the I^2C bus, use this protocol to communicate with the accelerometer, the Real Time Clock and the digital potentiometer used to configure the low battery alarm threshold. For these communications the microcontroller acts as master while the peripherals are slaves.

Waspmote also supports Serial Peripheral Interface (SPI) protocol which is a synchronous serial data link standard that operates in full duplex mode. Like I^2C devices communicate in a master-slave mode where the master is in charge of starting the communication. However in contrast with I^2C this standard only permits one master that can be connected to several slaves. Master address to a determined slave by using a chip selector (Figure 4.27). Compared to I^2C this protocol has a higher throughput (10Mbps), it is not limited to 8-bit words and has less circuitry associated if only one slave is involved (no need of pull-ups). Waspmote uses this protocol to communicate with the micro SD card.

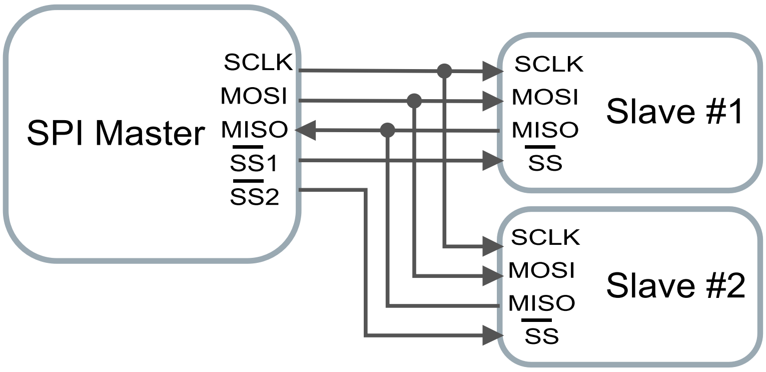


Figure 4.27 – SPI interconnection with two slaves

ATmega1281 also features the typical Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) protocol. It actually allows two different connections with this protocol. The Libelium platform uses the USART0 to receive and transmit data to the wireless radio. This connection is also shared with the USB port. UART1 is connected to a four channel multiplexer that is controlled via software. GPRS and GPS boards use one channel each other the other two can be accessed by the user.

Other Peripherals Related to the Computer Core

Another core IC is the Real Time Clock. Waspmote has a built in RTC which keeps it informed of the time. Alarms can be program to wake up the mote or just to specify that a determined task should be performed at that time. The RTC used is the Maxim DS3231SN which operates at a frequency of 32.768 Hz. This IC has an internal compensation mechanism used to compensate variations of the quartz crystal frequency due to temperature changes. Therefore, this model presents an inaccuracy of only ±2ppm (1min/year). In addition to the main battery this IC is also supplied by an auxiliary coin battery. Even if the main battery is disconnected and the microcontroller enters in sleep mode the RTC still keeps the track of time.

It is important to refer that Libelium provides an IDE for software development that is packed with software libraries developed in C++. Most of these features can, therefore, be programmed easily.

4.2.3 Wireless Radio

One of the advantages of the modular architecture presented by Waspmote is the possibility of choice between different wireless radios. This platform can integrate several DIGI XBee radios for communication in the Industrial Scientific Medical Band (ISMB). As it has been stated in chapter 2, ISM bands, although with some restrictions, can be used free of charge for research purposes. In order to communicate with these devices Waspmote's microcontroller uses the USART0 channel at a speed of 38400bps. Also a Bluetooth radio is available. It uses the same ports as XBee modules and it is based on eUnistone 31308/2 chip.

Together with the Bluetooth module there are eight different wireless radios distributed by Libelium for integration on Waspmote, next table shows their main characteristics.

Model	Protocol	Frequency	txPower	Sensitivity	Range*
XBee-802.15.4	802.15.4	2.4GHz	1mW	-92dBm	500m
XBee-802.15.4-PRO	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB	ZigBee-Pro	2.4GHz	2mW	-96dBm	500m
XBee-ZB-Pro	ZigBee-Pro	2.4GHz	50mW	-102dBm	7000m
eUnistone 31308/2 (Bluetooth v2.0)	802.15.1	2.4GHz	1.8mW	-86dBm	250m
XBee-868	RF	868MHz	315mW	-112dBm	12Km
XBee-900	RF	900MHz	50mW	-100dBm	10Km
XBee-XSC	RF	900MHz	100mW	-106dBm	12Km

* *Line of Sight and 5dBi dipole antenna*

Table 4.5 – Main characteristics of different wireless radios available for Waspmote

An overview of the capabilities and functionalities of these modules is presented on the next paragraphs.

4.2.3.1 XBee-XSC RF module

Due to the use of the 900MHz frequency band to communicate this module is only available for North-America and Canada. Even so, a small overview of its main features is done in order to give a base of comparison with other modules.

XBee-PRO XSC OEM RF module is an embedded wireless solution designed for long range communications. It is capable of deploying point-to-point, peer-to-peer and point-to-multipoint networks. It has a transmission rate of 9600bps and use spread spectrum frequency hopping techniques. Like most of the XBee devices here presented this module represents an out-of-the-box solution and does not need any configuration to start RF data communications. However, for advanced users it has an AT command mode where it is possible to select flow control, hopping channel, address mask and other specific configurations.

Its main advantage is the capability of achieving 24Km communications range on outdoor line of sight environments using a high gain antenna. This feature makes him a good choice when RF penetration and absolute transmission distance are paramount to the application.

4.2.3.2 XBee-900 RF module

This unit also uses the 900MHz frequency band to transmit that, therefore in the same way as last module, it is only available for North-America and Canada.

The main difference between the XBee-900 PRO and the last its higher data transmission rate and the possibility to be used in self-healing mesh topologies. It exhibits a RF data rate of 156.25 kbps and uses twelve channels with a bandwidth of 2.16 MHz (Figure 4.28). This module also implements algorithms for node discovering being possible to implement multi-hopping mechanisms. Other important feature is the support for encryption provided through the AES 128-bit algorithm.

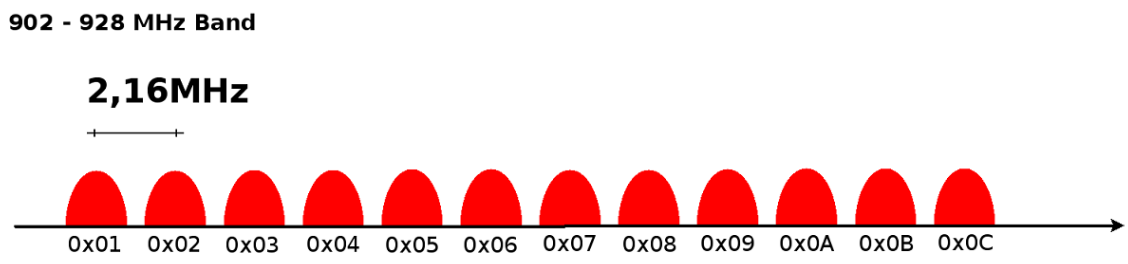


Figure 4.28 – Channel frequencies in the 900MHz band

Comparatively to the XBee-XSC it exhibits a lower transmission range, 3Km outdoor line of sight and 140m on urban environment.

4.2.3.3 Xbee-868 RF module

This module uses 868MHz frequency band and is therefore suitable from European applications. It is able to deploy point-to-multipoint networks and supports RF communications with ranges up to 40Km on a clear line-of-sight environment. As XBee-XSC this modules could be the right solution if RF penetration and transmission distance are critical to the application. The main disadvantage of this unit is the fact that it does only use one channel to transmit. In order to overcome the ambient interference that can exist in urban environments it uses a multiple transmissions and acknowledgements mechanism to make sure that the packet was correctly transmitted. The retry system may fail if a lot of retransmissions star to occur. This happens because the network would not be able to transmit such a large amount of data and transmission buffers may become full. The RF data rate of this module is 24kbps with a 10% duty-cycle. Like XBee-900 PRO it supports data encryption through the AES 128-bit algorithm.

4.2.3.4 Bluetooth module

Bluetooth technology is a short-range communications technology that is simple and secure. Its key features are robustness, low power and low cost. This technology has such high visibility that in 2005 the IEEE community published the 802.15.1 standard in order to normalize its communications. IEEE 802.15.1 specifies the physical and data link layers with the objective of increase the interoperability of devices using this technology [58].

Bluetooth uses the 2.4GHz frequency band and employ frequency hopping spread spectrum techniques in order to avoid interference. It defines 79 channels with a bandwidth of 1MHz. Also it changes between channels at a rate of 1600 per second in a pseudorandom fashion determined by the master device. This protocol employs a network topology called piconet where several devices are connected in a master-slave manner. In this network the master can communicate with up to 7 simultaneous slaves and the same device can have different roles in different piconets at the same time. The roles of master and slave can be exchanged at any time.

The first version of Bluetooth can transmit at a speed up to 1 Mbps; however, improvements have been made and version 2.0 can transmits at 2 or even 3 Mbps. Summing up this protocol is good for short range communications, it supports a lot of features like adaptive frequency hopping, which is an improvement of the traditional frequency hopping algorithm, high data rate transmission, high interoperability, low cost and low power.

Waspmote's Bluetooth module incorporates the eUnistone 31308/2 chip from Infineon. It integrates the core stack of IEEE802.15.1 and the Serial Port Profile (SPP), which emulates a serial port used for data transmission between the Waspmote and the module. Like the XBee modules it communicates with the microcontroller through the USART0 port at 38400bps. It features nodes discovering and a range of 250m in line of sight.

4.2.3.5 IEEE 802.15.4 and ZigBee modules

The IEEE 802.15.4 wireless technology is a short-range communication standard intended for applications with relaxed throughput and latency requirements. It main features are related to the requirements of wireless sensor networks, low power consumption, low complexity, low cost and low data rate transmissions. In fact IEEE 802.15.4 is considered by some authors the standard for WSNs [17].

As it was stated in chapter 2, IEEE 802.15.4 defines the two bottom layers of the OSI protocol model: Physical (PHY) and data-link (MAC). Also, it was given examples of protocols for the upper layers like ZigBee and WirelessHART. For the reason that there is no module available for the Waspmote integrating the WirelessHART stacks, in this section only ZigBee will be covered.

The physical layer described by the IEEE 802.15.4 standard operates in three different ISM bands according to the geographical area where the system is deployed. However, in this section only the use of 2.4GHz frequency band is discussed. This happens because 2.4GHz frequency band is available worldwide and it is the only one supported by XBee modules.

The PHY provides the interface with the physical medium. It is responsible for the radio transceiver activation and deactivation, energy detection, link quality, clear channel assessment, channel selection, and transmission/reception of the message packets. Furthermore, it is also in charge of the RF link, between two units, bit modulation and demodulation, synchronization between nodes and finally for packet level synchronization. At the 2.4GHz frequency band IEEE802.15.4 specifies 16 half-duplex channels (Figure 4.29). The Direct Sequence Spread Spectrum (DSSS) technique is then used in order to reduce the interference level.

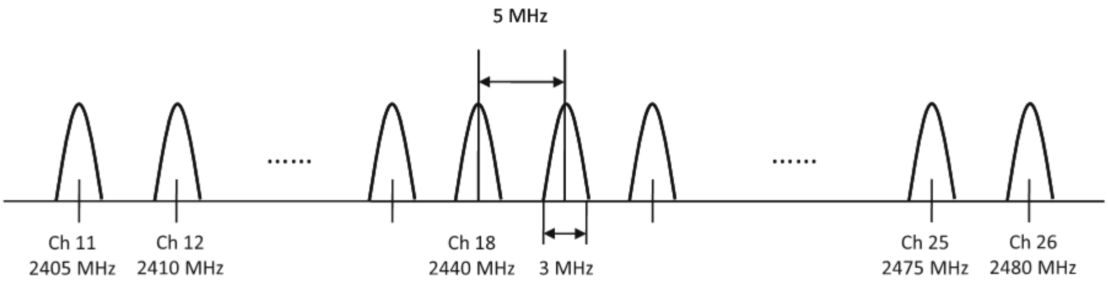


Figure 4.29 – Channelization at the 2.4 GHz band

This technique phase-modulates a sine wave pseudorandomly with a continuous string of pseudonoise code symbols called chips. These chips have a much shorter duration than an information bit therefore each information bit is modulated by a sequence of much faster chips. The pseudonoise sequence is known by the receiver, enabling it to reconstruct the original signal. Thereby, this technique spread the energy of the original signal into a much wider band. If a signal with different pseudonoise is transmitted over the same channel is rejected by the Code Division Multiple Access (CDMA) method. IEEE802.15.4 on the 2.4GHz ISM band implements a half-sine-shaped Offset Quadrature Phase Shift Keying (O-QPSK) modulation format with DSSS at 2 Mchip/s. Data rates of 250kbit/s are available and the ideal transmission range is 200m [17].

According to the IEEE 802.15.4 standard, transmission is organized in frames. The most important ones are designated by Physical Protocol Data Units (PPDUs) and can be a beacon frame, a data frame, an ACK frame and a MAC command frame. In order to detect faulty packets a Cyclic Redundancy Check (CRC) is used.

The MAC layer provides access control to a shared channel and reliable data delivery. Its main functions are: association and disassociation, security control, optional star network functions, such as beacon generation, generation of ACK frames and finally application support for the two possible networks topologies described by the IEEE802.15.4 standard.

This protocol defines two different modes of operation, beacon-enabled and non beacon-enabled. Each one of these modes uses different channel access mechanisms. Non-beacon-enabled mode uses an unslotted CSMA/CA protocol to access the channel. The algorithm is implemented using units of time called backoff periods. There are two main variables related to the process. The number of times that the algorithm was required to backoff while trying to transmit is the NB. The time that a node need to wait in order to attempt to access a channel is called the BE. If the node can not assess the medium in a period of time lower than the defined BE_{max} it will give up a return that the transmission failed [59].

In the beacon-enabled mode the medium access is managed by a superframe (Figure 4.30) that starts with a packet called beacon. This superframe is sent by the Wireless Personal Area Network (WPAN) coordinator periodically to synchronize and identify nodes associated with it. It is composed by the Contention Access Period (CAP) and the Contention Free Period (CFP) composed by GTS slots that can be assigned for specific unit gain access to the medium.

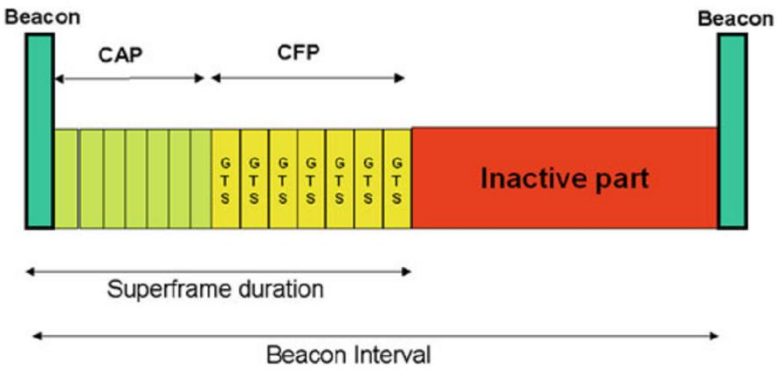


Figure 4.30 – The IEEE 802.15.4 superframe defined in the beacon-enabled mode [59]

A network device wishing to send data to the WPAN coordinator needs to listen for a beacon and if it does not have a GTS assigned transmit in the CAP period using CSMA/CA mechanisms to get access to the channel. Between two beacons there is an inactive interval of time where the units should go to sleep in order to save energy.

On the CSMA/CA algorithm used on the beacon-enabled mode there is one more variable to account. CW indicates the number of backoff periods that need to be clear in order to a unit get medium access (Figure 4.31). Using this technique the backoff periods are always synchronized with the beacon and all transmissions start on the boundary of a beacon.

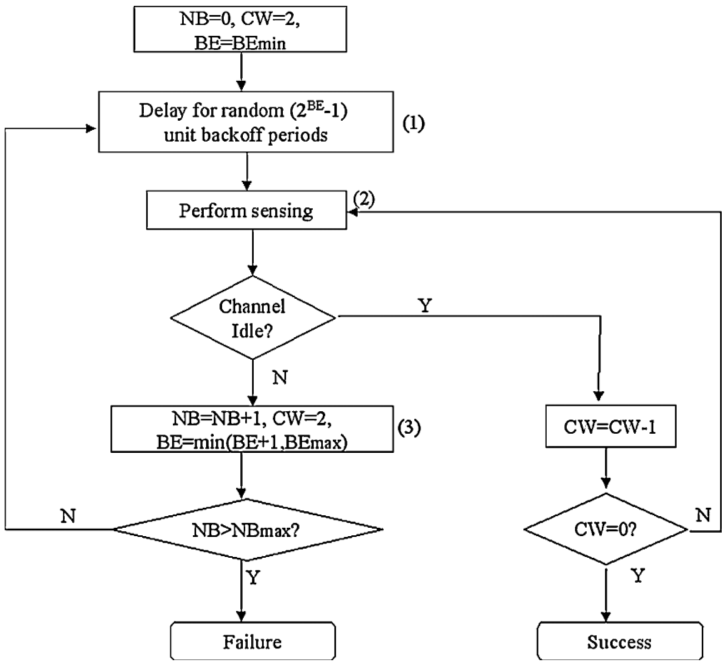


Figure 4.31 – The IEEE 802.15.4 CSMA/CA algorithm in the beacon-enabled mode with CW = 2 [59]

There are two types of devices defined by the IEEE 802.15.4 standard. The Full Function Device (FFD) contains the complete set of MAC services and can operate either as a network coordinator or a simple network device. The Reduced Function Device (RFD) contains a reduce set of MAC services and can only operate as network end device.

Two basic network architectures are supported but not completely described by the standard since the higher layers are out of the scope of IEEE 802.15.4 (Figure 4.32).

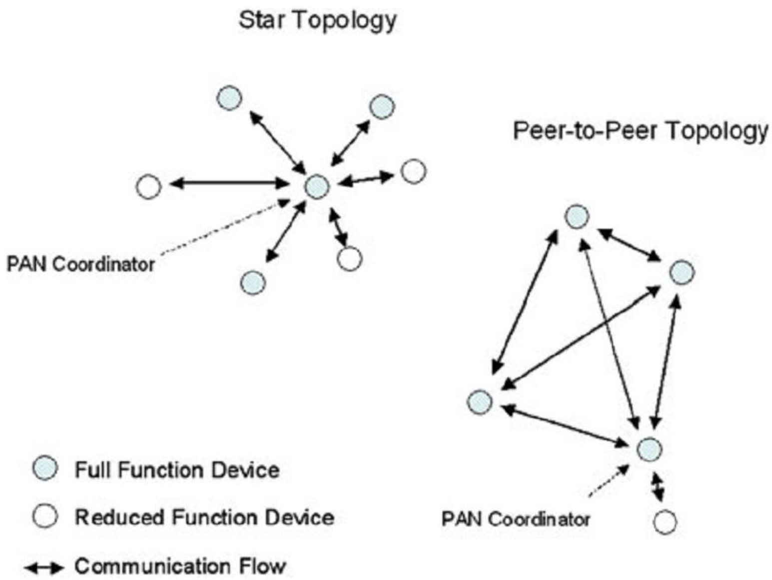


Figure 4.32 – The two IEEE 802.15.4 compliant network topologies: star and peer-to-peer topologies [59]

Start topology is usually used to cover a small area or for a system with higher latency requirements. The communication is controlled by the WPAN coordinator that acts as network master sending beacons for synchronization and device association. End devices are only allowed to communicate with the WPAN coordinator. If an end device wants to join the network it listens for the beacon signal and after send an association request to the coordinator. Star networks support also a non-beacon enabled mode. In this case, beacons are used only for the purpose of association [59].

Peer-to-peer topology is preferable when a large area has to be covered and latency is not a paramount requirement of the system. This architecture enables the formation of more complex networks and permits any FFD to communicate with any other FFD within transmission range. Also peer-to-peer topology prompts the use of multiple-hopping techniques however this feature requires additional device memory for routing tables.

All devices, regardless the topology, belonging to a particular network use their unique IEEE 64-bit address. Also a short 16-bit address is allocated by the WPAN coordinator to uniquely identify the network.

The election of the WPAN coordinator can be performed in several ways depending on the application. It can be chosen because of its special features like: having more computing power, having an important bridge capability with other protocols or simply because it was among the first participants in the formation of the network. In a peer-to-peer network several FFD devices should be deployed in order to assure that if the initial WPAN coordinator fail other device can fulfill this role.

The ZigBee protocol stack is a solution to be used on top of the IEEE 802.15.4. It has been developed by the ZigBee alliance in order to guarantee interoperability between devices from different manufactures. The ZigBee stack architecture as shown in Figure 4.33 provides the network layer and the framework for the application layer.

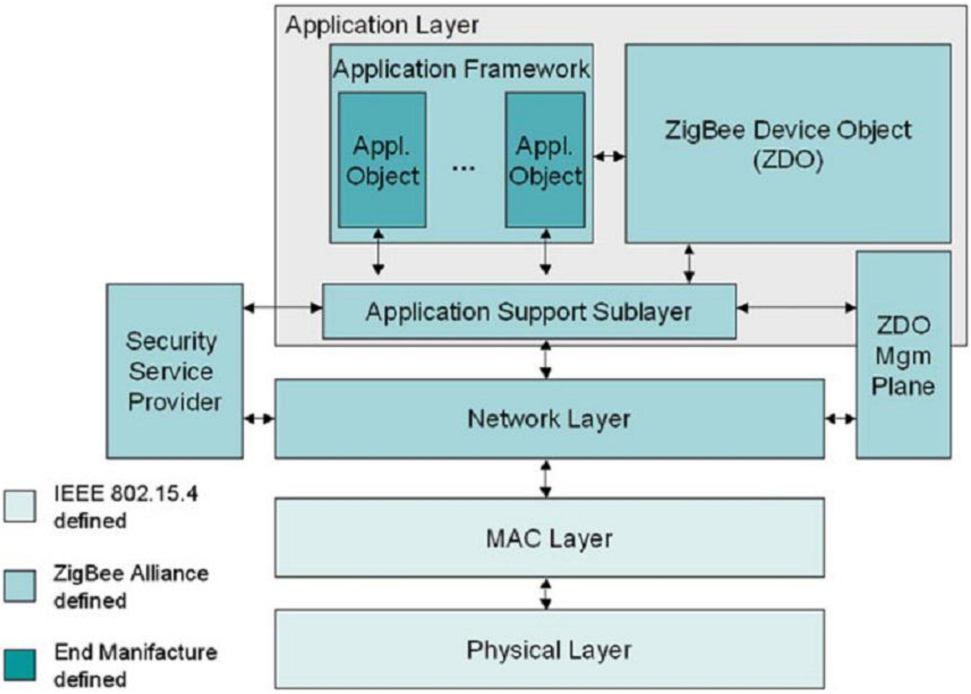


Figure 4.33 – A detailed overview of ZigBee stack architecture [17]

The network layer include mechanisms to join and leave a network, frame security, routing, path discovery, one-hop neighbours discovery and neighbour information storage.

The ZigBee application layer consists of the application support sublayer, the application framework, the ZigBee device objects and the manufacturer-defined application objects. The application support sublayer is in charge of maintaining tables for binding (defined as the ability to match two devices together based on their services and needs) and forwarding messages between associated devices. The ZigBee device objects is responsible for defining the role of the device within the network, initiating and responding to binding requests, establishing secure relationships between network devices, discovering devices in the network and determining which application services they provide [59].

ZigBee support three different topologies, star, mesh and tree. Star topologies are the simplest but not scalable for large systems with a high number of nodes. Also star architectures are usually used for small areas coverage because multi hopping schemes are not available and long range communications require more energy. Mesh and tree-based topologies on the other side are suitable for large networks distributed over large areas. Tree-based topologies employ simple routing schemes and are more efficient in regard to distributed data aggregation mechanisms. In this topology there is only one path between each couple of nodes, each node transmits its information to the node in the upper level (Figure 4.34). Although it enables very simples routing this topology is not robust to link failures. If a node disconnects all the information from its lower level nodes is lost.

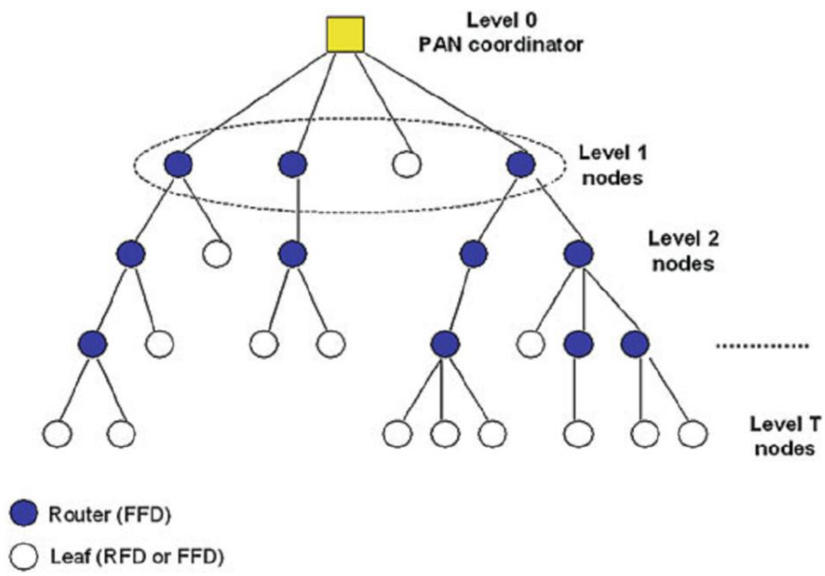


Figure 4.34 – Tree-based network topology [59]

On the other hand Mesh topology enables multiple paths between nodes. This results in more complex routing protocols but also increases the flexibility of the network which becomes more robust to link failures.

ZigBee supports three types of devices. ZigBee Routers (ZRs) are able to perform all the duties described in IEEE 802.15.4, including routing. ZigBee Coordinators (ZCs) are a type of ZRs that manage the wireless personal area network. The last devices represent the end unit, consequently, they do not have any routing capabilities and are called ZigBee End Devices (ZEDs). In comparison to the 802.15.4 standard, the ZRs correspond to WPAN coordinators, the ZCs are the Full Function Devices and ZEDs the Reduced Function Devices.

Libelium offers wireless radios compliant with these two protocols. The XBee 802.15.4 RF module supports the IEEE 802.15.4 standard. It has the two lower stacks defined by this standard which means that if the user wants to apply a more complex topology he has to develop all the routing algorithms. One of the key features of these modules is the possibility of changing the transmission power by software. This enables the construction of a more power efficient network. The XBee ZB RF module integrates the two upper layers defined by ZigBee protocol. These devices can be configured as ZCs, ZRs or ZEDs. Both XBee 802.15.4 RF and XBee ZB RF modules add some new features to the standards they employ. They both integrate node discovering algorithms and duplicated packet detection.

For the system proposed in this dissertation the IEEE 802.15.4 RF module was chosen. This choice relies on the simplicity of this module and the availability during the development process. Remembering the system functional diagram (Figure 4.35) the wireless communications should work in two star topology networks that are connected to a higher level wired network. Therefore the use of ZigBee modules would only add overhead to the frames without any major advantage. However, if in a real system the number of end devices on the star architecture increases too much, ZigBee modules should be applied and a mesh network would be better choice.

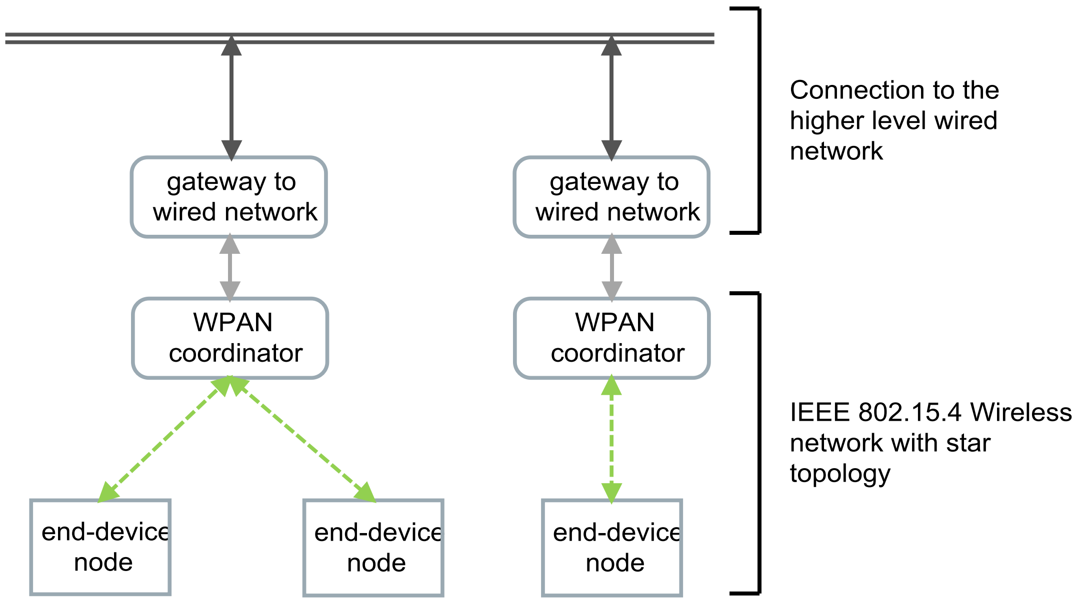


Figure 4.35 - System low level sensor network based on IEEE 802.15.4 with star topology

4.2.4 Other available modules

Other than the communication modules and sensing interface Waspmote system features a lot of specific boards that can be integrated easily. Starting by the sensor boards, Libelium has developed a number of boards with incorporated sensors for all type of measures. For example gases sensor board is aimed to monitor city pollution, emissions from farms and hatcheries, control of chemical and industrial processes and even detect forest fires. There are several boards like events sensor board, smart cities board, smart parking and agriculture board, which are easily attached to Waspmote. They are all supplied by the main battery and can be powered on or off via software. Libelium has documentation and software libraries in order to make the use of these extras extremely easy.

Not only sensor boards can be used with Waspmote. There are also some communication modules that can be of great use in specific applications. The GSM/GPRS platform integrates the SAGEM HiLo module which enables communication using the mobile telephone network (Figure 4.36). This module is able to make/receive call, send/receive SMS, connect to the internet through TCP/IP and UDP/IP sockets and provides SMTP, POP3 and FTP services, for managing emails and upload files. As for the sensor boards all these tasks have specific software libraries to be make the development easy. This model uses the USART1 at 38400bps to communicate with the microcontroller.

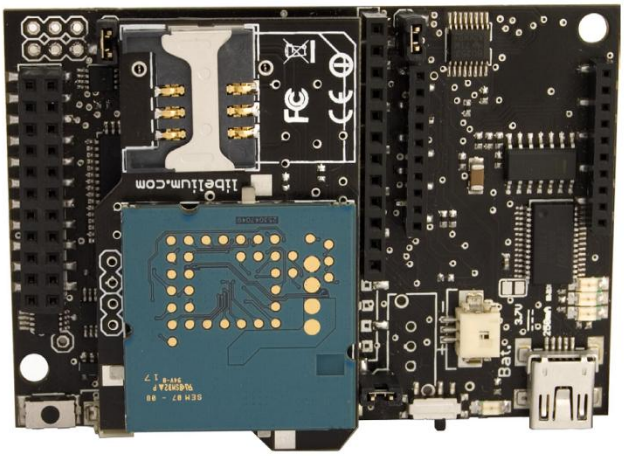


Figure 4.36 – GPRS platform connected to the Waspmote

Another available module is the GPS platform. It integrates the Vincotech A1084 GPS receiver which allows knowing the absolute outside location of the mote at anytime. It can also obtain the current time and date in order to synchronize the Waspmote RTC. Although it uses the USART1 to communicate with the microcontroller like the GSRM platform both can be integrated at the same time. This is physically possible due to a multiplexer installed.

4.2.5 Energy Concept

Waspmote has four operational modes. On normal operation everything is running full power and it consumes 9 mA. Sleep and Deep Sleep modes pauses the main program and the microcontroller passes to a latent state, from which it can be woken up by all asynchronous interruptions. On Sleep mode operation synchronous interruptions generated by the Watchdog can also woke the microcontroller. This state can last from 32 ms to 8 s. On Deep Sleep the only synchronous interruption that is able to wake up the device comes from the RTC. In this case the sleeping state can last from 8 seconds to a couple of days. Both these states consume a current of 62 µA. The last state called Hibernate is the lowest power consuming mode. All the modules including the microcontroller are completely disconnected. Only RTC is working at this point using the auxiliary battery and consuming only 0.7 µA. The only way to recover from this state is through a previously programmed alarm generated by the RTC. Next table sums up the different types of energy states that Waspmote supports.

	Consumption	Micro	Cycle	Accepted Interruptions
ON	9mA	ON	-	Synchronous and Asynchronous
Sleep	62µA	ON	32ms – 8s	Synchronous (Watchdog) and Asynchronous
Deep Sleep	62µA	ON	8s – min/hours/days	Synchronous (RTC) and Asynchronous
Hibernate	0.7µA	OFF	8s – min/hours/days	Synchronous (RTC)

Table 4.6 – Different energy modes supported by Waspmote

These different energy operation modes that Waspmotes supports enable the design of a really efficient low power system.

The last table summarize the power modes for Waspmote alone. The fact is that all the additional modules that can be installed require power to work.

4.3 Wired Communications and Gateways

There are several protocols that could work for the intermediate level network. The most used are probably Controller Area Network (CAN), Profibus and Ethernet. The main requirements for this part of the system are: distance range, data rate, robustness and cost.

On big structures like bridges distance between the nodes can be quite big. A possible scenario is the installation of a SHM system in the Vasco da Gama bridge. If sensor nodes are to be aggregated in its pillars it is expected upper nodes to communicate over 420 m which is the distance between pillars (Figure 4.37).

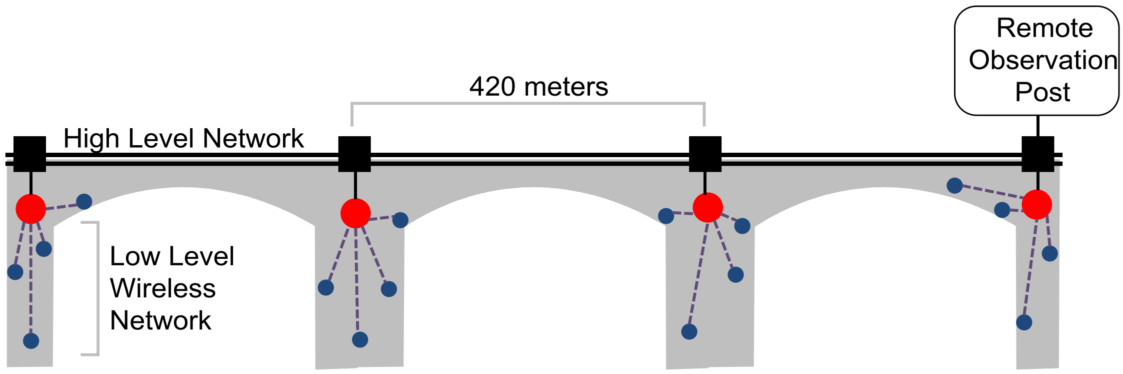


Figure 4.37 - Application of the SHM system in a big structure

The small black squares represent the gateways between the wireless communications and the higher level network. They also work as repeaters in order to keep the signal strong which makes the communication between ends of the structure possible.

The data rate and latency of the network are important in order to understand if it can throughput all the information acquired by the sensors. Assuming a sensor produces 4 bytes of information per reading and the bridge has 5 pillars with 20 sensors each; it is possible to calculate that 400 bytes of information are produced in one sampling cycle. It is possible to consider that most of the sensors used that require continuous monitoring are strain gauges and temperature sensors. Therefore sampling rates of 10 Hz maximum are the expected. With all these assumptions it is possible to say that the wired network has to exhibit a bandwidth of at least 32 Kbits/s. However a packet usually does not integrate only the information wanted; overhead has to be taken into consideration. If instead of 4 bytes each packet occupies 10 bytes the bandwidth needed is increased to 80 Kbits/s. The use of this bandwidth may never occur because distributed processing should be employed and only relevant information should get to the observation post.

Next requirement is relatively to the harsh environment in which this network is inserted. The structure under study may be situated in a highly urban environment or worst it can serve as support for railroads or subways. The wireless communications or the heavy machinery integrated in trains or subways cause electromagnetic interference which may be caught by the cables that support the network and corrupt message packets. Therefore, it is important to take into consideration the robustness of the protocol in order to avoid this kind of problems.

When trying to develop an inexpensive solution for SHM the reduced cost of all the components is a major requirement. This applies also to the wired network which means that cost benefit trade-off may be needed.

4.3.1 Controller Area Network (CAN)

In order to accomplish all these requirements the Controller Area Network (CAN) was chosen. This is a bus-based protocol, designed specifically for automotive applications but now also used in industrial automation. CAN was developed by Bosch between 1983 and 1985 with the aim of reducing the number of cables needed on the automotive industry. It uses a multi-master bus topology where every device can work as master with the capacity to start data transmission. This protocol can also be used for real-time communications although only for a short period of time. Since it is a field network, CAN protocol defines only the physical and data-link layers of the OSI model. All messages are transmitted in publisher-subscriber fashion. This implies that all packets are transmitted in broadcasting mode. Devices connected to the bus are in charge of filtering the messages received getting only the ones that are aimed to them. They do this by checking the identifier included on the message packet. This identifier is also the base for the arbitration algorithm that CAN employs [60]. The main characteristics of this protocol are described next:

- It is a multi-master protocol with multiple access to medium;
- Its top transmission data rate is 1Mbit/s for distances up to 40 meters;
- The physical medium is similar to the one defined by the RS-484: shielded twisted pair;
- All the messages have a priority level;
- Error control is implemented by hardware;
- Packets have a data field with zero to eight bytes;
- High reliability and security;
- Low cost;
- Ability to work in harsh environment conditions;
- Easy to implement.

4.3.1.1 CAN Physical Layer

At the physical level data is sent through a twisted pair cable. One wire of the pair is named CAN High (CANH) and the other CAN Low (CANL). The information is transmitted by a differential voltage signal, which improves the robustness and noise immunity. Because the two wires are twisted together if the signal in one line catches noise the other also catches the same and voltage difference remains the same.

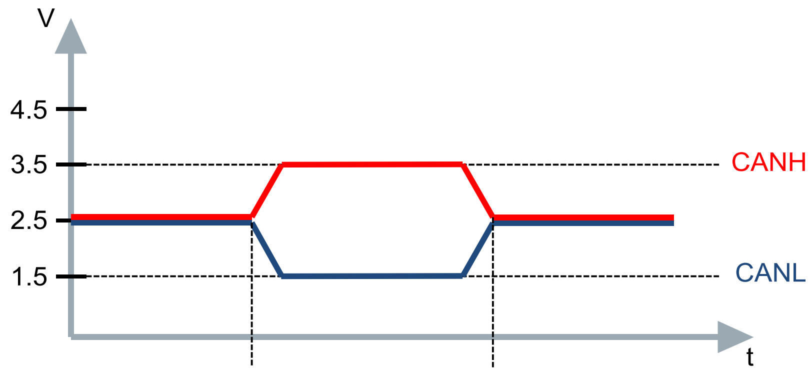


Figure 4.38 – Differential voltage signal in CAN bus

Because its immunity to noise, the use of differential voltage signals allows greater communication ranges without the need of repeaters. However due to propagation delays the use of a long bus without repeaters can cause the data rate to decrease. The next table shows the relationship between distance range and data rate, type of cables used and terminators.

Data rate	Bus length	Type of Cable / diameter	Cable resistance	Terminator
50 Kbits/s	Up to 1 Km	AWG18 0.75 - 0.8 mm^2	70 mΩ	150 - 300 Ω
125 Kbit/s	Up to 500 m	AWG20 0.5 - 0.6 mm^2	< 60 mΩ	120 - 300 Ω
500 Kbit/s	Up to 300 m	AWG22, AWG20 0.34 - 0.6 mm^2	< 40 mΩ	120 Ω
1000 Kbit/s	Up to 40 m	AWG23, AWG22 0.25 - 0.34 mm^2	< 26 mΩ	120 Ω

Table 4.7 – Relation between data rate and bus length for CAN protocol [60]

The bit encoding on CAN bus uses the Non Return to Zero (NRZ) method. The signal level remains constant over the bit time and thus just one time slot is required for its representation. If the signal remains constant for more than five bits a bit of opposite polarity is inserted. This practice is called bit stuffing and is used to detect faulty packets and to create additional signal transitions for synchronization.

CAN bus terminators are no more than simple resistors. They enable a perfect propagation of the signal through the bus assuring the reflection of the signal on the bus end. Figure 4.39 shows a typical CAN-bus integrating three devices.

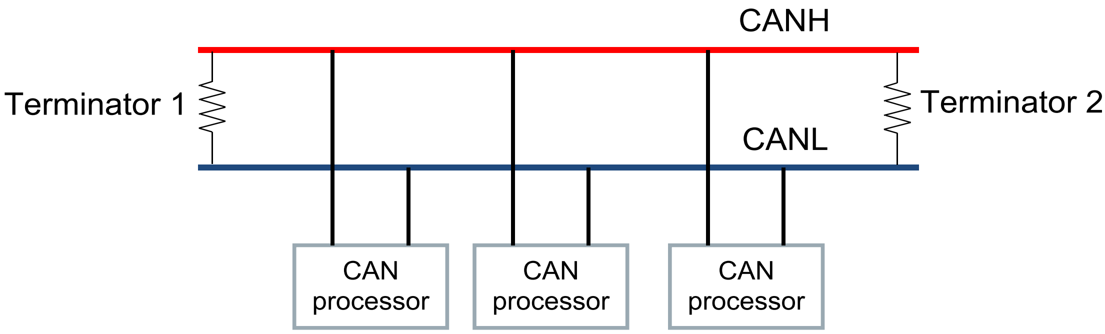


Figure 4.39 – Typical CAN bus with three devices

4.3.2 CAN Data-link Layer

The data link layer is divided in two sublayers; Medium Access Control (MAC) sublayer is responsible for framing, arbitration, acknowledgement, error detection and signaling. The Logical Link Control (LLC) sublayer is in charge of message filtering, overload notification and recovery management.

In a multimaster network, where any node is free to start a transmission, the bus access should be managed in order to avoid collision and consequently loss of packets. CAN frames are provided with a specific field (arbitration field) for handling arbitration between packets. Although a particular case, the mechanism used by CAN to control the access to the medium is of the CSMA/CA type. In the CAN specification it is called Non-Destructive Bit Wise Arbitration. It is not centralized and it grants nodes access to the bus based on priority. Whenever a node transmits a packet it starts by the frame arbitration field. While it is transmitting it is also listening to the bus. If the sender which is transmitting a recessive bit detects a dominant bit on the bus, it automatically stops transmitting and becomes a receiver of the dominant message (Figure 4.40). Once the highly priority message is sent and the bus becomes idle, the less dominant nodes are able to try again. This results in a highly efficient use of the network bandwidth [61].

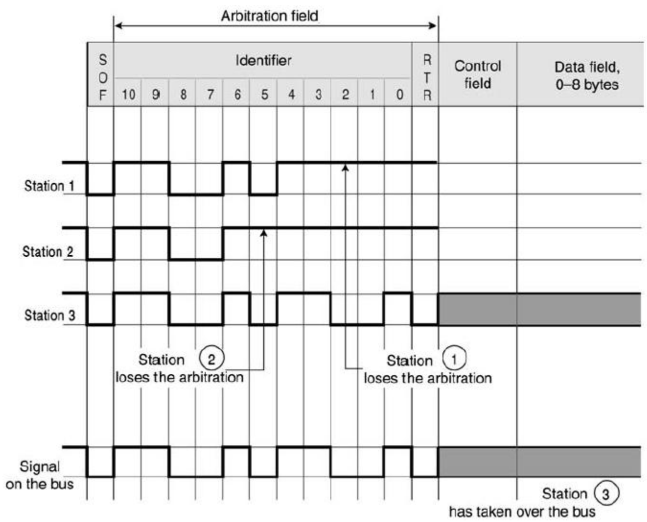


Figure 4.40 – CAN arbitration procedure in a wired-and bus [60]

The arbitration procedure implies, in a wired-and bus where the *logical one* assumes the *zero* bus value and a *logical zero* assumes the *one* bus value, the lesser a message identifier is the higher priority is granted to the frame.

4.3.3 CAN Frames

There are four types of frames defined by the CAN protocol specification. Data frames are the most common on a network and are used to data transmission. Remote frames have a similar structure but do not carry any data, they are used to request a data frame with the same identifier. The error frames are used to signalize a faulty message. They are sent immediately on the next bit after the error detection. They purposely violate the stuffing rule in order to be detected as error frames by the other nodes. Lastly overload frames have the task of creating a delay between the delivery of frames, so data is not lost by an overloaded device.

4.3.4 Error Detection Methods

Due to the fact that CAN has been designed for automotive use, where a faulty message can result in a catastrophic disaster, it has been equipped with error detection methods. These mechanisms can be applied at bit or frame level.

At the bit level, bus monitoring is one of the ways to detect errors. When a node is transmitting it is compares the value present in the line with the value it is outputting. In case they do not match a bit error is present. Also, the bit stuffing mechanism used enables the detection of errors. This method consists in adding a bit of opposite polarity after five consecutive bits of the same polarity. Therefore, errors in which the node's output stay fixed in a dominant or recessive level are detected.

The Cyclic Redundancy Check (CRC) mechanism is used to determine if the message received is complete and without errors. If the CRC determined by the sender and included in the frame does not match the one calculated by the receiver an error is detected.

Transmission errors can affect the frame's structure. Therefore, each node evaluates the received message and check whether it respects or not the frame's structure. In the case this structure is not respected an error is signalled.

The last method is based on acknowledgement of messages. If the sender does not receive any positive acknowledgment, it can conclude that the packet was not correctly transmitted and thus indicates an error frame.

4.3.5 Microcontroller for CAN Gateways

The devices used as WPAN coordinators for the wireless network are the Waspmote platforms which do not support CAN bus. Therefore it was necessary to develop a gateway between the wireless protocol and CAN. The Waspmote supports USART communications and has an available port for connection with other devices. Thus a USART to CAN gateway is proposed in this design.

For the development of the gateway it is necessary to use a microcontroller that supports both USART and CAN protocol. There are several devices with these characteristics on the market; they can vary in price, number of USART ports and available memory. Next table shows the main features of the devices considered for this role.

	Microchip PIC18F258	NEC UPD78F0881GB	Atmel AT90CAN64	Freescale Semiconductors MC9S12D64CFUE	Silicon Laboratories C8051F043-GQ
Program Memory	16 Kbytes	32 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes
Data Memory	1,5 Kbytes	2 Kbytes	4 Kbytes	4 Kbytes	4 Kbytes
Data Bus	8-bit	8-bit	8-bit	16-bit	8-bit
Number of USART Ports	1	1	2	1	2
Number of CAN ports	1	1	1	1	1
Price (Farnell)	€ 5,64	€ 4,12	€ 11,10	€ 12,49	€ 12,90

Table 4.8 – Comparison between different microcontrollers for the UART to CAN gateway

As Table 4.8 shows PIC18F258 exhibits the worst characteristics with only 1,5 Kbytes of data memory. This may be enough to implement a gateway application however for a reduced cost NEC UPD78F0881GB presents better characteristics. On the 64 Kbytes of program memory AT90CAN64 from Atmel exhibits a good performance. Its RISC enhanced architecture enables a throughput of 16 MIPS with 16 MHz of clock frequency. Although Freescale Semiconductors presents a controller with a data bus of 16-bits their IDE and compiler are not so well documented as the ones from Atmel.

The chosen microcontroller to integrate the gateway was the AT90CAN64 from Atmel. The large experience on Atmel microcontrollers was the main reason for this act. Although the other devices could do the same job, the previous knowledge in Atmel tools should decrease the development process time, which is critical in this kind of projects. Moreover, Waspmote also integrates an Atmel microcontroller therefore compatibility between these two devices is assured. Actually AT90CAN64 incorporates most of the features exhibit by ATmega 1281, which had already been explained. The main difference is the support of CAN bus protocol.

4.4 Human-machine Interface, Data Base and Website

As it was described earlier in chapter 3 the Human-machine interface should be implemented on the PC that supports the Remote Post Control unit. This PC should also be connected to a web server that is used to store all the relevant information. The website makes use of the information stored on the server in order to display the latest information about the state of the structure. The development of this top level unit is not under the scope of this document, therefore the chosen technologies are only indicated.

The tool used to implement the HMI was the Microsoft.NET framework. It supports several programming languages like C#, Visual Basic.NET and Visual C++. Due to a wide community support and ease of use, C# was chosen to develop the user interface. It is also important to state that communications between the HMI and web server are accomplished through *Structured Query Language* (SQL).

The website was developed in ASP.NET, which is a tool for web applications development created by Microsoft. It uses the same IDE as Microsoft.NET framework which represents a great advantage.

4.5 Conclusion

In this chapter the most important technologies used have been discussed. It is now clear which devices were employed and why their selection. Also the way how different modules should be integrated was described with several solutions being presented.

In order to finish this section the overall architecture of the proposed system is shown. This diagram should look similar to the one presented in the Figure 3.1 with the blocks being substituted by real devices.

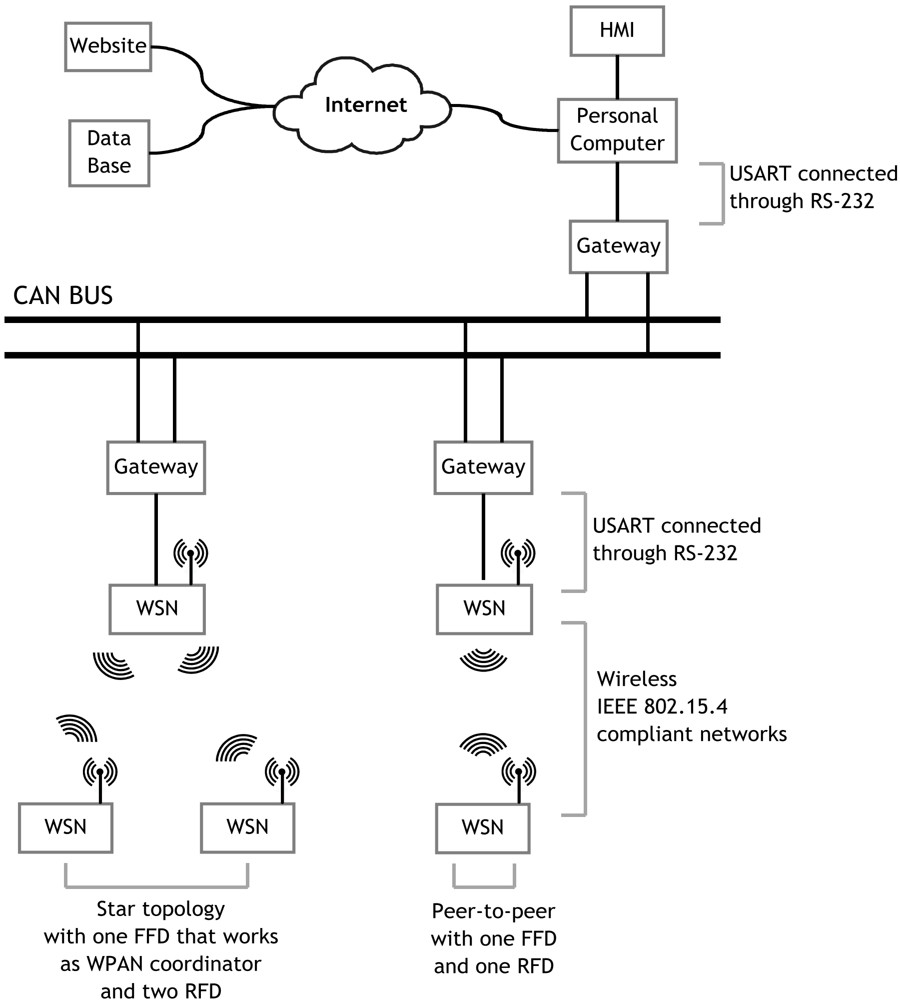


Figure 4.41 – Architecture overview where WSN blocks represent Waspmote platforms and Gateway blocks are base on the AT90CAN64

Next chapter presents the implementation process of the proposed system. All the major steps are discussed and the final prototype is shown.

Chapter 5

Implementation Process

This chapter describes the implementation process of the proposed system. It firstly presents the construction of circuits for sensors signal condition and then for the gateways. The components used are quickly described and their selection justified.

Afterwards, message frames developed to support the communication between different devices are explained. Subsequently, with all the frames analysed, the entire communication system is described.

5.1 Hardware Implementation for Sensors

On chapter 4 several sensors were discussed. It became clear that the resistive foil strain gauge was going to be used to measure strain deformations. Although a detailed description of this type of sensors was made the circuitry needed to transform the resistance change in an electrical signal was not clear. Therefore, before starting to explain the signal condition circuit implemented a brief introduction about the Wheatstone bridge is made.

5.1.1 Wheatstone Bridge

The Wheatstone bridge is an electrical circuit used to measure an electrical resistance. It is commonly used to measure the resistance change in strain gauges converting this change in a voltage signal [49]. It has some other uses and its principle can be applied to other types of transducers. However in this document only its application with resistive strain gauges is approached.

Next figure shows the Wheatstone bridge circuit (Figure 5.1), it is composed by four arms or branches formed by resistors. The excitation voltage (V_{ext}) can be continuous or alternated depending on the application. For strain gauge measurement, it is almost always continuous.

In order to measure resistance variation of the strain gauge, this sensor is applied to one arm of the bridge being all the other arms composed by resistors of known value. If the output of the bridge (V_{out}) is null it means that the bridge is in balance. Therefore, it respects the following equation $R1 \times R4 = R2 \times R3$.

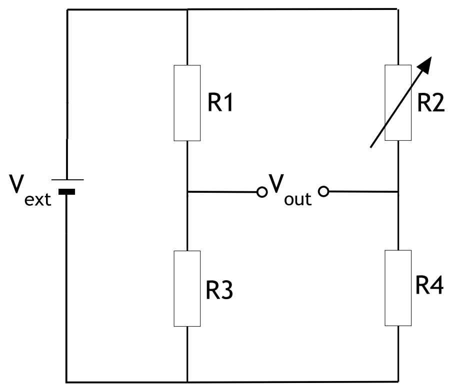


Figure 5.1 – Wheatstone bridge circuit

What this actually means is for a balanced bridge the resistance of one arm can be found knowing only the value of the others resistors. The output voltage can be expressed by the following equation:

$$\frac{V_{out}}{V_{ext}} = \frac{R_3}{R_1 + R_3} - \frac{R_4}{R_2 + R_4} = \frac{R_2 R_3 - R_1 R_4}{(R_1 + R_3)(R_2 + R_4)} \quad (5.1)$$

When strain gauges are used on a Wheatstone bridge, the balanced state is only reached with more or less deviation. Thus, most of the instruments designed to work with these sensors are equipped with balancing elements. The easiest process to balance a bridge is the integration of a potentiometer on one arm, replacing the common fixed value resistor. More sophisticated devices support auto calibration tasks.

In order to obtain strain measures using the Wheatstone bridge, resistors with the same nominal value are typically applied. Considering R_0 as the reference, the metal strain gauge resistance can be described by the expression $R_0(1 + x)$ where *x* is the variation due to strain deformation (Figure 5.2).

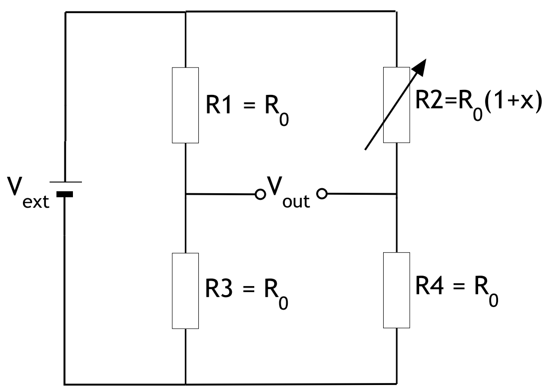


Figure 5.2 – Connection of a strain gauge to a Wheatstone bridge

The bridge arm where the strain gauge subjected to deformation and therefore resistance variation, is installed has the name active arm. The setup shown in Figure 5.2 is called quarter-bridge because it has only one active arm. Different types of setups are described further ahead.

The output voltage can be obtained using the equation (5.1) and replacing the resistors values for the ones applied in Figure 5.2. The obtained equation is then transformed in this one:

$$V_o = \frac{R_0 R_0(1+x) - R_0 R_0}{(R_0+R_0)(R_0(1+x)+R_0)} V_{ext} = \frac{x}{4+2x} V_{ext} \tag{5.2}$$

For small values of x, which are typical for metal strain gauges, the equation (5.2) can be simplified:

$$V_o \cong \frac{x}{4} V_{ext} \tag{5.3}$$

Manipulating the equation (5.3) and replacing the variation of resistance for its equivalent in strain, which is determined by equation (4.2), comes:

$$\varepsilon \cong \frac{4}{K} \frac{V_o}{V_{ext}} \tag{5.4}$$

This last expression determines the strain measured by a quarter bridge setup where the excitation voltage remains constant. It is important to understand that the linearity described by these last equations can be only considered true for small resistance variations [51].

Another advantage of the Wheatstone bridge is the possibility to use active strain gauges in more than one arm (Figure 5.3). If well positioned the use of various sensors can amplify the sensitivity of the circuit which increases the measure precision.

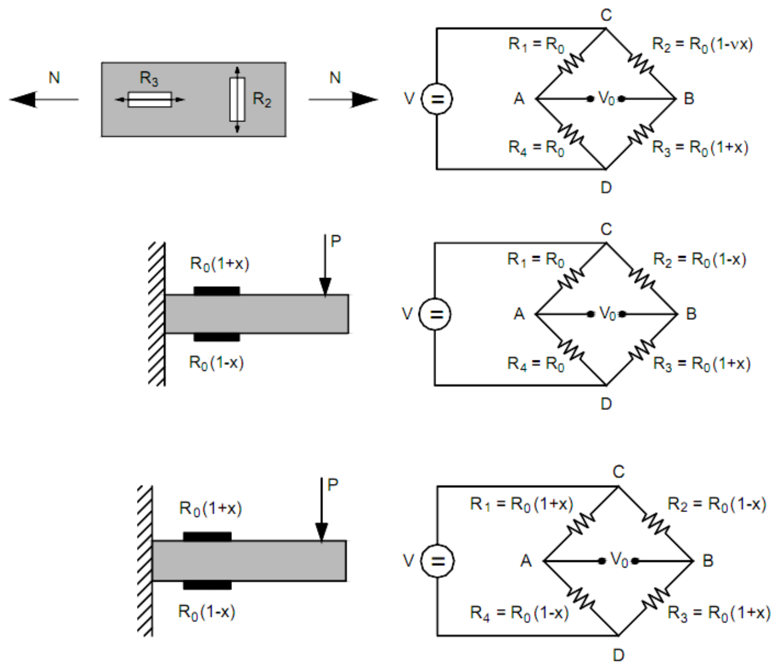


Figure 5.3 – Wheatstone bridge with more than one active arm [50]

Next table shows diverse outputs of Wheatstone bridge circuits using different configurations with one or more active arms. The excitation voltage is considered constant in all this cases.

Wheatstone bridge arms resistors				**Output Voltage**
R_1	R_2	R_3	R_4	V_{ext} constant
R_0	R_0	$R_0(1+x)$	R_0	$V\frac{x}{2(2+x)}$
$R_0(1+x)$	R_0	$R_0(1+x)$	R_0	$V\frac{x}{2+x}$
R_0	R_0	$R_0(1+x)$	$R_0(1+x)$	$V\frac{2x}{4-x^2}$
R_0	$R_0(1+x)$	$R_0(1+x)$	R_0	$V\frac{x}{2}$
$R_0(1+x)$	R_0	$R_0(1+x)$	R_0	$V\frac{-x^2}{4-x^2}$
$R_0(1+x)$	$R_0(1+x)$	$R_0(1+x)$	$R_0(1+x)$	Vx

Table 5.1 – Output voltage for different configurations of the Wheatstone bridge

On the quarter-bridge configuration the resistance variation of the wires connecting the strain gauge to the rest of the bridge may be a source of error. The distance from the strain gauge to the rest of the bridge, the type of material of the conductor and the temperature it is subjected to can influence their resistance value [51]. However all these effects can be minimized if the three-wire method is applied (Figure 5.4).

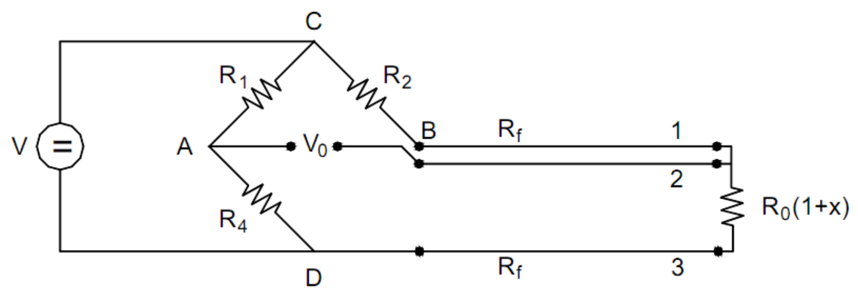


Figure 5.4 – Connection of a strain gauge in quarter bridge using three wires [50]

This method proposes the connection of the strain gauge using three wires instead of the common two. Conductors *1* and *3* have to be made from the same material, have the same length and be exposed to the same conditions. Any variation on the resistance of these cables is automatically compensated because they are in adjacent arms and have the same value and signal.

The main advantages of the three-wire connection are the maintenance of the bridge balance and the automatic compensation of thermal effects in the connection wires [51].

5.1.2 Hardware Development for Strain Gauge Sensors

The previously explained Wheatstone bridge was the solution applied to obtain a voltage signal from the variation of the strain gauge resistance. It is important to emphasize that the circuit here discussed is relative to a first iteration constructed in order to validate a design. Therefore, its performance is not expected to be incredibly good. Hence at the end of this subsection some future work is proposed.

A quarter bridge topology was used although without applying the three-wire method. As the installation of strain gauges is out of the scope of this dissertation, these sensors were applied to a steel bar by the LABEST technicians. The sensors used actually have three wires available and the method could be used. However, when the construction of this board started this was not certain. To avoid the need of reconstruction and because there was not much time for the development process, it was decided that for the first iteration strain gauges connected by two wires would be enough to get results. The resistors used on the Wheatstone bridge arms are the high precision foil resistors from VISHAY. They exhibit a tolerance of 0,01 % and a thermal drift of ±2 ppm/°C.

Like it has been previously stated the resistance variation of the resistance of a strain gauge applied to a steel bar is not big. Actually on the setup used, with the strain gauge CEA-06-250UW 350 from VISHAY, it was measured a maximum variation of 2 Ω. Furthermore the resistors that constitute the other arms of the bridge have the same nominal value as the strain gauge (350 Ω), which implies an output voltage variation on the range of milivolts. Waspmote incorporates an ADC of 10-bits and supports inputs between 0 V and 3,3 V, thus to use all its resolution the signal from the Wheatstone bridge has to be amplified. As Figure 5.5 shows the instrumentation amplifier AD 623 from Analog Devices was chosen to perform this task. It is a single supply amplifier that allows gains between 1 and 1000. It is also a low power device that consumes a maximum current of 550 µA. Its DC characteristics are quite good exhibiting inaccuracies of 0,35%.

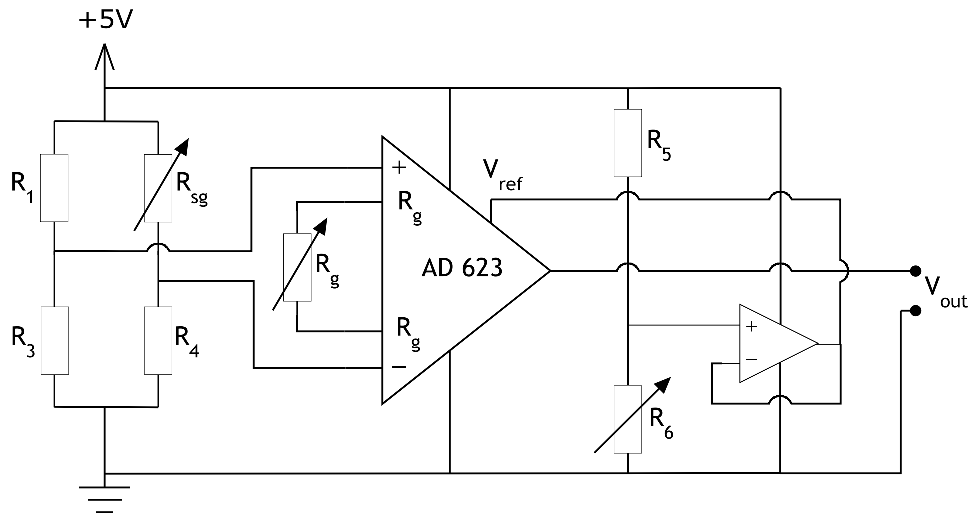


Figure 5.5 – Conditioning circuit for the Strain gauge sensors

The gain of AD 623 is set by connecting a resistor that can go from 100 KΩ, if a gain of 2 is desired, to 100 Ω for a gain of 1000. On the developed board instead of a resistor a trimmer was used. This way it is possible to manually change the gain of the amplifier to better suit the application.

Characteristics	Performance Exhibited by AD 623
Supply Type	Single supply (±6 V)
Supply Current	550 μA
Gain Range	1 to 1000
Gain Accuracy (G > 1)	0,35 %
Bandwidth	800 kHz
Slew Rate	0,3 V/μs
Thermal Drift	2.5 μV/^{o}C
Temperature Range	-40 to 80 ^{o}C

Table 5.2 – Main characteristics of AD 623

The other operational amplifier integrated in the circuit acts like a buffer to a voltage divider. Its output is connected to the reference voltage pin of the AD 623. Because all the resistors of the Wheatstone bridge have the same nominal value, if the strain gauge instead of being stretched is shrunk, the output is going to be a negative voltage. Single supply systems do not work with negative voltages, therefore a positive reference has to be added. In the platform developed this reference is obtained from a voltage divider that can be calibrated by a resistive trimmer. The operational amplifier employed is the general purpose OPA 350, which is also a single supply device.

Characteristics	Performance Exhibited by AD 623
Supply Type	Single supply (5 V)
Supply Current	5.2 mA
Differential Gain Error	0,17 %
Bandwidth	38 MHz
Slew Rate	22 V/μs
Thermal Drift	±4 μV/^{o}C
Temperature Range	-40 to 85 ^{o}C

Table 5.3 – Main characteristics of OPA 350

Using a single supply circuit is critical for this application because this platform is designed to be powered by a battery. On a future, the Waspmote will be in charge of managing the power supply of this board however for now it is supplied by an independent 9 V battery.

As part of the development process a printed circuit board (PCB) with the circuit here described replicated for six strain gauge sensors was created (Figure 5.6).

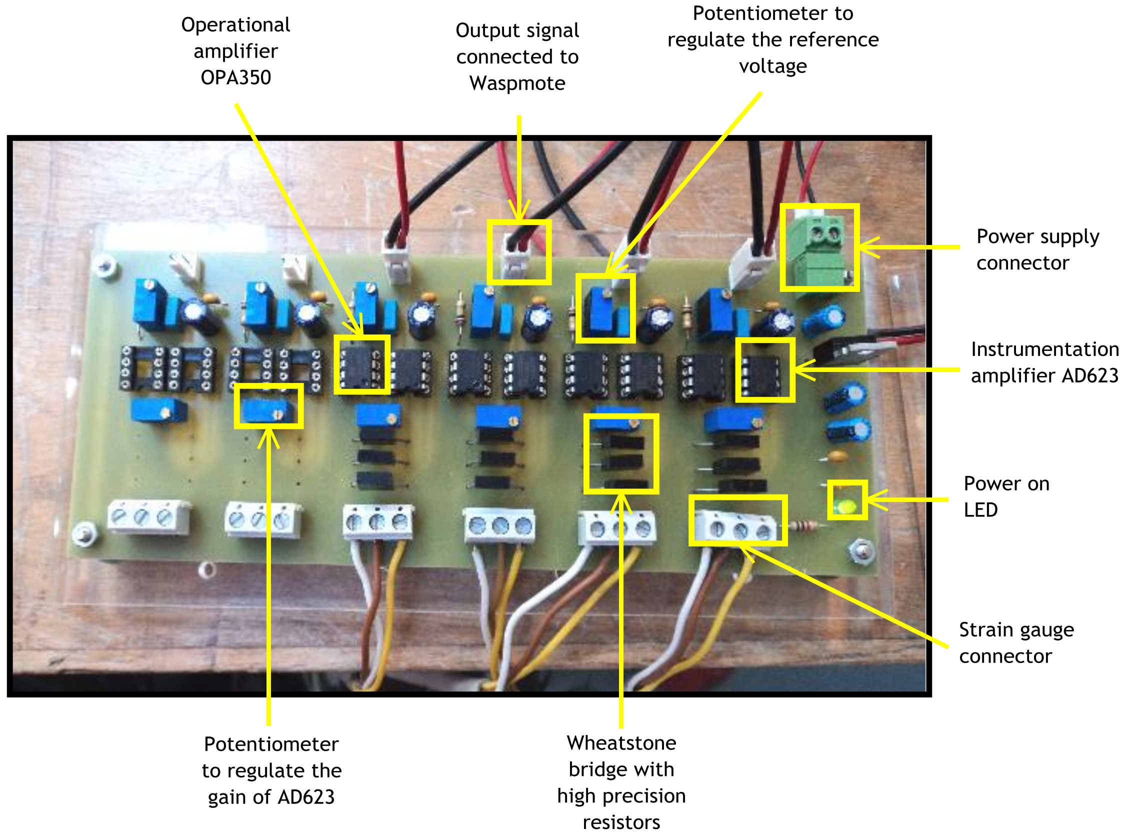


Figure 5.6 - Signal conditioning board for strain gauge sensors

After the signal passes through the conditioning circuit its relation with the actual strain is changed. The variation of the resistance of the Strain gauge is converted to a voltage and then amplified with relation to the reference voltage. The next group of equations describe this process mathematically in order to obtain the actual strain measure.

The Wheatstone voltage output is related to the resistance variation of the sensor by the next equation:

$$\frac{V_{out}}{V_{ext}} = \frac{R_3}{R_1 + R_3} - \frac{R_4}{R_{sg} + R_4} = \frac{R_{sg}350 - 350^2}{(350+350)(350+350)} \tag{5.5}$$

The excitation voltage applied to the bridge is constant and its value is 5 V. Therefore:

$$V_1 = \frac{R_{sg}350 - 350^2}{(350+350)(350+350)} \times 5 \text{ (V)} \tag{5.6}$$

Afterwards V_1 is amplified by a factor of 1000 by the AD 623 (the potentiometer R_G is calibrated to 100 Ω).

$$V_2 = V_1 \times 1000 \text{ (V)} \tag{5.7}$$

As it was previously described the AD 623 amplifies the voltage providing from the Wheatstone bridge with relation to an offset. The voltage reference (offset) considered is 1.65 V.

$$V_3 = V_2 + 1.65 \text{ (V)} \tag{5.8}$$

Only after this process, the analogue voltage is converted to digital. The relation is given by the next expression:

$$Digitalvalue = \frac{V_3 \times 1023}{V_{ADC}}, \quad V_{ADC} = 3.3V \tag{5.9}$$

So the result of the measure given by the Waspmote is relative to end of this process. In order to obtain the resistance value measured on the Wheatstone bridge the inverse process has to be made. Afterwards using the equation (4.2) it is possible to get the actual strain.

$$Digitalvalue = \left(\frac{5}{1,4} \times R_{sg} - 1248,35\right) \times \left(\frac{3,3}{1023}\right) \tag{5.10}$$

$$R_{sg} = 350(1 + K\varepsilon), \; K = 2,1 \tag{5.11}$$

$$Digitalvalue = (1,65 + 2625\varepsilon) \times \left(\frac{3,3}{1023}\right) \tag{5.12}$$

From the last set of equations is possible assess that the sensitivity of the signal condition circuit, including the Wheatstone bridge, is 2625 V/ε. Also, it is possible to determine the resolution of the entire system. This value is determined by the next equations.

$$Q = \frac{3,3}{2^{10} - 1} = \frac{3,3}{1023}, \; \text{Q is the Quantum value of the ADC} \tag{5.13}$$

$$Resolution = \frac{3,3}{1023} \times \frac{1}{2625} = 1,22 \times 10 - 6 \; (\varepsilon / DigitalValue) \tag{5.14}$$

From the result of the equation (5.14) it is possible to state that from measures on the micro Strain range this ADC does not provide enough resolution. Although it is possible to use more amplifier stages it should be reminded that noise will also be amplified. For a final system an ADC with at least 14-bits of resolution should be added. Because the sampling frequency is not critical on this type of measures it is possible to neglect this characteristic for more resolution. There are several devices on the market that are cheap and would give a much better performance.

The excitation voltage of the Wheatstone bridge is given by the LT7805 linear regulator. This device output a regulated voltage of 5 V when supplied with at least 7 V. On the designed circuit several capacitors have been connected to the DC bus in order absorb noise or other oscillations. However the LT7805 presents an output voltage drift of -1.1mV/°C. As the Wheatstone bridge output is sensitive to the excitation voltage this may cause small inaccuracies on the measurement. Another problem is the use of a trimmer to calibrate the voltage reference. It is almost impossible to set the exactly wanted reference manually. One way of solving this problem is the use of digital potentiometers with high precision. These devices can be controlled by the microcontroller making the measure more reliable and allowing auto calibration.

One way of solving all these problems is using a specific IC for strain gauge measurements. One example is presented by PICOSTRAIN with the model DB PS09. It has an embedded microcontroller and other specific hardware to drive Wheatstone bridge with the maximum accuracy. This device has an ADC of 28-bits and can be connected to the Waspmote through I^2C or USART. It enables the control of the excitation voltage and bridge calibration all by software. Moreover it consumes only 15 µA which makes it suitable for a low power application like the one presented in this project. Due to the lack of time it was not possible to test this IC; however, on a next iteration its use would be profitable.

Signal filtering was not employed at all. On laboratory it was used some passive low-pass filters to eliminate signal noise. Due to the static nature of these measures filters will only be used to eliminate white noise. Therefore, they should be designed depending on the environment where the system is to be deployed.

5.1.3 Signal Conditioning Board for Accelerometers

The acceleration is measured using the Analog Devices MEMS sensor ADXL203. It is a single supply component whose specifications are described in the Table 4.2. As it has been explained MEMS sensors incorporate amplification systems and this is no exception. Its output is ranged between 0 and 5 V where 2.5 V represents a no acceleration measure.

The sensor is mounted on an evaluation board that has two factory capacitors connected to its outputs (Figure 5.7). The 0.1 µF capacitors installed reduce the sensor bandwidth to 50 Hz. Vibrations of interest are on the range of 10 Hz therefore this limitation poses no problem for this application. Respecting the Nyquist’s law, the sampling frequency has to be of at least 100 Hz in order to avoid aliasing.

Figure 5.7 – ADXL203EB evaluation board

The ADC used by Waspmote accepts only 0 to 3.3 V signals. Hence the output of this sensor has to be scaled. Next figure shows the circuit used to accomplish this task. Its core is a voltage divider that can be manually calibrated through a resistive trimmer.

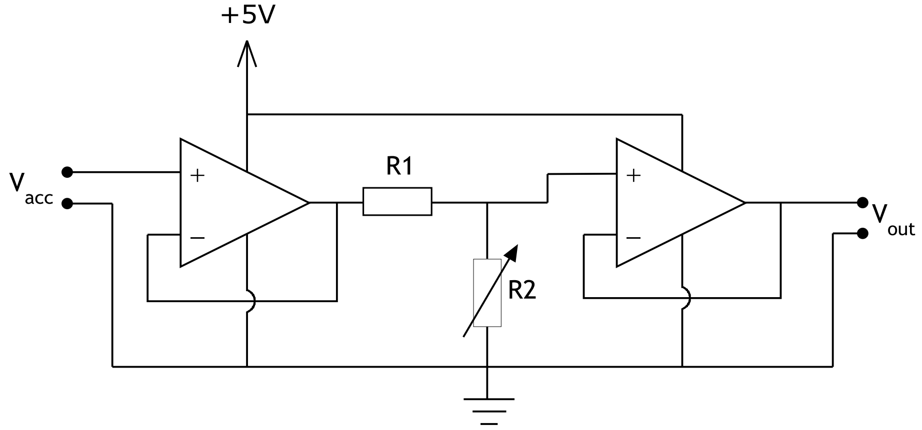


Figure 5.8 – Signal conditioning circuit for the ADXL203

In order to avoid interference caused by the resistors that compose the voltage divider two buffers were employed. The operational amplifier used is the OPA2350, which integrated two OPA 350 amplifiers. The next equations describe the effect of the signal condition circuit on the sensor output.

$$V_2 = V_{sensor} \times 0{,}66 \text{ (V)} \tag{5.15}$$

The sensitivity of the sensor is 1 V/g, with the scale factor it is decreased to 0,66 V/g. Considering the Waspmote ADC the resolution of the entire system is given by the next mathematical expression.

$$Resolution = \frac{3{,}3}{1023} \times \frac{1}{0{,}66} = 0{,}0049 \ (g/DigitalValue) \tag{5.16}$$

Therefore it is possible to measure variations as small as 0,0049 g. For this stage of the development process this resolution is enough to validate the system concept. Although for sampling frequencies up to 100 Hz, there is a vast number of ADC with higher resolutions for reasonable prices.

The final circuit is shown in the next image. It is important to remember that each accelerometer has two outputs representing the horizontal and vertical axis. Therefore design previously discussed is applied twice.

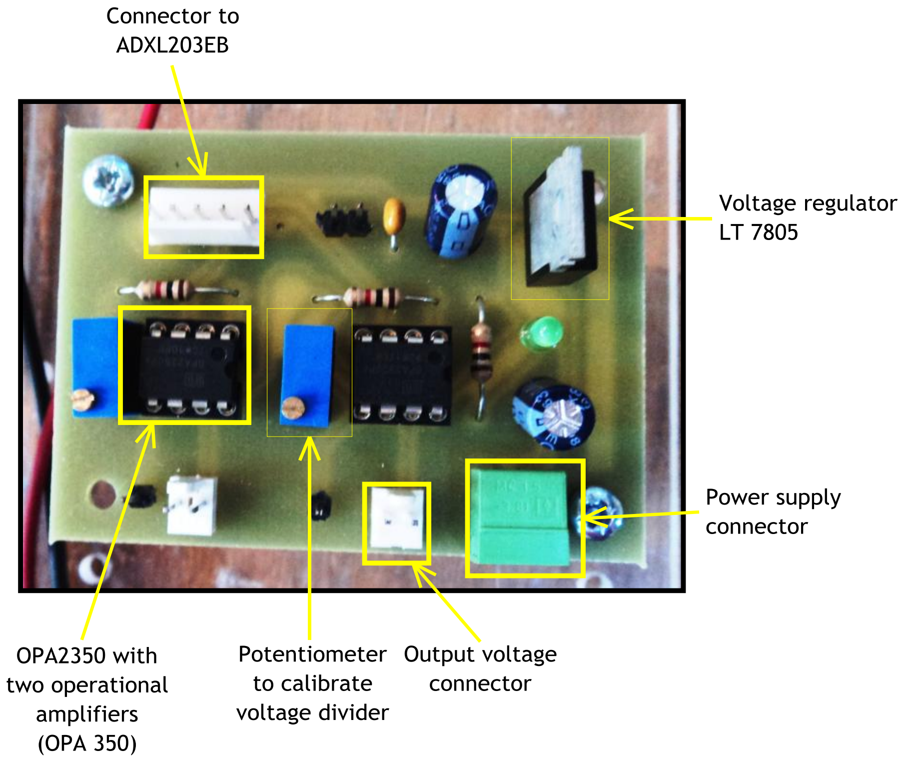


Figure 5.9 – Signal condition circuit for the accelerometer

Like the strain gauges board, this platform is powered through the linear voltage regulator LT7805. The power source is also a 9 V battery. In future work the Waspmote should be able to manage the energy consumption of this circuit.

5.2 Hardware Implementation for CAN to RS-232 Gateways

The need of using gateways has already been explained. There are devices available on the market that can fulfil all the requirements. However, in order to reduce the cost of the system it was decided to develop a solution capable to connect both protocols.

The gateway proposed connects to the Waspmote through the USART1 port. Like it was said the AT90CAN64 from Atmel is the microcontroller in charge to bridge CAN to USART. Also both transceivers for CAN and RS-232 were used. The transceiver used to connect the CAN protocol controller to the physical bus is the MCP2551 from Microchip. This device provides differential transmission and reception capability for the CAN controller. It is capable to operate at speeds of 1Mb/s and integrates a buffer between the microcontroller and the high-voltage spikes that may occur on the CAN bus.

On the RS-232 side the transceiver used is the well known MAX232 from MAXIM. This device uses charge-pump capacitors to generate the required voltages for RS-232 communications using a single 5V power supply. It operates with data rate up to 120 Kbit/s and integrates two transmitters and two receivers.

The Atmel AT90CAN64 microcontroller is connected to the CAN transceiver through the ports 30 and 31 and to the MAX232 through the USART1 ports. On the printed circuit board designed, a connector for SPI communications is available. This connector was used, mostly, to program the microcontroller using the AVRISPmkII programmer from AVR.

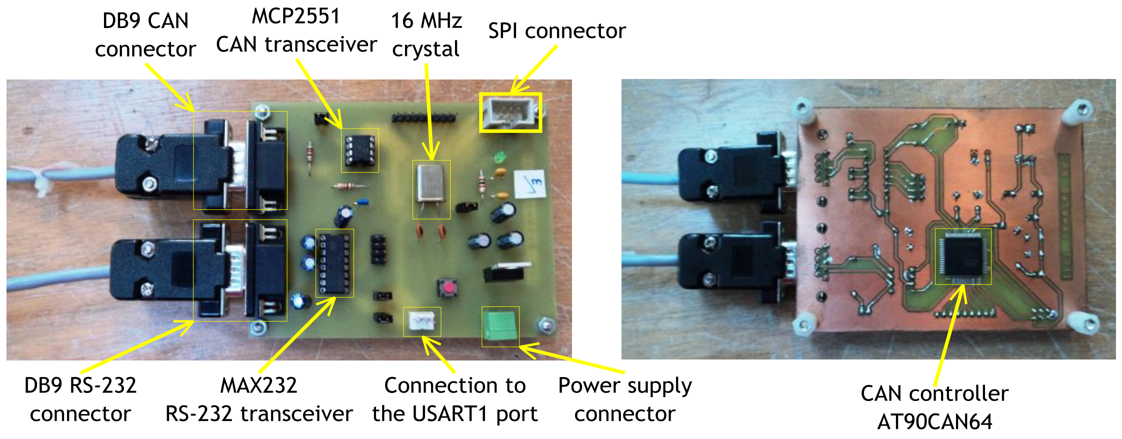


Figure 5.10 – Developed CAN to RS-232 gateway

Like on the other boards the circuit is powered through the LT7805 voltage regulator. However in this case the power supply does not need to be portable. Therefore the electrical grid used to power the Remote Control Post is also used to supply these gateways.

5.3 Communication Frames Development

In order to support the communication between the various equipments on the system, it was developed a group of message frames. Although, most of the fields defined by these frames are common to both networks, the way they are implemented in CAN and wireless communications is different. This is mainly due to restrictions imposed by CAN and IEEE802.15.4 protocols. Each frame type is relative to a determine function of the system. Therefore, seven different types of messages were created:

- Type F – is used to define the task the system should perform and to activate sensors;
- Type R – is used to ask readings from active sensors;
- Type C – is used to synchronize the RTCs of the different wireless units;
- Type A – asks for the state of the actuators on a determine unit;
- Type M – is used to set the state of the actuators;
- Type W – asks for alarms from a determine device;
- Type L – is used to get 200 consecutive readings from a sensor.

All the frames start with the character '~' and finish with '|'. Also, the different fields of the frame are all separated by the character '#'. This way it is easy to determine whether a frame is being received or not. Moreover, the separation character allows the separation of the string received in the different fields.

Although they are considered from different types, all the frames have the same header and footer. Only the data field structure varies, thus, a generic frame is shown in the next image.

~	idMaster	#	idSlave	#	idMessage	#	frameType	#	data field	%	CRC	|

Figure 5.11 – Generic message frame structure

After the starting character is the *idMaster* field. As it has been stated the system implemented works in a master-slave fashion with the Remote Control Post being the central master. For this unit the *idMaster* field represent the first destination of the message. The master id on this cell is relative to the WPAN coordinators. On this first prototype there are two WPAN coordinators taking the id 1 and 10. When the wireless network coordinators receive the message they route it to the slave defined by the *idSlave* field.

Next field represents the message id. After sending a frame the central master expects to obtain a reply. The message id of the sent frame is the same of the reply message. Thus, central master can manage which messages are still waiting for reply. This also allows the transmission of various frames before the answer of the first one has been received.

The *frameType* field defines which type of message is being sent. It can take the values previously described (R, F, C, A, M, W, L). This parameter enables the receivers to understand how they should treat the data field. The data field depends on the message type and its structure is presented on the next subsection.

The last field is divided from the rest of the frame by '%'. It is composed by the Cyclic Redundancy Check. The different divider character indicates to the receiver to break the message from this point. Afterwards, the receiver calculates the CRC and compares it to the one that was indicated by the message. If they are the same the frame received is not corrupted.

5.3.1 Frame Type F

This frame was developed in order to allow the user to choose what task the system should perform. Through this frame it is possible to define what sensors should be active and the operation mode of slave. There are three operation modes available, read sensors continuously, read sensors during a defined period of time and sleep mode. Some of these operation modes are not fully available for example continuous reading mode only allows the user to choose which sensors should be active.

Depending on the operation mode wanted, the data field of this frame type can take different forms.

~	idMaster	#	idSlave	#	idMessage	#	'F'	#	"sleep"	%	CRC	|
~	idMaster	#	idSlave	#	idMessage	#	'F'	#	"continuousA"	%	CRC	|

Figure 5.12 – Type 'F' frames structures representing two different modes of operation

Figure 5.12 shows two examples of type 'F' frames. The first one sends the slave to sleep mode as the other commands it read continuously all sensors. These are the simplest examples of this type of message. If the user wants the slave to read continuously some specific sensors, the type 'F' frame will look like the following.

Header	'F'	#	"continuousS"	#	NumOfSensors	#	id_sensor1	:	id_sensor2	%	CRC	|

Figure 5.13 – Structure of a type 'F' for reading continuously some specific sensors

Messages that order readings for an interval of time can also define specific sensors or not. Again if no sensors are specified the slave will read all sensors available to him. These messages also define the period of time when the measures should be taken.

The central master is in charge of transmitting these frames; however slaves are expected to reply. If the order transmitted by these messages is accomplished with success, slaves should reply with the exactly same frame. On the other hand, if an error occurs, slaves do not reply causing the central master to detect a time out. On a future version error messages should be developed.

5.3.2 Frame Type R

The use of this frame is very simple. It asks for the readings from active sensors on a determined slave. The user can specify the active sensors using the 'F' type frames and then use these frames to ask for readings. Next figure shows the structure of an 'R' frame sent by the central master.

~	idMaster	#	idSlave	#	idMessage	#	'R'	%	CRC	|

Figure 5.14 – Structure of a type 'R' frame sent by the central master

The reply sent by the slaves is a bit more complicated. It adds the number of sensors read their ids and the measured value. An example frame is presented next.

Header	'R'	#	NumOfSensors	#	Id_sensor1	:	Value_read	%	CRC	|

Figure 5.15 – Structure of a slave reply for type 'R' frames

The frame size can increase until a maximum of six sensors read. In order avoid sending float type values, the *Value_read* parameter is filled with the result of the ADC. Therefore the central master is the one responsible to the conversion of this value to actual measurement.

5.3.3 Frame Type C

This message is sent in broadcast to the entire system. It is used to synchronize the RTCs of all the units. The synchronization process is started by the user with his interaction with the human-machine interface. All the nodes are synchronized by the clock of the central master. Thus, it is needed to send the actual date and time. If the synchronization is successful the central master receives the exact same frame from both WPAN coordinators.

As it is shown on figure Y the synchronization frame has fields for the year, month, day, weekday, hours, minutes and seconds.

Header	'C'	#	year	#	month	#	day	#	weekday	#	hour	#	min	#	sec	%	CRC	|

Figure 5.16 – Type 'C' frame structure

The synchronization process starts on the central master like it was said before. It sends the same frame to both WPAN coordinators. Upon receiving this frame WPAN coordinators are responsible to update their clocks and transmit the same message to its nodes. If no errors found, the end-nodes reply to their masters with the same frame. The WPAN coordinators retransmit the message received to the central master.

5.3.4 Frame Types A and M

These messages are used to get the state of the actuators and to change them. Frames of the ‘A’ type are used to ask slaves for the state of their actuators. The structure of the request message has only the actuator id and the reply adds its state (Figure 5.17).

<table>
<tr><td>Header</td><td>‘A’</td><td>#</td><td>idActuator</td><td>%</td><td>CRC</td><td>|</td><td></td><td></td><td></td><td>request</td></tr>
<tr><td>Header</td><td>‘A’</td><td>#</td><td>idActuator</td><td>#</td><td>stateActuator</td><td>%</td><td>CRC</td><td>|</td><td></td><td>reply</td></tr>
</table>

Figure 5.17 – Request and reply structures for the ‘A’ type frames

The ‘M’ type frames are even simpler. They are used to set the state of an actuator. If no error occur the request message is equal to the reply message (Figure 5.18).

<table>
<tr><td>~</td><td>idMaster</td><td>#</td><td>idSlave</td><td>#</td><td>idMessage</td><td>#</td><td>‘M’</td><td>#</td><td>idActuator</td><td>#</td><td>stateActuator</td><td>$</td><td>CRC</td><td>|</td></tr>
</table>

Figure 5.18 – Structure of the ‘M’ type frames

Remember that all the reply messages have the same idMessage of the requests. This allows the central master to understand to what request is a unit replying.

5.3.5 Frame Type W

The ‘W’ type frame is more complex than the ones already described. It is used to request detected alarms from the different devices on the system. It can be sent by the central master to the WPAN coordinators or by the WPAN coordinators to their slaves. The basic mode of operation can be divided by the two different types of networks. The masters of the wireless networks should periodically ask for alarms from their slaves. If any alarm is detected they store it. On parallel the central master periodically asks them for alarms stored. This way it is possible to respect the master-slave architecture and keep the alarms updated. Remember that structural changes are slow, therefore, it is not critical to have an event based architecture in order to detect alarms.

This message is sent in broadcast to both WPAN coordinators and from these devices is also broadcasted to the end-nodes. The structure of the frame is composed by one field that determines if the message is a request or a reply. Other field determines the type of alarm detected and the last one the sensor value that triggered the alarm.

<table>
<tr><td>Header</td><td>‘W’</td><td>#</td><td>AlarmType</td><td>#</td><td>AlarmDescription</td><td>#</td><td>AlarmValue</td><td>%</td><td>CRC</td><td>|</td></tr>
</table>

Figure 5.19 – Generic structure of a ‘W’ type frame

The *AlarmType* field determines a request message by using the character ‘R’. On the reply this field is filled with the same character if there are no active alarms or by another character representing the generic type of alarm. The characters which may appear on this field are:

- R – indicating a request or a reply with no alarms detected;
- T – indicates that a temperature reading exceeded a defined threshold;
- E – indicates that a strain reading exceeded a defined threshold;
- A – signal that a threshold was exceeded by an acceleration measure;
- S – indicates that a sensor may be corrupted;
- B – indicates low battery.

Next field regards a more specific description of the alarm. If a 'U' is exhibited it means that the upper limit of the threshold was exceeded. On the other hand if this field has the character 'L' it means that the lower limit was surpassed. If no alarm was detected it presents a 'R'.

Last field has the id of the sensor that triggered the alarm and the read value. Again if no alarms are detected this parameter exhibits a zero.

5.3.6 Frame Type L

This frame was developed regarding the accelerometers. The sampling rate of these sensors is about 100 Hz. Therefore requesting single values of acceleration does not allow the vibration analysis of the structure. In order to surpass this problem the 'L' type frame allows messages with 200 measures. Figure 5.20 shows the generic structure of a request using an 'L' type frame.

Header	'L'	#	SensorID	#	MsgIndex	#	MsgNumber	#	MeasuresNumber	Footer

Figure 5.20 – Generic structure of a request using 'L' type frames

The *SensorID* field determines which sensor is to read. Next is represented the message index. This parameter is important on the messages exchange process analysed below. The following fields indicate the number of messages and the number of measures. The exchange process is the following:

1. The system master sends a request message with index 1. This message has only the id of the sensor and the *MsgIndex* field filled.
2. The device that receives the request frame replies with an acknowledge message with the index 2. This frame has also the number of messages that are required to send all the measures. The last field is also filled with the number of measures that are going to be made.
3. Afterwards the central master reply to the WPAN with the same frame which means it is prepared to receive the messages with the measures.
4. Next the WPAN device orders its slave to start reading the sensor in cause. The end-node reads the sensor and sends the measures to the master that accumulates them in packets of 20 measures.
5. After, all the packets are sent to the central node. When the process is finished the central master sends a frame with index 3.

5.4 Software Implementation for Wireless Nodes

After defining the generic communication protocol it is important to explain how it was implemented. Also the software structures created have to be described in order to explain the operation mode of the system.

The platform used to support the Wireless Sensor Nodes (WSN) is the Waspmote. This device has already been described in detail on chapter 4. It was also stated that the XBee IEEE 802.15.4 RF module was chosen for the wireless communications. Therefore its configuration is described on the beginning of this section.

Afterwards the software architecture developed is discussed. The operation mode of the WSN is described and its integration with the different communication frames is presented.

In order to close this section the overall operation mode of the wireless part of the system is presented.

5.4.1 Configuration of the XBee IEEE 802.15.4 RF Module

Libelium provides a software library developed in C++ language, which has functions for nodes configuration. In order to create a network only two parameters are necessary. In order to be differentiated from other networks a personal area network has a unique 16-bit identifier. So the first step was to set all the devices to this PAN ID. This was accomplished using the libraries provided by Libelium.

The other important parameter is the channel on which the devices communicate. As it was explained earlier IEEE 802.15.4 supports 16 different channels. There are two possibilities when choosing a channel. It is possible to assign a random channel from the 16 available or choose one based in an energy scan. The second possibility is more advisable because it detects the channel with lower energy and assigns it to the network. The energy of a channel is related to the number of devices that are using it to transmit. It is higher if more devices are connected through it which increases the probability of collision. Therefore the channel with minimum energy is the better suited for a new network. After all devices are configured with the same channel and PAN ID the network is formed.

The Application Programming Interface (API) developed by Libelium allows the configuration of more network parameters. The Clear Channel Assessment (CCA) threshold, which defines the maximum energy that a channel can exhibit for a node to transmit, can be defined by software. This feature allows the optimization of the CSMA/CA process and can be configured in regard to the place where the system is deployed. Also the number of retries and the minimum value at which the back-off algorithm starts can be configured.

Another important characteristic is the output power level at which the RF module transmits. The API allows the use of 5 different levels which allows a more energy efficient system.

The message frame structure is defined by the RF modules used. However, Libelium developed an application header that is used treat the received packets in an easier way (Figure 5.21).

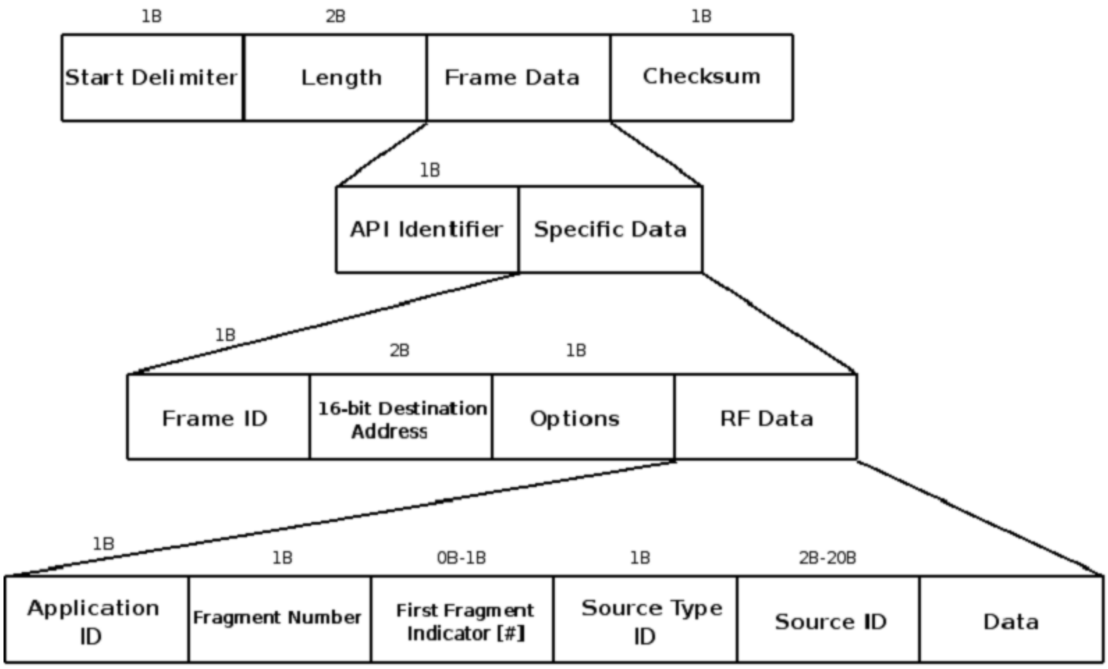


Figure 5.21 – Packet frame structure used by Waspmote

The additional header is filled by Waspmote before the packet being sent and treated when received by functions developed by Libelium. The maximum payload of data that one packet can transmit is 100 Bytes. However, using the Libelium API, up to 1500 Bytes can be sent. The API provides fragmentation capacity using internal functions. These payloads allow the direct implementation of the protocol frames developed.

5.4.2 Software Architecture of Wireless Sensor Nodes

In order to keep the programming code implemented on the wireless nodes scalable and reusable, different data structures were created. This way it is possible to increase the network without having to reprogram all the nodes and even, reutilize most of the code from the original platforms to the new ones.

The data structure sensor was developed in order to store sensors main attributes like name, id, last value read and its state. The same approach was used for actuators, although, in this case only name, id and actual state are stored. These structures are aggregated to the *slave* data structure, which in addition indicates: the slave id, number of sensors, number of actuators and the current operation mode (Figure 5.22). Also three data structures with date variables are connected to the slave. They are mainly use for functions that are only active for a period of time.

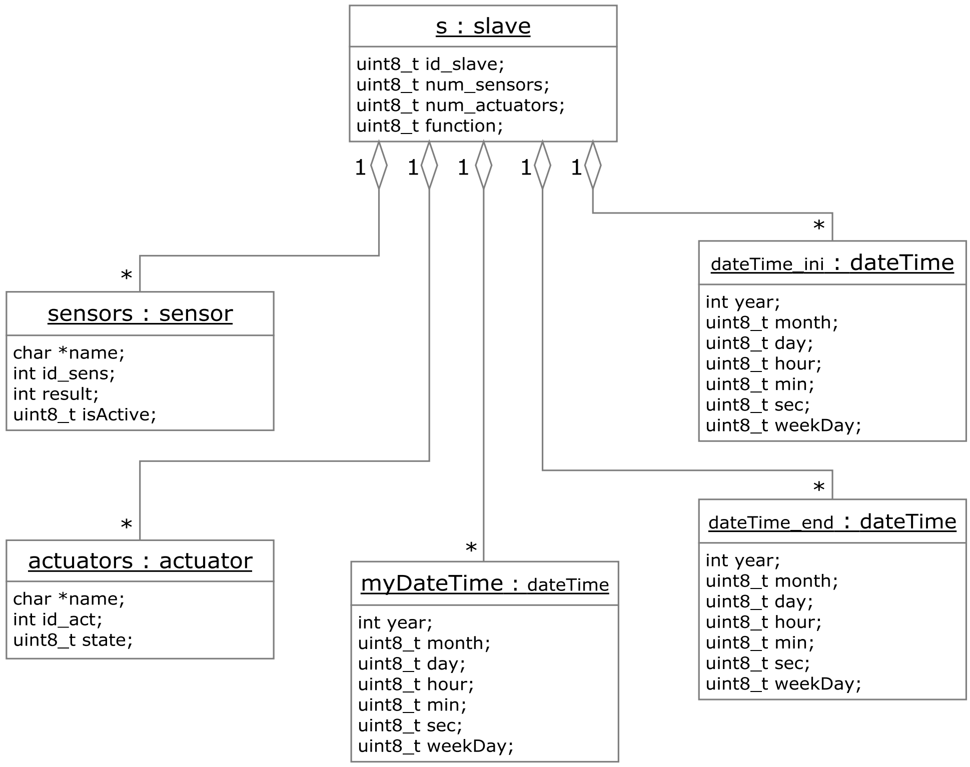


Figure 5.22 – Composition of the *slave* structure

The WPAN coordinators act like masters on their networks. Therefore a data structure with the parameters related to the master was created. This data structure is composed by various slave structures and a date structure. In addition, the variables that indicate the master id and the number of slaves allocated to it are also stored (Figure 5.23).

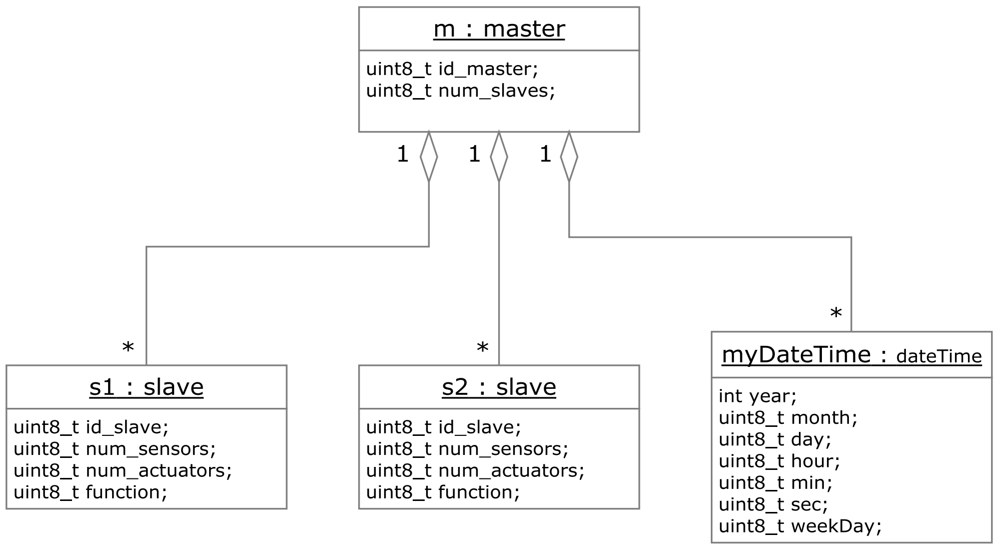


Figure 5.23 – Composition of the *master* structure

As message frames need special treatment a data structure to handle all its fields has been developed (Figure 5.24).

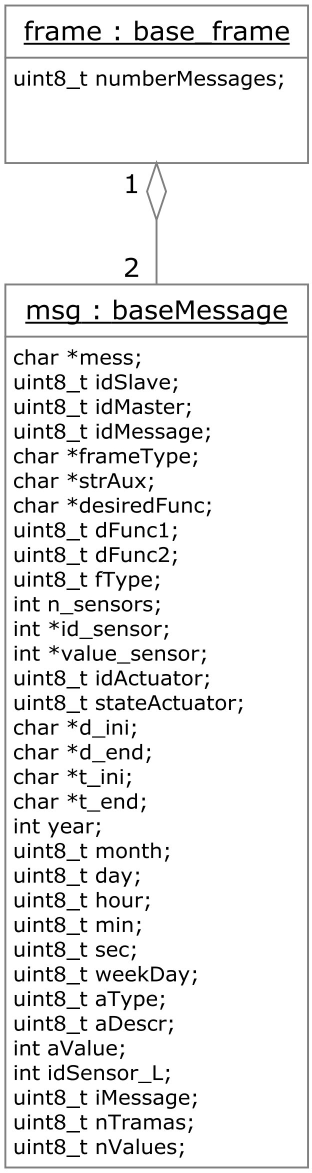


Figure 5.24 – Data structure for message frames treatment

The stored variables are related to the different parameters that may be sent in a packet. One implemented feature was the possibility of sending two frames in the same packet; therefore the data structure *base_frame* can incorporate two *baseMessages*. The use of these structures will be clear later, when the overall operation of the system is explained.

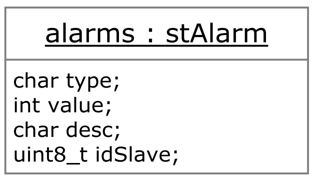


Figure 5.25 – Data structure used to handle the alarms

Last data structure developed is used to handle alarms. It stores the different parameters related to this variable.

Next section describes the use of these structures and presents the flowcharts which represent the system behaviour.

5.4.3 Behaviour of the Implemented Software

It is now possible to describe the behaviour of the wireless sensor platforms. This section presents different flowcharts in order to show how the required tasks are accomplished at application level. To begin with, the overall behaviour of a node is explained. The way how the message frames are handled and how the nodes reply is described.

5.4.3.1 Overall Node Behaviour

The overall behaviour of the wireless master node is described on the figure bellow. As one can see this unit waits for a message from the central slave that arrives through USART.

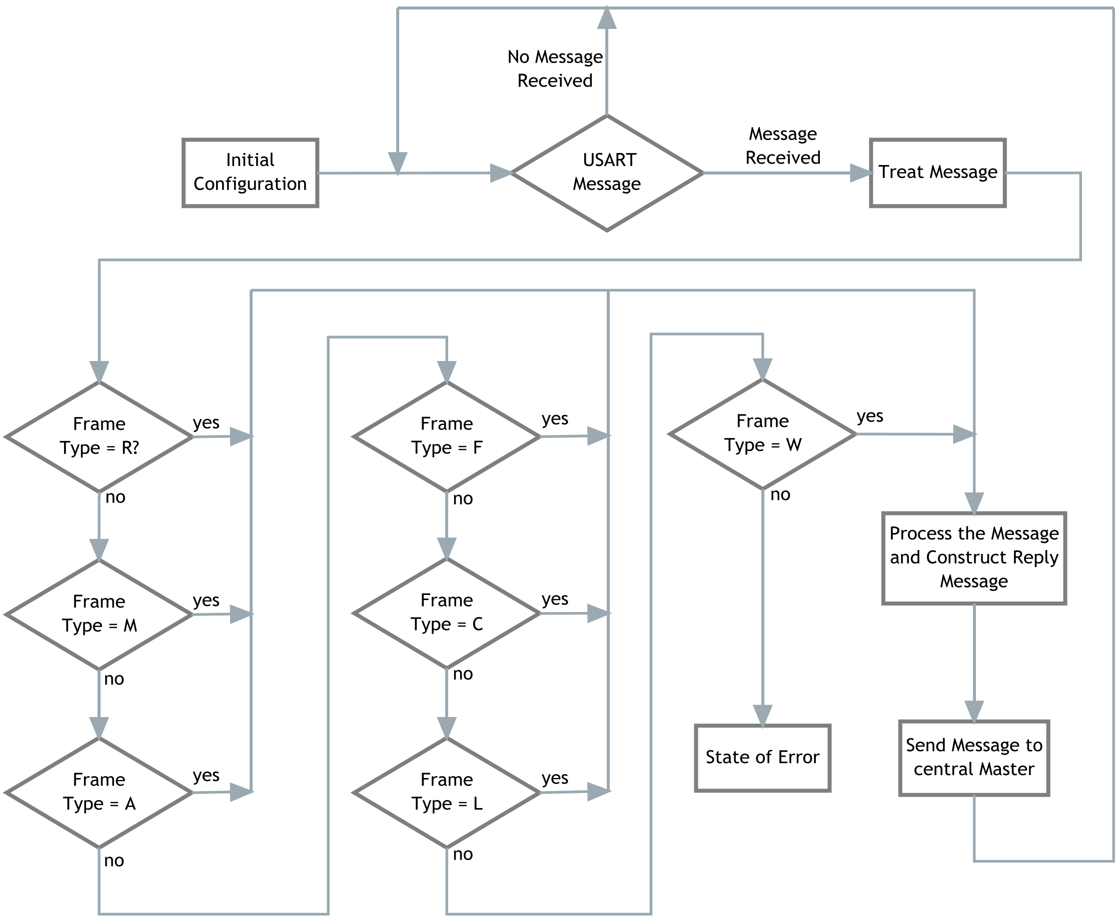


Figure 5.26 - The overall behaviour of the wireless network masters

When a message is received it is treated. This block represents the process of splitting the frame and filling the *base_frame* data structure. With all the frame parameters organized inside the structure the application check what type of message was received. Afterwards the mote processes the orders received and reply to the master. The manner how the orders are accomplished is explained in detail on the next subsections.

Slave nodes exhibit the same behaviour but instead of waiting a message from the USART they receive it through the XBee module.

5.4.3.2 Proceedings for Frames 'F', 'R', 'M' and 'A'

These frames are relative to the end-nodes of the wireless network. Therefore the WPAN coordinator has only to route them to the respective node. As for the slave node each one of these frames represents a different task to accomplish. Next figure shows the behaviour of a slave upon receiving a message of one of these types.

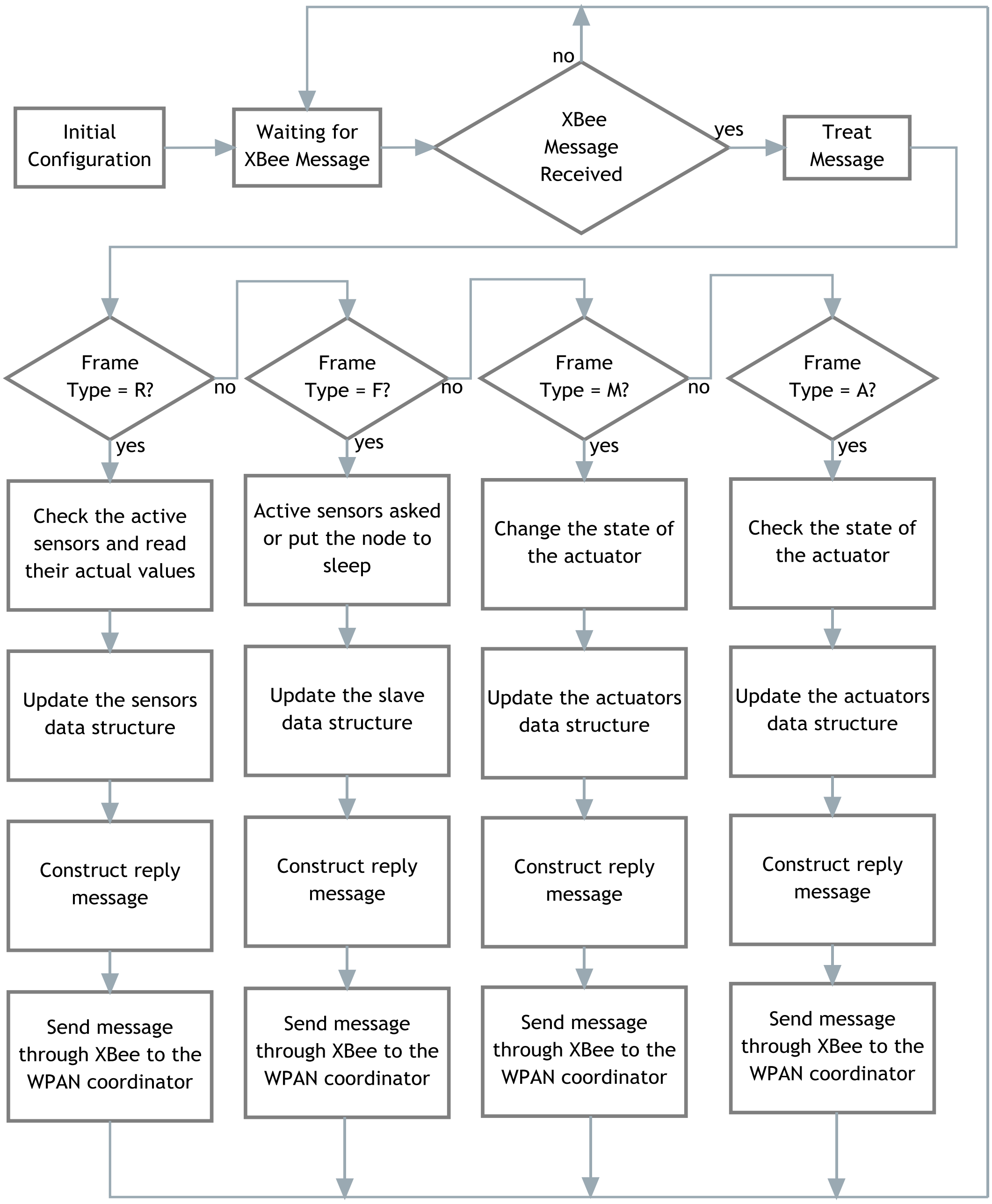


Figure 5.27 – Flowchart representative of the behaviour of a slave node after receiving an 'R', 'F', 'M' or 'A' frames

It should be noted that the initial configuration process of a slave node is relative to the creation of all the data structures already described. The structures for XBee packets treatment are initialized and the XBee radio connected. Finally the analogue-to-digital converter and RTC of Waspmote are configured.

When the WPAN coordinator receives the message from the slave with new data it updates its data structures and send the frame back to the central master through the USART1.

5.4.3.3 Proceedings for Frames of 'C' Type

As it was said before this type of frames is sent in broadcast to all the devices in the system. Masters from the wireless networks receive the frame from the central master and synchronize their RTCs. After they send the message to all their slaves and wait for the reply. When all the nodes on the network are synchronized the WPAN coordinates reports to the central master. The following diagram explains the behaviour of a wireless network master.

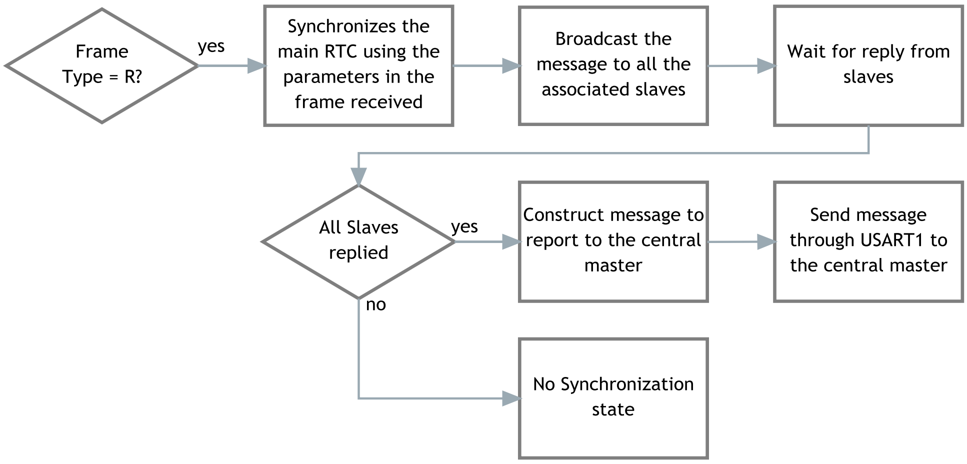


Figure 5.28 – Wireless network master behaviour when receives a synchronism frame

As it is possible see from Figure 5.28, the communications delays are not accounted on the synchronism process. This can be a major problem for systems using damage detection algorithms based on vibration analysis [46]. Also the Waspmote RTC, although very precise, exhibit resolution on the seconds. This means that even if a better synchronism algorithm was applied, the RTCs would be at the maximum synchronized to the second.

The development of an accurate synchronism method using the crystal oscillators together with the RTCs, in order increase the clock resolution, was left out to a second iteration.

5.4.3.4 Proceedings for Alarm frames

The treatment of alarms is a bit atypical given the rest of the system. In this case the central master is not the only device with permission to initialize this task. The WPAN coordinators are responsible to request alarm reports from theirs slave periodically. On the implemented system this period is set to 500 ms. If any alarm is detected, the WPAN coordinator stores it and when the central master request the alarms from that network it send them in a FIFO manner. Next flowchart represents the request of alarms by a WPAN coordinator to its slaves.

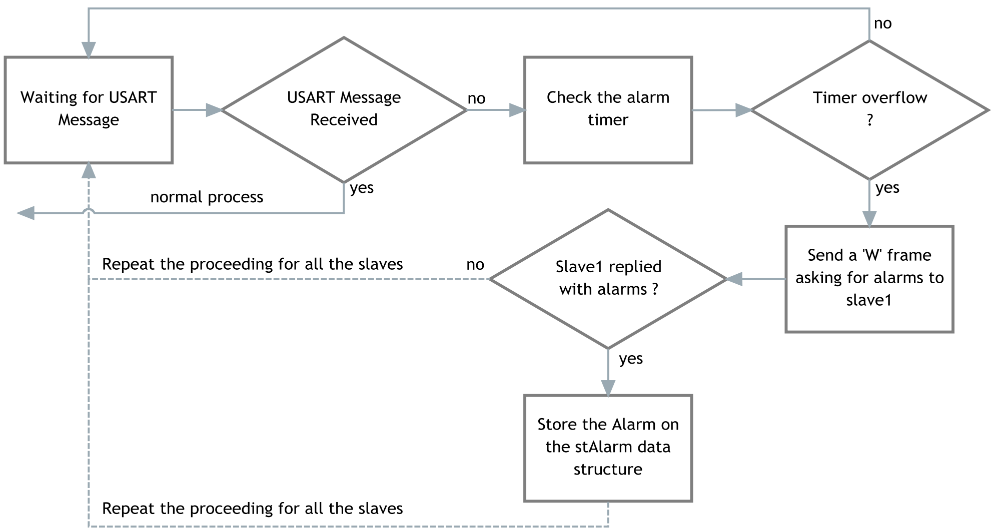


Figure 5.29 – Flowchart described the process of requesting alarms to end-nodes by WPAN coordinator initiative

As it is shown in Figure 5.29 if no USART message is received and the 500 ms timer overflow the wireless network master asks its slaves for alarms. The request frames are sent in unicast for each slave and if alarms are reported they are stored in chronological order.

In order to respect the master-slave architecture WPAN coordinators cannot initiate a communication with the central master. Therefore the central master also sends periodic messages requesting alarms to the WPAN coordinators. The active alarms are reported one by one with from the oldest to the newest.

The end-nodes behaviour is much simpler. They receive message frames of 'W' type and check all their sensors in order to determine if there is any alarm situation. They replying using the protocol already described in the section 100

5.4.3.5 Proceedings for 'L' Type Frames

As stated before the treatment of 'L' frames is probably the most complex of the entire system. It was developed in order to record measures for a period of time with higher sampling rates. Therefore this frame is only used to record accelerations.

When the central master sends an 'L' frame with index 1 it expects to receive a message with the number of frames that the task requires and the number of measures that last frame will have. These parameters are calculated by the wireless network master and sent back to central master. This proceeding is done in order to allow the central master to prepare for receiving a large quantity of data. Afterwards it sends an acknowledge message saying that it is ready to receive all the measures.

Now the wireless network master can ask the slave to start sending the measures. The sensor is read at 100 Hz and the slave sends these values without any header in order save time. After the master has received all the readings it puts them in the number of frames accorded before. A frame can have a maximum of 20 measures.

After receiving all the measures the central master sends an 'L' frame with the index 3 which means that all the data was received successfully. During this time neither the central master nor the network can process other types of frames.

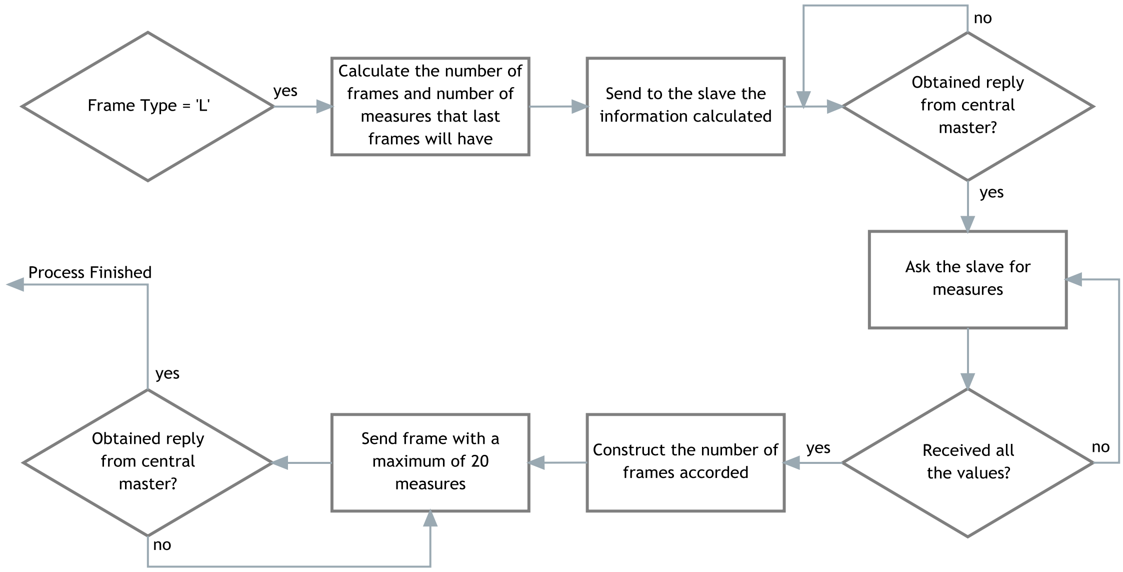


Figure 5.30 – Treatment of an 'L' type frame by the WPAN master

Once again in this process the slave will be only responsible to read the sensor and send the data back to the WPAN coordinator through the XBee module.

All the communications between the WPAN coordinators and the central master are done using the CAN bus. Therefore the CAN gateways have to be prepared for all these types of frames. Following it is presented the implementation of the protocol defined on these devices.

5.5 CAN Bus Implementation

The CAN bus is controlled by the AT90CAN64 from Atmel. This microcontroller supports both the physical and data-link layers of the CAN 2.0 protocol. It allows a maximum bandwidth of 1 Mbits/s. All the types of frames defined in CAN specifications and already described in chapter 4 can be managed by this controller.

In order to treat and construct CAN frames this unit has 15 structures called Message Objects (MOb). These structures can be compared to the data structures created for the wireless protocol, in particular the *baseMessage,* however with additional features. They contain all the information to handle a CAN frame and are seen as object. The AT90CAN64 structure is shown on Figure 5.31.

Before sending or receiving a message the MObs have to be initialize according to the task. They may be initialized to work as transmitters, receivers, receivers with auto reply or multi frame receivers. The most important fields that need to be initialized are the message id, type of message, number of data, and the message mask. This last specification is used to filter messages that are not wanted. If no mask is applied the controller receive all the messages in the CAN bus.

It is important to remind that CAN frames support a maximum of 8 data bytes. Therefore the protocol specified for the wireless network has to suffer some changes in order to be used in CAN. All the adapting process of one protocol to another is done on the gateways developed.

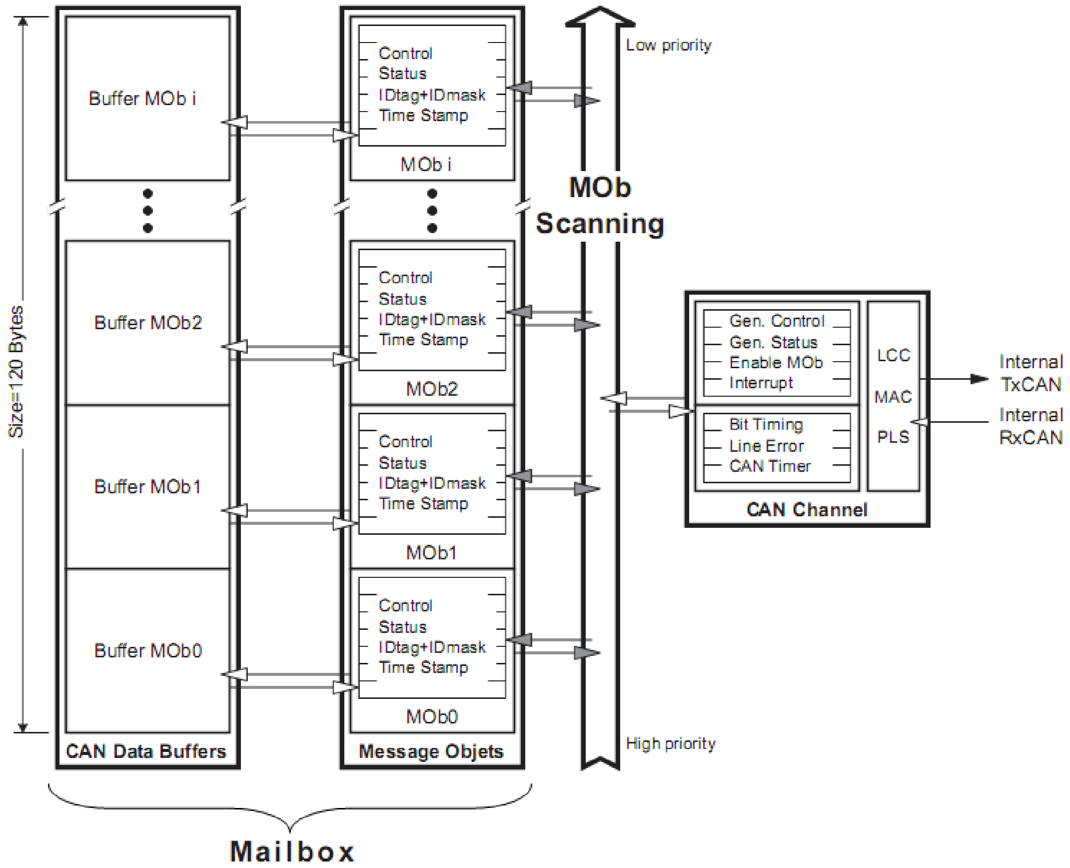


Figure 5.31 – CAN controller structure [62]

5.5.1 Implementation of Developed Frames on CAN

One important characteristic of the protocol implemented in CAN is the format in which the data values are transmitted. In order to save space all the measures are sent in integers instead of chars like on Waspmote. Therefore the value “123” can be sent in one byte instead of three. Other differences are the exclusion of WPAN coordinator ID and CRC field. CAN protocol implements CRC by itself therefore there is no need in repeating this process. Actually CRC is only applied on the application level in communications between the AT90CAN64 and the Waspmote. This happen because USART protocol, used to connect the two devices, does not support one. WPAN coordinator ID is also not used in CAN because each slave has a unique ID for the entire system. This means that it is possible to route the message using only the end-node identifier.

Next the developed CAN messages are described.

5.5.1.1 Type ‘F’ Frame on CAN

In order to transmit a type ‘F’ command three CAN messages have to be sent (Figure 5.32). This happens due to the need of 24 bytes which represents three CAN messages. As it is possible to see the WPAN coordinator ID is not present and there are two bytes to define the type of function. *FunctionP* field indicates if it is a continuous, interval or sleeping function and *FunctionS* determines if it applied to all the sensors or just to some.

The rest of the fields are exactly the same of the original frame however it should be notice that the year is sent in two bytes.

	Bytes [0:7]							
	0	1	2	3	4	5	6	7
F.1	IdSlave	IdMsg	Ftype	FunctionP	FunctionS	Year_I1	Year_I2	Month_I
F.2	Day_I	Year_E1	Year_E2	Month_E	Day_E	Hour_I	Minute_I	Sec_I
F.3	Hour_E	Minute_E	Sec_E	N_Sensors	IDSensor1	IDSensor2	IDSensor3	IDSensor4

Figure 5.32 – Messages sent to transmit a type 'F' frame on CAN

5.5.1.2 Type 'R' Frame on CAN

On this case only one message is needed to effectuate the request. The reply, however, needs two messages. All the data fields are the same as the ones defined on the original protocol. The measures values are transmitted in two byte because the ADC of Waspmote has a resolution of 10bits. Next image shows the messages used to process an 'R' task.

	Bytes [0:7]							
	0	1	2	3	4	5	6	7
Request - R	IdSlave	IdMsg	Ftype	x	x	x	x	x

Reply - R.1	IdSlave	IdMsg	Ftype	N_Sensors	IDSensor1	V_1_1	V_1_2	IDSensor2
Reply - R.2	V_2_1	V_2_2	IDSensor3	V_3_1	V_3_2	IDSensor4	V_4_1	V_4_2

Figure 5.33 – Messages used to transmit a 'R' order

5.5.1.3 Frame Types 'A' and 'M' Implemented on CAN

It is only needed one message to send this kind of orders. The message structure is the same of the original protocol with the exception of the WPAN coordinator ID, which is not present, like it was said in the beginning of the section.

Bytes [0:7]							
0	1	2	3	4	5	6	7
IdSlave	IdMsg	Ftype	IdActuator	State_A	x	x	x

Figure 5.34 – Frame to read and change actuators state

5.5.1.4 Type 'C' Frame on CAN

Like for the 'F' type frames the year in this case is sent on two bytes. Other than that the structure is the same of the original protocol divided in two messages (Figure 5.35).

	Bytes [0:7]							
	0	1	2	3	4	5	6	7
C.1	IdSlave	IdMsg	Ftype	Year_1	Year_2	Month	Day	Weekday
C.2	Hour	Minutes	Seconds	x	x	x	x	x

Figure 5.35 – Synchronizing frame on CAN bus

5.5.1.5 Type 'W' Frame on CAN

Once again only one message is needed to implement the transmission of this type of tasks on CAN bus. The alarm value is divided in two fields because like it was said before the measures made by Waspmote have 10 bits. Next image shows the structure of the frame.

Bytes [0:7]							
0	1	2	3	4	5	6	7
IdSlave	IdMsg	Ftype	AlarmType	AlarmDescription	AlarmValue_1	AlarmValue_2	x

Figure 5.36 – Alarm frame implemented for CAN

5.5.1.6 Type 'L' Frame on CAN

The last type of frames is divided in two different kinds of messages. The control messages that order the WPAN coordinator to start the measuring process and manage the exchange of messages are equal to the ones already presented for wireless sensor nodes (Figure 5.37).

Bytes [0:7]							
0	1	2	3	4	5	6	7
IdSlave	IdMsg	Ftype	IDSensor	MsgIndex	MsgNumber	MeasureNb	x

Figure 5.37 – Control frames for the type 'L' task on CAN

The messages responsible to transmit the measured variables use two bytes per value. Therefore for a maximum of values sent in a wireless network message of 20, CAN has to use 5 frames.

5.5.2 Configuration of Gateways

As it was said before there are several registers that need to be configures in the AT90CAN64 to initialize communications through a CAN bus. The data rate used is one of the most important. On this system it is set 100 kbps in order to use cables with lengths up to 500 meters.

Also MOBs have to be configured. The architecture presented by the CAN controller used, supports only 15 MOBs. However, it is possible to initialize many more. The idea is to store MOBs configurations on RAM and when a transmission has to be done associate these virtual MOBs to the real ones. This way it is possible to previously define MOBs for all type of frames that can be sent through the CAN bus and use them only when they are required. For example four virtual MOBs are initialized to treat 'W' type frames. They actually have all the same structure; however, their identifiers are different depending on the node with which the unit intends to communicate. Also MOBs have different configurations if they are to transmit or receive frames. Therefore in the case of 'W' type orders, there are two MOBs two transmit to each WPAN coordinators and two others to receive from each WPAN coordinator.

In order to create and configure all these variables Atmel developed a programming library in C language. Although not user friendly this library allowed a faster implementation of the specified protocol on the AT90CAN64 microcontrollers.

The gateways developed have to connect with the central master (PC) and WPAN coordinators (Waspmote) through USART. Therefore the AT90CAN64 USART ports had to be configured to be in accordance with the other devices. It was used a data rate of 115200 bits/s with no parity and a stop bit. This data rate was chosen in order to not limit the CAN bus bandwidth.

5.6 Conclusion

This chapter described most of the implementation process. It started with the hardware developed to support all the communications and sensors. Where the most important components used are discussed. Afterwards the protocol designed to allow different units to understand each other was presented. Its main features were also discussed. Finally the implementation of this protocol both on the wireless nodes and CAN gateways was explained.

At this point of the document the basic mode of operation should be clear. The user interacts with the system through the human-machine interface implemented on the central master. This unit transmits orders to the different WPAN coordinators which are responsible to process the requests. Most of the tasks are relative to end-node devices, as so, each WPAN coordinator communicates with its slaves wirelessly in order to gather the information needed and then replies to the central master.

The HMI and associated modules are out of the scope of this dissertation. However, they were developed to integrate the system by the fellow worker Pedro Moreira.

Next chapter shows the tests related to the validation of the prototype here described.

Chapter 6

Validation Tests

In order to validate some of the technologies used some tests were executed in the LABEST laboratory. However, the environment conditions are not the same for a real civil infrastructure, these tests allows an analysis of the system performance.

This chapter describes the experimental proceeding and some of the obtained results. In addition some results are discussed in order to understand what should be improved on the next iteration.

6.1 Overall Communications Performance

Several tests were made in order to understand the communications performance. It was already explained that static tests on civil structures do not require an extremely fast system. However, latencies higher than one second are not advisable. Also, in a posterior phase, the concept here proposed may be used for monitoring dynamic tests. This means a lot of measures being transmitted and if the communications are to slow they may not keep up with the data volume.

In order to determine the actual communication speed, all the tests are executed using the protocol developed. Actually what is being monitored is not the communication speed but rather the system speed. The central master was programmed to count the elapsed time from the moment it receives a command from the user to the moment when the requested task is finished. These tests were conducted using only two gateways and two wireless nodes. Figure 6.1 shows the architecture used. As one can see the message passes through the entire system, the task is performed and then an answer is replied back to the central master.

The timer used to count the elapsed time offers a resolution of milliseconds; therefore all the results are exhibited in this time unit. All types of message frames were tested excluding the synchronization task and alarms detection. As these frames are sent in broadcast they depend on the number of the nodes in the network. The results would not be coherent with the performance of the system when all devices are connected.

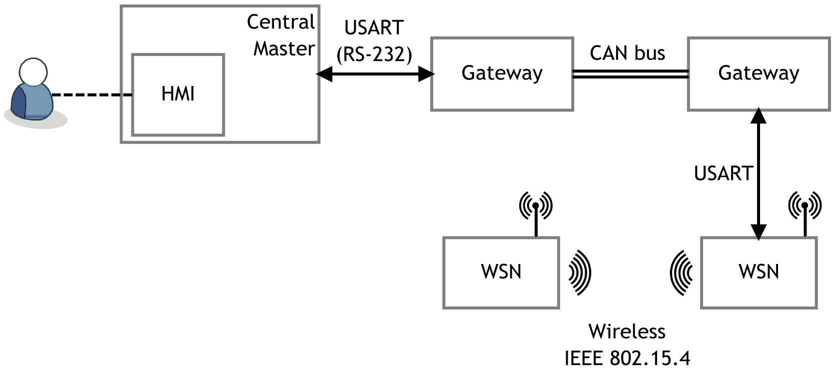


Figure 6.1 - Architecture used for communication tests

Next table shows tests made using the ´R´ type frame. Actually these tests allow the analysis of the highest sampling rate of the system using this process.

Test Number	Elapsed Time (ms)
Test 1	313
Test 2	343
Test 3	328
Test 4	344
Test 5	351
Average	335.8

Table 6.1 – Time that the system requires to accomplish a normal reading task

The configuration of the ADC was set to max speed at free running mode, therefore its sampling frequency was much higher than the communications delay. As it was expected the communications are restraining the speed of the system. However these results are still good enough to monitor static tests.

Next table shows tests using the type ‘A’ frame. It is only used to check the state of an actuator.

Test Number	Elapsed Time (ms)
Test 1	328
Test 2	332
Test 3	330
Test 4	343
Test 5	344
Average	335.4

Table 6.2 - Time that the system requires read the state of an actuator

The results are similar to the ones obtained on the previous test. A large drift from these values was not expected however it is strange that the Waspmote is faster reading an ADC than checking the state of a pin. One possible justification is related to the implementation. For example after receiving a message, Waspmote checks its type by comparing the *FrameType* field with expected values. It starts by checking if it is not a 'W' type frame, then if it is not an 'R' type frame, and the last value it tests is the 'A' type. Thus, the process of checking the frame type takes more time for these frames then for 'R' frames. Other type of implementation issues can be involved and the system should be analysed from end to end in order to understand what led to this result.

The following table shows the obtained results for an 'M' type message. This frame is processed like the one before, however, now the state of the pin is forced to a value.

Test Number	Elapsed Time (ms)
Test 1	375
Test 2	343
Test 3	375
Test 4	344
Test 5	348
Average	357

Table 6.3 - Time that the system requires to force the state of an actuator

The measured times are a little bit superior to the ones showed on Table 6.2 which was the expected. Next test is relative to 'F' type frames. It was used the function read in a period of time determine sensors. This is one of the biggest frames that can be transmitted (Figure 5.32).

Test Number	Elapsed Time (ms)
Test 1	1094
Test 2	1094
Test 3	1109
Test 4	1094
Test 5	1125
Average	1103.2

Table 6.4 – Time required for processing the longest 'F' type frame

As it was expected the times increased quite a lot. This is mainly because CAN bus has to send this frame in three messages, which increases the communication time but also the message treatment.

Last frame tested is from the 'L' type and is used to request a number of measures from a sensor. In this case it was asked for 200 measures which means 10 frames on the wireless network implementation and 50 on the CAN network. It should also be reminded that in this process several messages are traded between the WPAN coordinator and the central master. Next table shows the results obtained.

Test Number	Elapsed Time (s)
Test 1	21,907
Test 2	22,094
Test 3	21,750
Test 4	22,437
Test 5	21,875
Average	22,012.6

Table 6.5 – Time required for processing an 'L' type frame

The results shown are not very promising. However, remember that these times do not affect the sampling rate, actually all the measures are taken and only then are packed in frames and sent to the central master. It is possible to say with some degree of certainty that these values are caused by the 50 frames that have to be constructed, sent and treated by the CAN network.

6.2 Communication Tests on Different Networks

In order to analyse the communications in more details, all networks were testes one by one. The first test made is relative to the communication between the Waspmote and the gateway connected to it. A frame is sent repeatedly from the gateway to the Waspmote with 100 ms intervals. After transmitting the AT90CAN64 sets a pin high for 1 ms. On its side the Waspmote set a pin high when receives the message. This behaviour is monitored using an oscilloscope. The idea is to understand the time that a message takes to be transmitted by USART. Next image shows the results obtained.

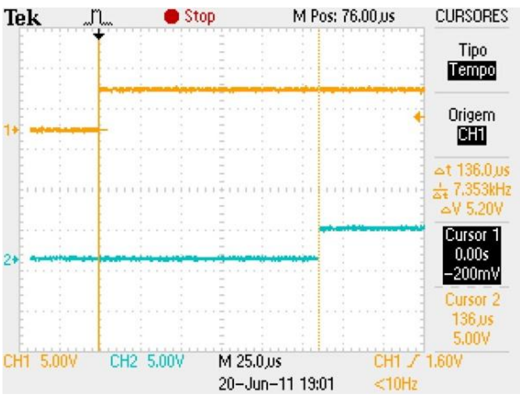


Figure 6.2 – USART communication delay when 1 byte is transmitted

As last image shows the first byte is received by the Waspmote 136 µs after it was sent by the AT90CAN64. It should be minded that USART protocol adds 1 start bit and 1 stop bit per byte transmitted. The frame sent was "~1#1#8#M#2#2%204|\r\n", which has 19 bytes. Adding two more bits per byte it gives 190 bits that need to be transmitted. The results obtained are shown on the figure bellow.

Figure 6.3 – USART communication delay when 19 bytes are transmitted

Last figure shows a delay of 2.68 ms which is near the expected. The theoretical value is also not far from the results obtained. The USART is set to work at 115200 kbps which means that the entire frame should have been transmitted in 1.64 ms. If considered the processing that the microcontroller has to do in order to convert the transmitted signal to characters and store them in a variable, the results obtained seems coherent.

The same test was made between two CAN gateways. After transmitting a message the AT90CAN64 sets a pin high for 1 ms. When the other gateway completes the reception of the frame it also sets a pin high for 1 ms. The obtained results are shown on the next figure.

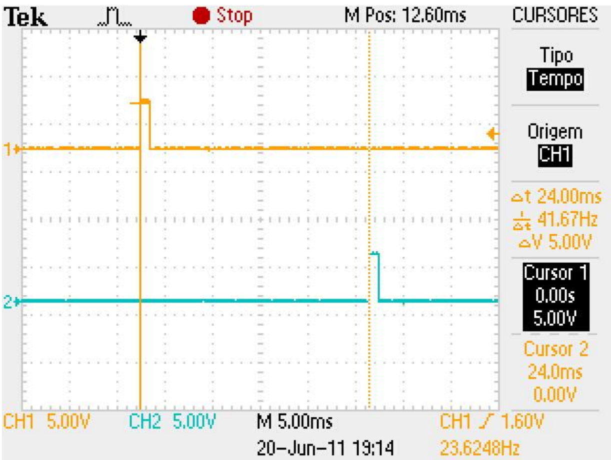


Figure 6.4 – Delay in communications CAN between two gateways

The results are once again behind the expectations with a delay of 24 ms. The same frame was used and adding the overhead uses by CAN protocol it giver 111 bits needed to be transmitted. The results for a bus with a bandwidth of 100kbit/s should have been near 1,11ms. One possible justification for these results is related to frames processing. The CAN controller presents a quite complex routine to filter and extract the data from the frames received. This process allied with the allocation of MOBs to transmit, can influence the network performance.

Last test is related to the wireless network based on IEEE 802.15.4. Its theoretical bandwidth is 250 kbit/s and the same frame is once again used. As it was study before the XBee module adds some overhead to the message. Also the API used uses a special header which occupies more 24 Bytes. The message transmit has therefore, 50 Bytes.

The experimental proceeding is the same already explain, on next image the yellow line represents the behaviour of the transmitter and the blue the receiver.

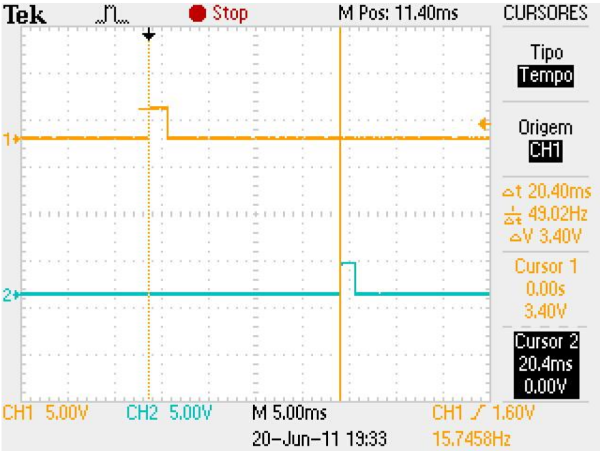


Figure 6.5 – Delay on the wireless network with one Waspmote transmitting to another

A delay of 20.4 ms is obtained, like Figure 6.5 shows. This is a bit more than the expected which was 1.6 ms. However the Waspmote mote API has a rather complex process to treat the frames received in order to unpack them and save them in a data structure. This process may add some more delay.

6.3 Maximum Wireless Communication Distance

In order to determine the maximum communication range between two Waspmotes, they were tested on line of sight. The result obtained is shown on the Figure 6.6.

Figure 6.6 – A maximum distance of 160 meters was achieved on tests inside the University Campus

This test was made using the maximum output power that XBee IEEE 802.15.4 RF module allows (1 mW). Nevertheless it was only obtained a maximum range of 160m when the theoretical value is 500m. One reason that could explain this result is the fact that all the University Campus is covered with a wireless network working at the same frequency.

For a future development the modules use can be exchanged for the XBee PRO, which allows an output power of 63 mW. It should be reminded that increasing the communication power reduces the battery time.

6.4 Tests with Sensors

In order to understand how the complete system was working some tests with a metallic plate were made. This plate has four resistive strain gauges and an accelerometer applied to it. The strain gauges used are the CEA-06-250UW 350 from VISHAY and the accelerometer is the ADXL203 from Analog Devices. All these sensors were already discussed in chapter 4.

The metallic plate has a length of 74 cm and has one of its extremities fixed to a still support. The distance between the fixed point to the actual end of the plate is 2,5 cm. Its section is given by a width of 8 cm and 0,8 cm of height. Also it presents a section reduction at one third of its length (Figure 6.7).

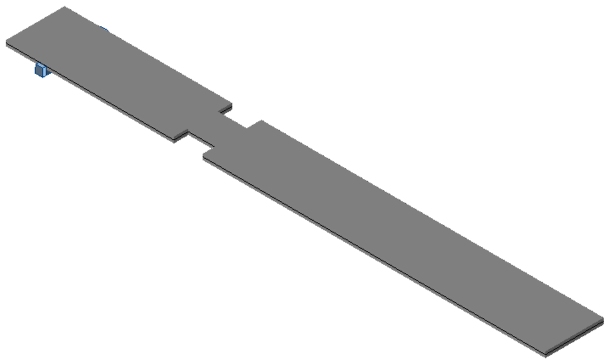


Figure 6.7 – Mathematical model of the metallic plate used

Last figure shows the model created in *Autodesk Robot Structural Analysis* by the LABEST engineer Augusto Faria [63]. This model is used to determine the theoretical behaviour of the metallic plate and compare the results to the ones obtained in reality.

Next figure shows a schematic of the metallic plate used with the location of some sensors. As one can see it has four strain gauges indicated by the initials *ERE* and one accelerometer marked as *AP*. The nomenclature defines also the section where sensors are installed, A or B, and if they are installed on the superior or inferior side, 1 or 2.

One of the tests realized was to fix a weight of 10 Kg in the loose extremity of the plate. This weigh deforms the plate which leads to stretching of the superior fibers and contraction of the inferior fibers. This test was executed in order to read the strain gauges and compare the actual values with the ones calculated.

The accelerometer is also put under test. To achieve a good plate excitation, the weight is cut off instantaneously. After the cut the plate starts oscillating and the monitoring system measures the acceleration that the plate is subjected to. Using mathematical analysis is then possible to estimate the natural frequency of the plate and compare the results with the ones determined by the mathematical model.

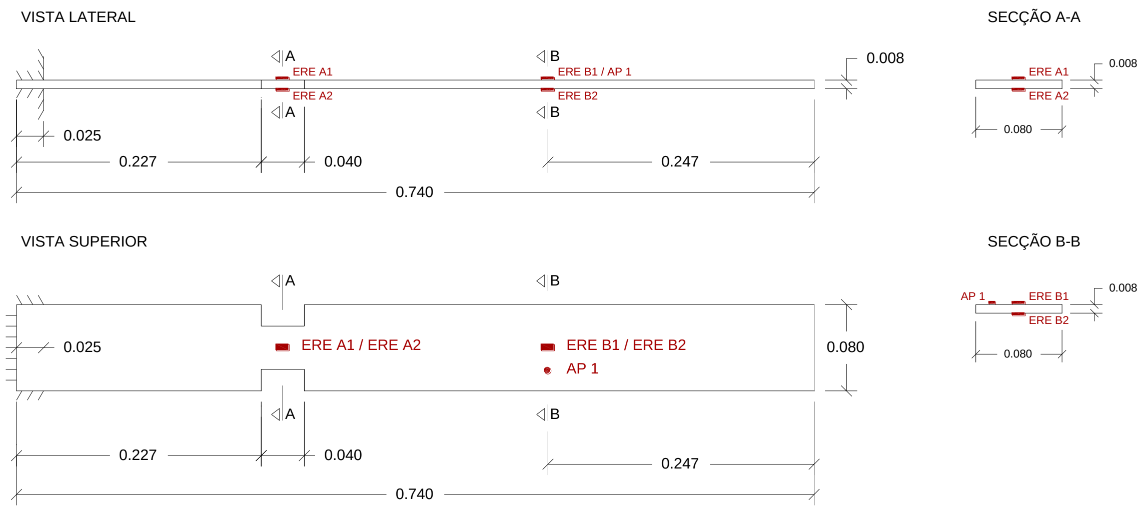


Figure 6.8 – Schematic of the metallic plate with the location of some sensors

As the developed system integrates a lot of sub systems the source of errors is wide. In order to narrow it down some sub systems were tested independently. The next test presented has the objective of validate the signal conditioning circuits.

Using an oscilloscope the output of these circuits was monitored. The plate excitation was done manually, which means that the results here shown does not give any absolute values about the state of the plate. However, it is possible to assess about the behaviour of the circuits developed and if they are working correctly.

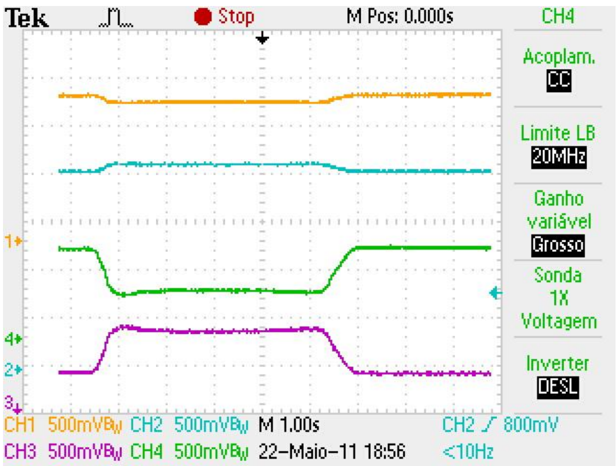


Figure 6.9 - Output signal from the strain gauge signal condition board

Figure 6.9 shows the output signal from the signal condition board developed for strain gauges. The channel 1 in yellow is relative to the *ERE B2* sensor. As it is possible to see, when a force is applied to the loose end of the plate the output value decreases representing a compression of the inferior fibres. The same type of variation is observed for the *ERE B1* sensor on the channel 2 (blue). However in this case the voltage output increases which represents the stretching of the superior fibres.

The same phenomenon happens for the sensors mounted on the *A-A* section. On channel 3 (purple) is the output signal related to the *ERE A1* gauge. Channel 4 (green) has the readings from sensor *ERE A2*. Once again both these signal have the same amplitude variation but with opposite signs.

All the channels are adjusted to have a voltage reference of 1,65V and a gain of 1000. Thus, the bigger amplitude variation presented by the signals coming from sensors mounted on section A, indicates that more tension is applied to this part of the plate. This phenomenon also occurs on the mathematical model as Figure 6.10 shows

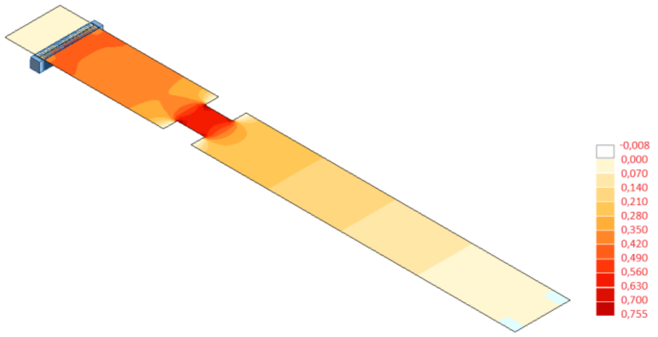


Figure 6.10 – Strain map of the superior side of the metallic plate. The colour is stronger where the deformation is bigger.

The same test was realized for the signal conditioning board developed for accelerometer measures. In this case, there is no theoretical information to assess about the waveform presented in Figure 6.11.

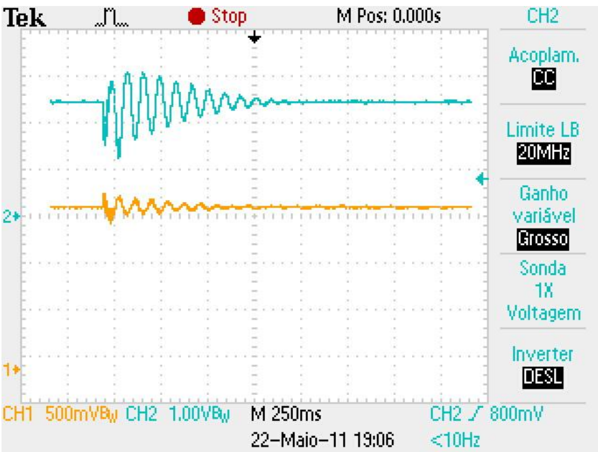


Figure 6.11 – Output signal from the signal conditioning board for the ADX203 accelerometer

The channel 2 (blue) shows the signal waveform from the output of the acceleration on the *Y* axis. This is the axis perpendicular to plate therefore where more acceleration should be felt. The result looks like the expected, however it is not possibly to compare them with anything. The simulation of the steel plate only infers about the oscillations frequency.

Channel 1 (yellow) represents the acceleration felts on the *X* axis. This is the axis parallel to the plate, so like expected the variation of the acceleration is almost zero. Like the signal condition board for the strain gauges, this one is working correctly.

6.5 Complete System Validation

Finally the complete system is assembled together. As it was previously described the steel plate is used to simulate the structure under study. For these tests a weight of 10 Kg is placed on the loose end of the plate and the deformation given by the various strain gauges is read. After the wire fixed the weight to the plate is cut and the resulting oscillation is measured used the accelerometer.

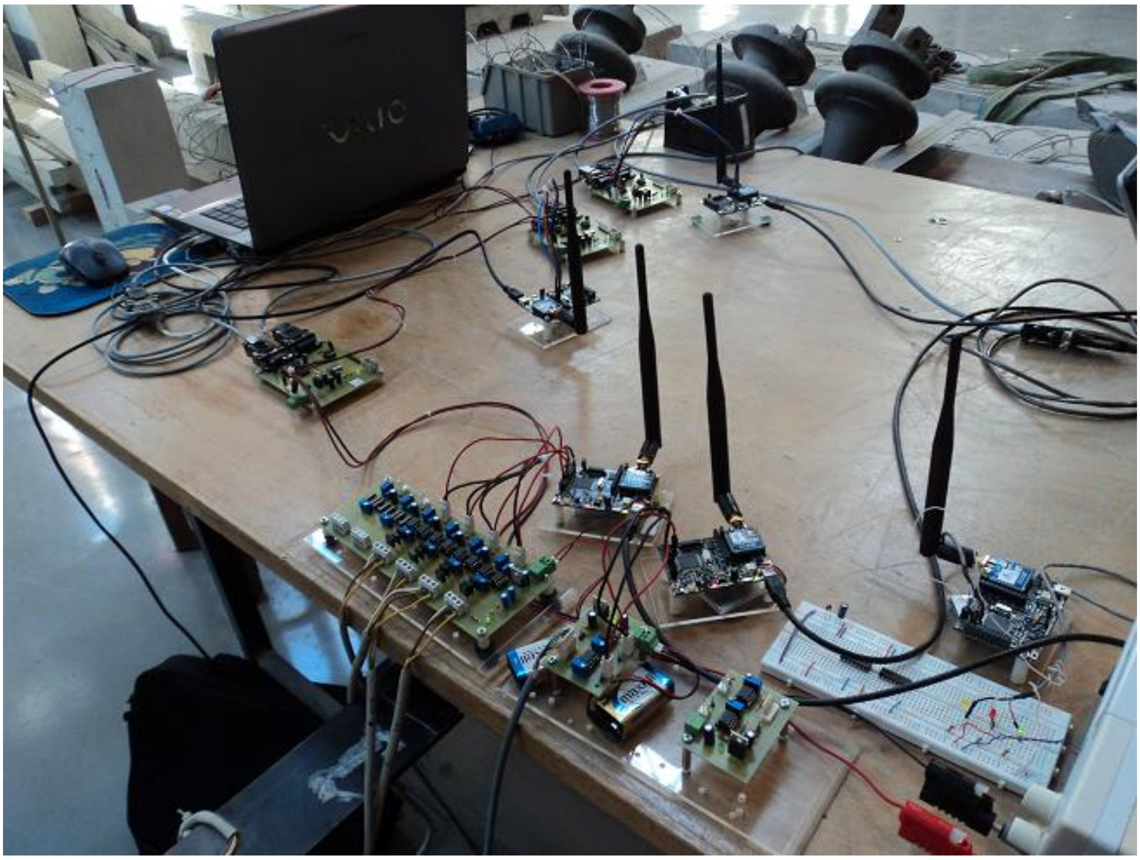


Figure 6.12 – Prototype developed on this project, assembled on the LABEST laboratory

Figure 6.12 shows the entire system ready to monitor the steel plate. On this figure the laptop is working as the Remote Control Post and is connected to World Wide Web. There are three gateways; one is connected to the laptop and other two to their respective WPAN coordinators. On the bottom there are the three end-devices connected to the signal condition circuits for strain gauges and accelerometers.

Using this setup various functionalities were tested. Prior to measuring the sensors, all the message frames were tested in order to check if they were working correctly. Even actuators were simulated by three LEDs of different colours that represent a semaphore. Although some bugs were detected the behaviour of the system can be considered good.

At this point it was decided to start acquiring the data from sensor. The system designed is connected to four strain gauges, one accelerometer and temperature sensors. Remember that the temperature sensors used are ICs integrated on the Waspmote; therefore, they are not visible on the setup shown. Next figure shows the steel plate with the sensors applied.

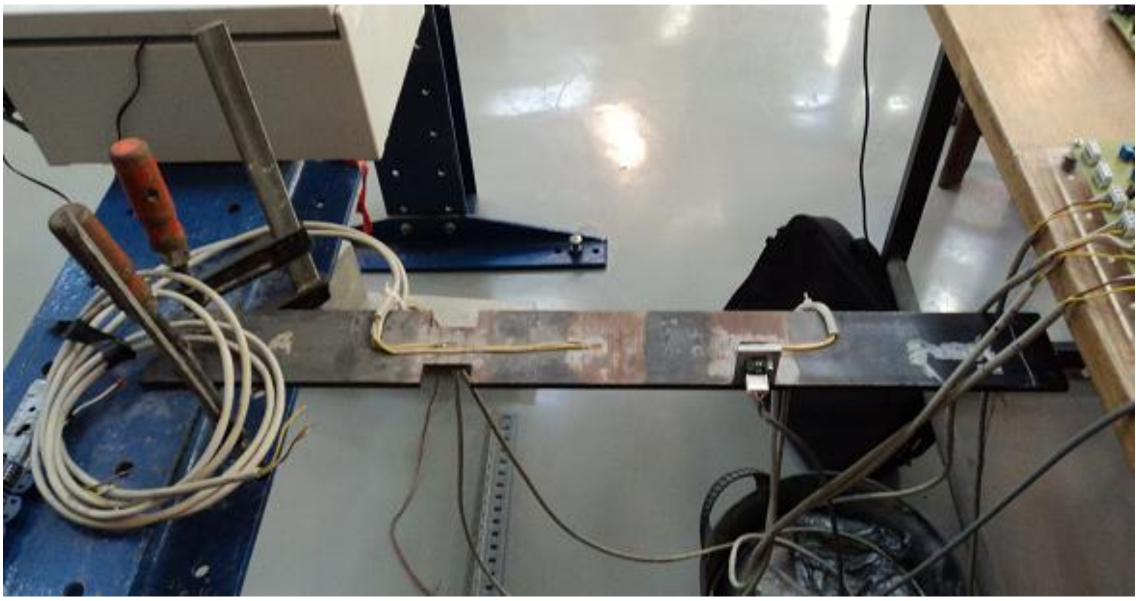


Figure 6.13 – Steel plate with the associated sensors

After all the sensors are connected to the respective end-nodes, the steel plate was deformed and data relative to the sensors was acquired. Next table presents the data gathered from tests with a weigh of 10 Kg applied to the loose end of the steel plate.

ERE B1		*ERE A1*		*ERE B2*		*ERE A2*	
µStrain	**R_{sg}**	**µStrain**	**R_{sg}**	**µStrain**	**R_{sg}**	**µStrain**	**R_{sg}**
18,11	350,013	296,67	350,218	-79,59	349,942	-144,86	349,894
20,02	350,015	294,76	350,217	-69,11	349,949	-143,91	349,894
18,59	250,014	297,15	350,218	-60,51	349,956	-142,96	349,895
18,59	350,014	296,67	350,218	-68,63	349,950	-143,43	349,895
13,35	350,010	289,99	350,213	-66,04	349,951	-143,43	349,895

Table 6.6 – Strain gauge readings acquired by the prototype developed

As it is possible to see the results presented are a bit incoherent. This is mainly due to the erroneous calibration of the signal condition circuit and the lack of resolution of the ADC. Also, the excitation of the steel plate was not made in the best conditions. The wire fixed to the plate, where the weight was suspended, was changed from position several times. Table 6.6 shows strains of 300 µε, which cannot be correct. Even with all the errors already descried these values are still too high. The reason found for this phenomenon what the lack of zero calibration. After the strain gauge is applied to the steel plate its value is not exactly 350 Ω. Therefore, a first measure should be taken in order to be used as reference for the posterior readings. This initial value should be read without any force applied to the steel plate. On a next iteration the zero calibration is mandatory in order to get relevant information about the integrity of the structure.

Yet the results obtained can be used to validate the system concept. It is even possible to see that strain gauges mounted near the point of fixation of the steel plate measure a higher variation of deformation than the others. These values were acquired by and end-node and stored on a web server, validating all the communications.

The ADXL203 mounted was also tested. As it was described before the wire holding the weigh is cut in order to excite the steel plate. The acceleration is than measured using the type ‘L’ message frame. This frame indicates that the system should acquire 200 samples of acceleration measures. Next figures show the results obtained.

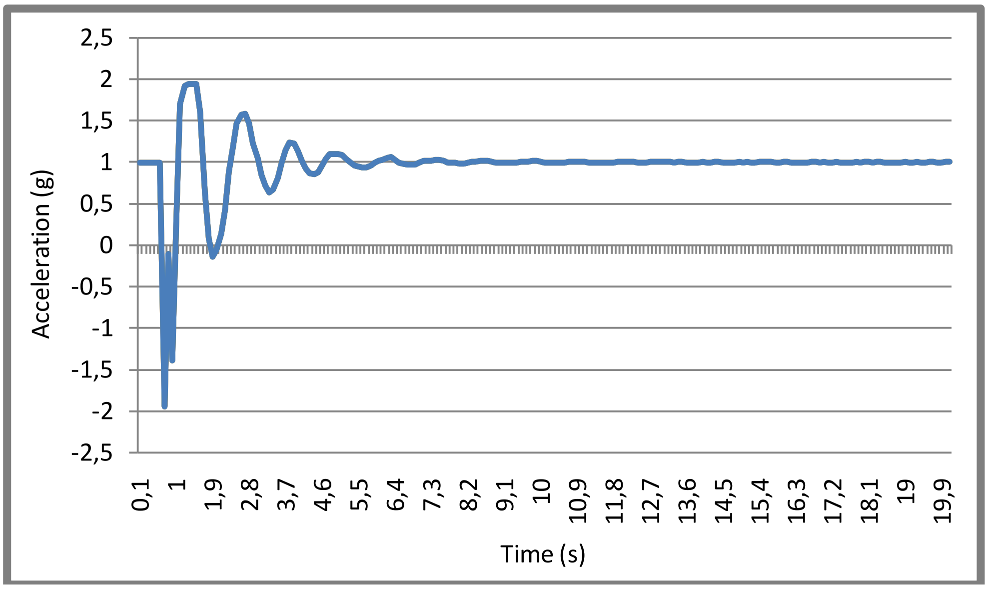


Figure 6.14 – Acceleration acquired by the prototype developed

As one can see the sampling frequency was not enough to capture the complete behaviour of the steel plate. The system was not able to work with sampling frequencies higher than 10 Hz, which clearly is not enough. Transmitting the readings as soon they are acquired is not possible if higher sampling frequencies are required. A new prototype is already been developed where all the data acquired is stored on the end-nodes. When all the measuring tasks are finished the central post request all the data stored. With this solution the sampling frequency will only depend on the ADC conversion time and the time spent to store the data on the SD card.

As it was stated before the ADXL203 is capable to measure acceleration on the range of ±1.7 g. On Figure 6.14 it is possible to see accelerations of almost 2 g. It is possible to see that the sensor is saturated near these values; hence it is not measuring the real values of acceleration. Depending on the structure where this system is to be installed, the accelerometer should be changed to one with higher measuring range.

6.6 Conclusion

The results here presented may not be much impressive. However, various solutions were presented to overcome the problems encountered. On a second iteration the signal condition circuits should be changed according to the suggestions given through the entire document. Also, sensors have to be chosen according to the infrastructure under study. The tests made on the steel plate only serve to validate the overall system concept.

Considering the operation mode of the system it is possible to say that for measuring dynamic quantities, where sampling frequencies up to 100 Hz are needed, the data should be stored on the end device where it is acquired.

Even with all the restraints the concept presented shows promising results when used for monitoring static tests. Next chapter the final conclusions of this project are presented and future work is suggested.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Throughout this document, the benefits of using systems based on electronics to monitor the state of civil infrastructures are extolled. One major concern related to the traditional systems used for monitoring structures is their cost. The long cables necessary to connect all the sensors to the acquisition module are expensive and require a lot of maintenance. Therefore, on this dissertation, wireless sensor networks are study in order to understand if they are suitable for structural health monitoring. On chapter 2 some solutions based on wireless networks are discussed with the purpose of understanding their main advantages and drawbacks. The main problem noted is the battery dependence that these systems exhibit. A trade off between performance and platform autonomy is always necessary.

Next chapter describes the proposed architecture. The concept shown in this section does not take into consideration technologies, just the functionalities that the system should support. During the process of defining the system architecture, a functional analysis of all the requirements was made. All these steps are described in chapter 3.

On this work a large amount of time was dedicated to the study of different technologies that could fit the proposed architecture. Chapter 4 presents a discussion of the most commonly used technologies and justify the choices made for this project. It is important to point that due to the project being in its initial phase some of these technologies may not be used on the final product.

The validation of wireless sensor networks for structural health monitoring is one of the main goals of this project. Therefore a prototype of the entire system was constructed and tested. Chapter 5 describes the implementation process and discusses some of the options made. As it was already noted this dissertation is relative to a first iteration of the process of developing a product. Therefore, there is still room to optimize the implementation of some of the used technologies.

Results from tests made in laboratory environment are presented in chapter 6. All the parts of the system were tested separately in order to give an idea of their potentialities. Although the performance exhibited is far from commercial devices for a first approach the results are promising.

Taking into consideration all the tests made it is possible to determine what are the main restrictions of the prototype developed are. The signal condition circuits present a lot of problems. Using homemade solutions when the precision is a critical factor is not the better approach. On a second phase these circuits should be updated as suggested on chapter 5 and chapter 6. Other problem spotted was the limitation of the maximum sampling frequency due to the communications delay. The use of CAN bus for sending large amounts of data revealed difficult. This is mainly because it only supports 8 data bytes per message.

Even with all these limitations the system showed capable of monitoring static tests made on civil infrastructures. Suggestions to overcome most of the problems encountered are given throughout the document. If this line of work is followed, next iteration should exhibit a better performance and can be even applied to monitor a real infrastructure. This way some other tests can be performed to better understand the behaviour of the system.

During this dissertation it was possible to analyse a lot of different technologies. The results prove that on a near future wireless sensor networks can be a solution to reduce the costs of structural health monitoring. Also the collaboration between two distinct areas of engineering revealed to be advantageous.

7.2 Future work

As it was mentioned throughout the entire document this thesis represents an initial approach to the use of wireless sensor networks for monitoring civil structures. Therefore, the development should not stop here.

First of all the implementation part should be revised in order to fully understand the potentialities of the technologies employed. For example the hardware developed for signal condition should be reconstructed following the tips given in section 5.1. These boards should also be tested in construction environment. The harsh ambient may cause the appearance of noise in the signal measures, which have to be eliminated by analogue and digital filters.

As it was demonstrated in section 5.5 CAN specifications restraint the amount of data that can be transmitted on a single message. In order to optimize this part of the system, the communication frames developed should be reviewed bearing in mind CAN limitations.

After all these modifications a new prototype should be assembly and the performance of the system analysed. This would represent the second iteration of the development process. It would be interesting as well to test the behaviour of the system for a real number of sensors. Only applying this concept to a real structure it is possible to validate its actual performance.

There is a possibility that in the future structural health monitoring will rely more in vibration analysis. If the system is to be employed for this kind of measures a robust synchronization algorithm should be developed. A small introduction concerning the problems of synchronism between motes is presented in chapter 2.

In order to finish this section it is necessary to refer the damage detection algorithms. On this dissertation it was assumed that an engineer would be responsible for the data treatment. However, in order to exploit all the benefits of distribute processing data should be analysed on the fly and damage algorithms applied. It has to be reminded that the requirements of the system should be changed according to the algorithms inputs.

Instead of using CAN-bus networks, solutions that employ powerful wireless networks should also be tested. These networks may not be transmitting the entire time being used only when data is needed to be sent to the remote control post. Also the antennas used can be optimized in order to achieve a more efficient system. On the future the entire system may communicate wirelessly, this would put an end to the large amount of cables needed to assembly these monitoring systems.

Lastly the use of renewable energy sources maybe integrated to supply the wireless sensor nodes. Waspmote already integrates this feature; however, further studies should be made in order to understand the efficiency of these power sources.

References

[1] H. Wenzel, *Health Monitoring of Bridges*: Jonh Wiley & Sons, Ltd., 2009.

[2] C. R. Farrar, "Historical Overview of Structural Health Monitoring " in *Lecture Notes on Structural Health Monitoring Using Statistical Pattern Recognition*, ed. Los Alamos, 2001.

[3] L. A. Bisby and M. B. Briglio, "An Introduction to Structural Health Monitoring," ISIS Canada and SAMCO Network of the European Commission2005.

[4] G. Park, *et al.*, "The fundamental axioms of structural health monitoring," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, pp. 1639-1664, 2007.

[5] D. Balageas, *et al.*, *Structural Health Monitoring*: IEST, 2006.

[6] B. Bakht, *et al.*, "Guidelines for Structural Health Monitoring," ed. Canada: ISIS Canada, 2001.

[7] M. F. Green, "Bridge Dynamics and Dynamic Amplification Factors - a review of analytical and experimental findings," *Canadian Journal of Civil Engineering*, pp. 876-877, 1993.

[8] C. E. Ventura, *et al.*, "Effective Use of Ambient Vibration Measurements for Modal Updating of a 48 Storey Building in Vancouver, Canada," Natural Sciences and Engineering Research Council, Vancouver2001.

[9] C. Sikorsky, "Development of a Health Monitoring System for Civil Structures Using a Level IV Non-Destructive Damage Evaluation Methold," in *Proceedings of the 2nd International Workshop on Structural Health Monitoring 2000*, Stanford University, 2000, pp. 66-81.

[10] L. Han, *et al.*, "Centralized remote structural monitoring and management of realtime data," in *International Symposium on Smart Structures and Materials*, San Diego, CA, 2004.

[11] E. H. Clayton, *et al.*, "Monitoring Infrastructural Health: In-situ Damage Detection and Localization Utilizing Distributed Smart Sensor Technology," *4th World Conference on Structural Control and Monitoring*, 2006.

[12] S. Cho, *et al.*, "Smart Wireless Sensor Technology for Structural Health Monitoring of Civil Structures," *Steel Structures Journal*, pp. 267-275, 2008.

[13] J. P. Lynch, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," *The Shock and Vibration Digest*, vol. 38, pp. 91-128, 2006.

[14] F. L. Lewis, "Wireless Sensor Networks," in *Smart Environments: Technologies, Protocols and Applications*, D. J. Cook and S. K. Das, Eds., ed. New York: Jonh Wiley, 2004.

[15] L. Mittag, "Magic in the air," *Embedded Systems Programming*, vol. 14, pp. 49-60, 2001.

[16] C. Stiernberg. (2011). *Five Things Every Scientist and Engineer Should Know about Wireless Sensor Measurements*. Available: http://www.laboratoryequipment.com/article-wireless-sensor-measurements-0903.aspx

[17] D. Gislason, *Zigbee Wireless Networking*: Newnes, 2008.

[18] T. Lennvall, *et al.*, "A comparison of WirelessHART and ZigBee for industrial applications," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, 2008, pp. 85-88.

[19] E. Sazonov, *et al.*, "Self-Powered Sensors for Monitoring of Highway Bridges," *IEEE Sensors Journal*, vol. 9, pp. 1422-1429, 2009.

[20] U. Battery. *What's the best battery?* Available: http://batteryuniversity.com/learn/article/whats_the_best_battery

[21] E. G. Straser and A. S. Kiremidjian, "A Modular, Wireless Damage Monitoring System for Structures," Department of Civil and Environmental Engineering Stanford University1998.

[22] J. P. Lynch, *et al.*, "A Wireless Modular Monitoring System for Civil Structures," in *20th International Modal Analysis Conference*, Los Angeles, CA, 2002.

[23] K. Mitchell, *et al.*, *Lessons learned about wireless technologies for data acquisition* vol. 4700. Bellingham, WA, INTERNATIONAL: Society of Photo-Optical Instrumentation Engineers, 2002.

[24] V. A. Kottapalli, *et al.*, "Two-tiered wireless sensor network architecture for structural health monitoring," San Diego, CA, USA, 2003, pp. 8-19.

[25] J. P. Lynch, *et al.*, "Field Validation of a Wireless Structural Monitoring Sytem on the Alamosa Canyon Bridge," presented at the SPIE's 10th Annual International Symposium on Smart Structures and Materials, San Diego, CA, 2003.

[26] S. Aoki, *et al.*, "Intelligent bridge maintenance system using MEMS and network technology," San Diego, CA, USA, 2003, pp. 37-42.

[27] L. Mastroleon, *et al.*, "Design of a new power-efficient wireless sensor system for structural health monitoring," San Diego, CA, USA, 2004, pp. 51-60.

[28] E. Sazonov, *et al.*, "Wireless intelligent sensor network for autonomous structural health monitoring," San Diego, CA, USA, 2004, pp. 305-314.

[29] D. W. Allen, "Software for Manipulating and Embedding Data Interrogation Algorithms into Integrated Systems," Master, Virginia Polytechnic Institute and State University, 2004.

[30] C. R. Farrar, *et al.*, "Sensor network paradigms for structural health monitoring," *Structural Control and Health Monitoring,* vol. 13, pp. 210-225, 2006.

[31] Y. Wang, *et al.*, "Wireless Structural Sensors Using Reliable Communication Protocols for Data Acquisiton and Interrogation," in *23rd International Modal Analysis Conference (IMAC XXIII)*, 2005.

[32] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*: Morgan Kaufmann, 2004.

[33] N. A. Tanner, *et al.*, *Structural health monitoring using modular wireless sensors* vol. 14. London, ROYAUME-UNI: Sage, 2003.

[34] S. D. Glaser, "Some real-world applications of wireless sensor nodes," 2004, pp. 344-355.

[35] B. F. Spencer Jr, *et al.*, "Smart sensing technology: Opportunities and challenges," *Structural Control and Health Monitoring,* vol. 11, pp. 349-368, 2004.

[36] Crossbow/MEMSIC, "TelosB - Datasheet," ed, 2005.

[37] Moteiv, "Tmote Sky - Datasheet," ed, 2006.

[38] L. Nachman, *et al.*, "IMOTE2: Serious computation at the edge," 2008, pp. 1118-1123.

[39] D. Gascón. (2010, 2011). *12Km ZigBee link with Waspmote*. Available: http://www.libelium.com/libeliumworld/articles/100550554427

[40] Libelium. (15-04-2011). *Waspmote*. Available: http://www.libelium.com/products/waspmote

[41] Crossbow/MEMSIC. Available: www.crossbow.com

[42] P. Levis, *et al.*, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*, W. Weber, *et al.*, Eds., ed: Springer-Verlag, 2005, pp. 115-148.

[43] J. A. Rice and B. F. J. Spencer, "Flexible Smart Sensor Framework for Autonomous Full-scale Structural Health Monitoring," NSEL, NSEL Report2009.

[44] S. Bhatti, *et al.*, "MANTIS OS: an embedded multithreaded operating system for wireless micro sensor platforms," *Mob. Netw. Appl.*, vol. 10, pp. 563-579, 2005.

[45] Q. Cao and T. Abdelzaher, "liteOS: a lightweight operating system for C++ software development in sensor networks," presented at the Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder, Colorado, USA, 2006.

[46] B. Sundararaman, *et al.*, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks,* vol. 3, pp. 281-323, 2005.

[47] E. Figueiredo, "Monitorização e Avaliação do Comportamento de Obras de Arte," Master Degree, Civil Engineering, FEUP, oPorto, 2006.

[48] T. Nagayama and B. F. J. Spencer, "Structural Health Monitoring Using Smart Sensors," NSEL, NSEL Report2007.

[49] J. S. Wilson, "Sensor Technology Handbook," ed: Elsevier.

[50] C. Félix, "Monitorização e Análise do Comportamento de Obras de Arte," PhD, Civil Engineering, FEUP, oPorto, 2004.

[51] W. M. Murray and W. R. Miller, *The Bonded Electrical Resistances Strain Gage*. New York: Oxford University Press, 1992.

[52] J. Fraden, "Handbook of Modern Sensors - Physics, Designs and Applications (3rd Edition)," ed: Springer - Verlag.

[53] P. G. VISHAY, "Strain Gage Selection: Criteria, Procedures, Recommendations," ed, 2010.

[54] H.-N. Li, *et al.*, "Recent applications of fiber optic sensors to health monitoring in civil engineering," *Engineering Structures,* vol. 26, pp. 1647-1657, 2004.

[55] D. C. Betz, *et al.*, "Structural monitoring using fiber-optic bragg grating sensors," *Structural Health Monitoring,* vol. 2, pp. 145-152, 2003.

[56] E. OMEGA, *Omega Temperature Measurement Handbook*, 7th Edition ed. vol. MMXIV, 2010.

[57] Atmel, "ATmega1281 Datasheet," ed.

[58] J. P. Lynch, "Overview of Wireless Sensors for Real-Time Health Monitoring of Civil Structures," *Proceedings of the 4th International Workshop on Structural Control and Monitoring,* 2004.

[59] C. Buratti, *et al.*, *Sensor Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity*: Springer, 2010.

[60] C. i. A. CIA. (10-03-2011). Available: http://www.can-cia.org

[61] G. Jaman and S. Hussain, "Structural Monitoring using Wireless Sensors and Controller Area Network," in *Communication Networks and Services Research, 2007. CNSR '07. Fifth Annual Conference on*, 2007, pp. 26-34.

[62] Atmel, "AT90CAN64 Datasheet," ed.

[63] A. Faria, "Aplicação do Sistema de Aquisição National Instruments Compact DAQ," LABEST2011.