Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

FINAL DEGREE PROJECT

TITLE: Wireless Sensor Network based on UAV for Weevils Surveillance in Saudí Arabia

DEGREE: Air Navigation Engineering

AUTHOR: Marc Donadiós Garriga

DIRECTOR: Oscar Casas Piedrafita

DATE: 30 de Juny del 2016

Títol: Wireless Sensor Network based on UAV for Weevils Surveillance in Saudí Arabia

Autor: Marc Donadiós Garriga

Director: Oscar Casas Piedrafita

Data: 30 de Juny del 2016

Resum

El Morrut de les palmeres o Rhynchophorus ferrugineus és una espècie invasora que ha esdevingut una plaga nefasta per les plantacions de tot tipus de palmeres a Aràbia Saudita, entre molts altres llocs. Actualment, per a vigilància de l'estat de la plaga s'instal·len trampes a prop de les palmeres i una persona encarregada, visita una a una les trampes portant un recompte manual del nombre d'escarabats.

Aquest projecte té com a finalitat la millora de l'actual mètode de vigilància. Per a dur-ho a terme s'ha creat una xarxa de sensors sense fils i un conjunt de nodes sensors, que siguin capaços de saber el número de escarabats a la trampa i transmetin les dades cap a un node que les recollirà, per a ser processades posteriorment. Aquest nodes sensors estaran dins de les trampes, de manera que mesuraran el pes dins la trampes, i en definitiva, el número d'escarabats. Com a sistema de mesura s'utilitzen unes galgues extensiomètriques, amb un sistema d'acondicionament del senyal.

Aquest senyal es processa a una placa Arduino. A aquest microcontrolador se li pot incorporar una placa per encabir-hi dispositius de comunicació, en el nostre cas, els Xbees. Aquests dispositius utilitzen un protocol de comunicació de radio freqüència de baixa potència anomenat ZigBee. Un cop està creada la xarxa, i es permeti la comunicació entre dispositius, envien la senyal via port sèrie de l'Arduino.

Degut a la distància entre trampes, els nodes de terra no són capaços de comunicar-se entre ells. Per aquest projecte s'utilitza un dron, que sobrevola node per node agafant la informació de cada un dels nodes de terra i els emmagatzema a la memòria EEPROM (memòria no volàtil) per a poder ser processats a posterior. Els nodes de terra només són capaços de parlar amb el node mòbil, però el node mòbil és capaç de parlar amb qualsevol dispositiu.

Aquest projecte s'ha realitzat de manera que la solució sigui adaptable a diferents casos. És a dir, si les necessitats del projecte canvien, que es pugui mantenir la infraestructura o el sistema de mesura per a nous casos o aplicacions.

Title: Wireless Sensor Network based on UAV for Weevils Surveillance in Saudí Arabia

Author: Marc Donadiós Garriga

Director: Oscar Casas Piedrafita

Date: June 30th 2016

Overview

The Red Palm Weevil o Rhynchophorus ferrugineus is an invasive spice that had become a catastrophic plague for lots of palm trees in Saudi Arabia, but all around the world too. Nowadays, the surveillance of the plague is done using tramps placed near the palm trees, and one person comes to the tramp and counts manually the number of weevils into the tramp.

This project is aimed to improve the actual surveillance method. To realize it we have been created a Wireless Sensor Network and different nodes able to obtain the number of weevils into the tramp and they must transmit this data to a mobile node that collect all the data for processing. These sensor units are placed into the tramps, so they measure the weight of the weevils, and the number of them. The sensors are Strain Gauges and the signal conditioning system.

This signal obtained from the gauges is processed into the Arduino board. To this microcontroller we can add one shield in order to conect communication devices, in our case, Xbees. This devices use the ZigBee communication protocol, a low power Radio Frequency protocol. Once the network is created, and the communication between devices is allowed, we send the Package information via Serial communication, with the Arduino COM.

Due to the distance between ground nodes, they are no able to communicate between them, because the coverage is not enough to establish communication. For this project we use a drone that overflies all the ground nodes and collect the data from each node. This information is stored in the non-volatile EEPROM Memory. The ground node only can talk with the mobile node, but the mobile node must be able to talk with all the nodes.

This project is thought-out to be a adaptable method. So, we must be able to readapt the finality of the project, but maintain the network, infrastructure or sensing unit. The necessities can vary, but the base of the project must be the same.

INDEX

INTRODUCTION

Every day, the **technology developed** for control and surveillance **against plagues** and infestations, is improving and becomes more effective. In this project we are aimed to develop a wireless sensor network for surveillance of **Red Palm Weevils (RPW)** in Saudi Arabia. This little weevil is a palm borer and is one of the biggest palm trees plagues worldwide. In Saudi Arabia, the date palm is an important part of the religious, cultural, and economic heritage. And this heritage is threatened by the RPW in the last years. Due to this problem, some organizations work for control and eradicate this plague, but with some new methods and some implementations of the actual methods. This project is proposed by the Food and Agriculture Organization (FAO) of the United Nations to the company HEMAV SL.

The **main of the project** is to create a **technological demonstrator**, where we must be able to **compute the number of weevils** of the different tramps and sent this information via a **Wireless Sensor Network (WSN)**. The weevils are computed using scales connected to Arduino UNO. This WSN have **fixed ground nodes**, where we place the sensor that measures the number of weevils, and a mobile node. This **mobile node** is carried in an **Unmanned Aerial Vehicle (UAV)** or drone, which collects the data of all the ground nodes. For establish communication between nodes we use the **ZigBee communication protocol** and for its implementation we use **Xbee devices** with an **Arduino** like microcontroller. For this project we just want to realize a technological demonstrator, showing its correct work and a possible implementation. We are not aimed to do an optimisation of each part of the project. Thanks to this Wireless Sensor Network we improve the surveillance of the RPW in the different palm trees in Saudi Arabia.

1. RED PALM WEEVIL AND SPECIFICATIONS

1.1. Red Palm Weevil

The Red Palm Weevil (RPW) *Rhynchophorus ferrugineus* (Olivier, 1790) is a palm borer native to South Asia, which has spread mainly due to the movement of cryptically infested planting material to the Middle East, Africa and the Mediterranean during the last two decades. Globally, the pest has a wide geographical distribution in diverse agro climates and an extensive host range in Oceania, Asia, Africa and Europe. RPW is now known from all the continents of the world and is a key pest of coconut (*Cocos nucifera*) in South and South-East Asia, date palm (*Phoenix dactylifera*) in the Middle East and *P*. *canariensis* in Europe, and wherever they overlap. Although it was first reported as a pest of coconut in South Asia, it has become the major pest of date palm (*Phoenix dactylifera*), and the Canary Island date palm (*P. canariensis*) in the Middle East and Mediterranean basin, respectively. In total, RPW is reported to attack over 40 palm species belonging to 23 different genera worldwide. [1]

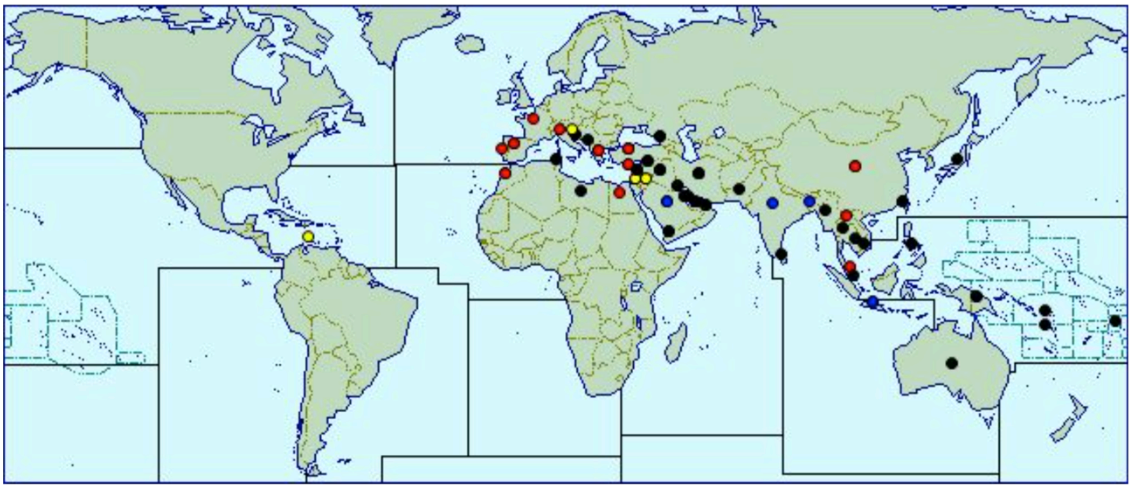

Fig. 1.1 Red Palm Weevil detected cases [5]

1.1.1. *General Biology and Damage*

Once they arrive at a palm, males and females typically seek protection from water loss by burrowing down into the petiole bases in the crown region, into fleshy wounds, or into the junction between off- shoots and the mother stem in palms such as the date palm. Like most weevils, RPW females use small mandibles at the distal tip of the distended rostrum to chew a hole into suitable host tissue before oviposition of a 2–3 mm long yellowish-colored egg. Eggs are often laid in close proximity to one another and take 2–4 days to eclose as small, first instar, legless larvae. After the egg eclose, the larvae have different stages, where its appearance change, and finally adopt the morphology of an adult RPW. The process from larvae to adult weevil can have a longevity of 150 days aprox. The duration of each larval phase is variable and depends upon available host nutrition, temperature and humidity. During this process, the larvae destroy the interior of the palm tree, eating everything inside the palm. [1]

Table 1.1 Morphology differences between Male and Female for the Red Palm Weevil [3]

Parameter	Male	Female
Weight (g)	0,81	1
Length (mm.)	33,5	34,4
Width (mm.)	11,5	11,7
Colour	dark reddish brown	Dark reddisch brown
Markings	black spots on pronotum vary from 1-6	Black spots on pronotum vary from 1-6
Snout	Short, stout with pad of small hairs on dorsal apical region	smooth, slender, slightly longer and shiny with out hair pad
Longevity with food (days)	62-78	59-75
Longevity without food (days)	33-43	33-40

In general, studies suggest that a gravid female will lay about 250 eggs over her lifetime (which may last up to 120 days) and may require multiple inseminations to insure fertility. Lethal infestations of RPW are in the range of 20–100 per palm but easily exceed 200. The adults can stay in the cocoon for several weeks before they emerge, according to abiotic conditions. Depending upon the condition of the host palm, the weevils may disperse for long distances (>900 m) or remain in the host to mate and recycle for another generation. It often takes 2–3 generations before a Canary Island date palm or a date palm will succumb to an RPW infestation. Depending on temperature, these generations can take place in one single year, but often require a minimum of 2 years (Dembilio and Jacas, 2011). Mark-release-recapture studies suggest that RPW can disperse about 7 km in less than a week (Abbas *et al*., 2006). [1]

1.1.2. *Detection and control*

The methods currently employed to control red palm weevil are largely based on the application of large quantities of synthetic chemical insecticides which, applied in the conditions of the sandy soils of the Gulf, has lead to the pollution of water courses around areas with palm weevil infestation.[2] Because signs and symptoms of RPW infestation are only clearly visible during the later stages of attack, efforts to develop early detection devices are being undertaken. Once infested by RPW, palms are difficult to manage and often die because of the cryptic habits of this pest. However, in the early stages of attack palms can recover after treatment with insecticides. [1]

Strict pre- and post-entry quarantine regulations have been put in place by some countries to prevent further spread of this highly destructive pest. Early detection of RPW-infested palms is crucial to avoid death of palms. Current tactics to manage the weevil, in zones like the Gulf and Asia, are largely based on insecticide applications although there are now deep concerns about environmental pollution. Much research has been conducted on other techniques, notably pheromone traps. However, there is now a strong emphasis on the development of integrated pest management (IPM) based on pheromone traps and biological control rather than insecticides. [2]

Research is continuing in other early detection methods in order to detect the RPW in early stages. The research techniques include acoustic detection of RPW larval stages, thanks to the sound impulses produced by the feeding and movements of the larvae. Other techniques are based on the odour of the trees infested by the RPW. This odor can be detected using trained dogs, but require frequent retraining (Nakash *et al*., 2000). [1] When a palm is infested by RPW the temperature of the tree increases significantly compered to a healthy palm. The study realized by Mohammed El-Faki Mozib in [4] are done using adult RPW, so we do not have to wait to egg laying and larvae stages for detect thermal changes in the palm trees. For this reason, this method is included in early detection methods.

The current methods for Biological control are:

- Insecticides: Insecticides are applied in a range of preventative and curative procedures designed to limit and contain the spread of an infestation. These procedures have been developed and refined since commencing in India in the 1970s when work on application of organophosphates and carbamates ensured these chemicals became the mainstay of the chemical approach to control. Methods employed have ranged from the precise, such as dusting of the leaf axils after pruning, to more general spraying or soaking of the tree trunk, to the direct injection of chemicals into the trunk of date or other palms. [2]

- Baiting and Trapping: Baiting and trapping of weevils uses a mixture of materials that are available to the smallholder and the commercial farmer. A trap receptacle such as a plastic bucket containing host plant material producing fermenting plant volatiles plus a quantity of synthetic aggregation pheromone is known to attract palm weevils. [2] This method is implemented since Hallett, R. H. In 1993 discover that RPW females produces and aggregates pheromone in the palm. This pheromone aggregated by the male has the following composition: ‘ferrugineol’ (4-methyl- 5-nonanol) and a minor component (4-methyl-5-nonanone) In [6] you can find some studies about the effectivity of different pheromones tested for trapping adult RPW.

1.1.3. *Trapping*

One another part for the weevil control is the trapping. Thanks to this part we can obtain the required information about the total population in one zone. Some studies in [1] defines some Trapping Protocols for succesfull RPW trapping and retention. These protocols have been obtained from some trapping experiments, where they obtain the best efficiency of each method. Related to our project we resume more important protocols:

- Trap design: RPW pheromone traps have been designed to facilitate easy entry of adult weevils into the trap while ensuring operational ease for handling and servicing (renewing food bait and insecticide solution) in the trap. The capacity of the trap must be 5L. The upright bucket trap with windows (the experts does not conclude about the number and the size of the holes for the perfect trap) cut equidistantly below the upper rim of the bucket is baited with a new pheromone lure hung from inside the lid of the bucket with a piece of wire. About 200 g of kairomone-releasing food bait (dates, green coconut petiole, sugarcane, etc.) is also added to the trap and is vital to ensure entry of the adults into the trap. he food bait is mixed in one liter of water laced with insecticide (0.05% carbofuran 3G or 0.1% carbaryl 50WP) solution to immobilize and kill the captured weevils. In the UAE, black traps recorded higher captures when compared with white traps.

- Trap placement: In order to maximize the lure longevity in the field it is essential to set RPW pheromone traps in the shade. Care should also be taken to avoid setting traps on palms in the age group susceptible to RPW attack of <20 years, or near, or on very susceptible palms. Higher RPW captures in UAE, were recorded when traps are currently set at ground level.

Fig. 1.2 Trap exemple [6]

- Trapping density: The trap density depends on the type of actuation required. For surveillance, 1 trap/100 ha or 1 trap every 1–2 km along motorable roads is enough. The trapping density depends on the intensity of the population and resources available for servicing the traps: for high density: 4–10 traps/ha; for low density: 1 trap/ha.

1.2. Actual surveillance and trapping method

The actual method in order to compute the number of weevils in each tramp is without using any kind of sensors. Every X days/weeks/months one person visit one by one all the tramps, and count the number of weevils in each one. Once he have count the number of weevils, kill all of them and empty the tramp, until another FAO's responsible come to the tramp. We must take into account that one tramp can be hundreds of meters, or a Kilometer, away from another one. So this project aims to develop a wireless technology capable to improve this method.

1.3. Specifications

Once we know the problem we can introduce the specifications that we must accomplish. These are the following specifications:

- We must create the Wireless Sensor Network. This networkwill have one node in continuous movement and 3 fixed ground nodes
- Define the type of network. The distance between tramps usually is higher to 100 meters, so we just need to ensure communication between the mobile node and the different ground nodes.
- Each ground node has to have a physical sensor in order to measure the number of weevils in each tramp. With a resolution of 1 weevil and with the characteristics exposed in 1.1.3

2. NETWORK ARCHITECTURE

In this chapter we explain what is a Wireless Sensor Network, the sensor architecture, the different topologies for this kind of networks and the wireless communication protocols.

2.1. Wireless Sensor Network

Wireless Sensor Networks (WSN) were first used in military missions. They are currently deployed in a wide range of civil applications as a sensor is becoming smaller and production costs are smaller. Due to his high utility and lots of many possible applications, the wireless sensor network experiments a boom and every day we use WSN to solve or improve solutions to current problems. In this work we will use one WSN to improve one surveillance method.

2.1.1. *Definition and parts*

WSN is an infrastructure composed of sensors nodes, computational devices and communication elements that provides information about the environment, and gives the ability of observe, instrument and react to events and phenomena occurred in this observed area. The WSN are composed of densely distributed nodes that support sensing, signal processing, embedded computing and connectivity, but the function of each node depends of the different topology and characteristics of our network. [7] A WSN is composed by 4 basic components:

- An assembly of distributed or localized sensors,
- An interconnecting wireless-based network
- A central point of information and clustering
- A set of computing resources at the central point to handle data correlation, event trending, status querying and data mining.

Arguing the previous list, we can resume that our network can transmit information from the sensor to the central point using the interconnected network. Sensor nodes are currently becoming smaller with more powerful capabilities and cheaper production costs. [8] The orders of magnitude, in the physical size, of the sensors nodes for the WSNs are very large: you can have sensors from nanometres to centimetres. Apart from the size, there are also some other stringent constraints for sensor nodes. These nodes must:

- Consume extremely low power
- Operate in high volumetric densities
- Have low production cost and be dispensable
- Be autonomous and operate unattended
- Be adaptive to the environment.

This extremely low power is related to the necessity of self-sufficiency of power, using batteries and energy captured from the environment. So the consume of the sensor node must be very low. [9]

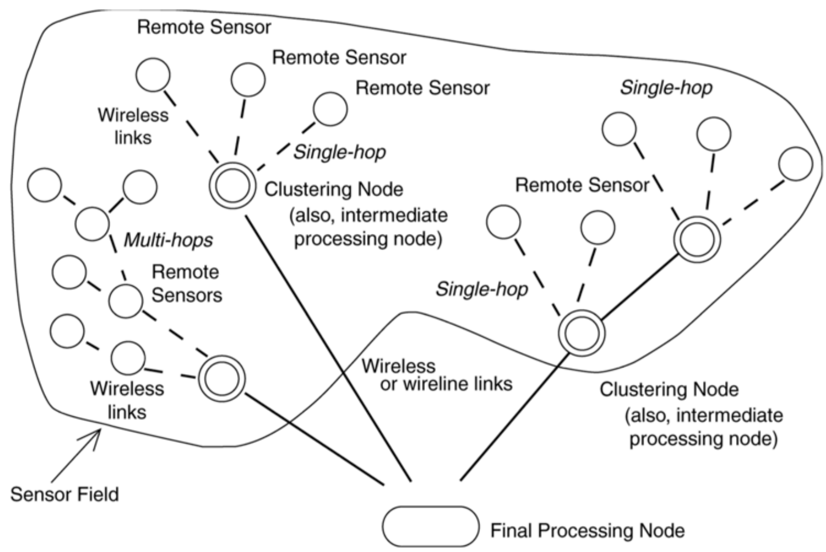

Fig. 2.1 Sensor network structure [7]

These sensors can be active or passive sensors, depending on the functionality and the energy required. Passive sensors tend to be low-energy devices and some examples of this type of sensors can be the strain-, humidity-, or temperature-measuring devices. Active sensors, instead of passive, tend to be high-energy systems and some examples of this type are the radars and sonars. [7]

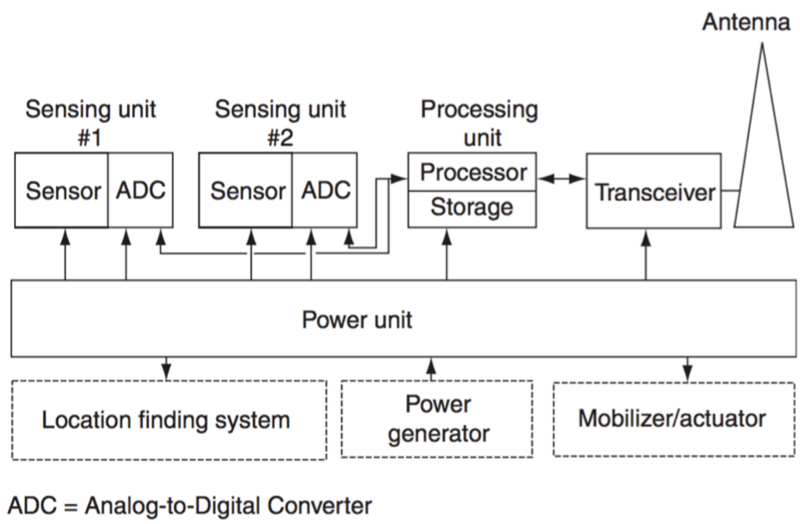

Fig. 2.2 Hardware for a sensing node [7]

2.1.2. *Basic components of Sensor Node*

The basic components of these sensor nodes will be described below:

- Sensing unit: The main functionality of the sensing unit is to sense or measure physical data from the target area. The analogue voltage or signal is generated by the sensor corresponding to the observed phenomenon. The continual waveform is digitised by an analogue to digital converter (ADC) and then delivered to the processing unit for further analysis.

- Processing unit: The processing unit, which is generally associated with a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the as- signed sensing tasks. [9]. There are currently several families of this unit including microcontrollers, microprocessors, and field-programmable gate arrays (FPGAs). [8]

- Communication unit: This unit is used to exchange data between individual nodes. The communication medium between the two nodes is through radio frequencies (wireless medium). Radio frequency-based communication fits the requirements of most wireless sensor applications because it provides relatively long range and high data rates, acceptable error rates at reasonable energy expenditure, and does not require line of sight between sender and receiver.

  For actual communication, both a transmitter and a receiver are required in a sensor node. The essential task is to convert a bit stream coming from a microcontroller (or a sequence of bytes or frames) and convert them to and from radio waves. In the transceiver, circuitry includes modulation, demodulation, amplifiers, filters and mixers. [10]

- Power unit: An appropriate energy infrastructure or supply is necessary to support operation from a few hours to months or years (depending on the application). [7] It is required that sensor nodes consume energy in the order of micro amps. Power savings may be achieved by reducing radio activity using low duty-cycle techniques and local computation to reduce data transmissions. Also, events from multiple sensors may be combined by a group of nodes before actual transmission to the rest of the network. [10]

Almost all of the node sensor platforms are designed to run on batteries that have a very limited lifetime. In order for wireless sensor networks to become a ubiquitous part of our environment, alternative power sources must be employed. Furthermore, the power supply (usually a battery) is also the limiting factor on the lifetime of a sensor node. In order to compare the best Energy reservoir for our nodes we must obtain the best energy per unit volume (J/cm^3). Therefore, in some cases such as micro-batteries, the maximum power density (μW/cm3) is also an issue for energy reservoirs. [11]

We can find different Energy Reservoirs for the Sensing Nodes [11]:

- Macro-Scale Batteries: Primary batteries are perhaps the most versatile of all small power sources. Power to the nodes is usually provided through primary batteries. However, batteries have a finite energy and need replacement when depleted, which increases the maintenance costs. Additionally, the state of charge (SoC) of a battery cannot be easily determined, so preventive replacement in critical applications prior actual battery depletion can be a must. In some other applications the replacement of batteries can be cost prohibitive or even not feasible. [12] Because batteries have a fairly stable voltage, electronic devices can often be run directly from the battery without any intervening power electronics.

- Macro-scale secondary (rechargeable) batteries are commonly used in consumer electronic products such as cell phones, PDA's, and laptop computers. It should be remembered that rechargeable batteries are a secondary power source. Nowadays we tend to only use rechargeable batteries due to its cost, the long lifetime, the energy provided and the autonomous life. We could find examples of other energy reservoirs but for a very specific uses and applications.

- Micro-Scale Batteries: The size of batteries has only decreased mildly when compared to electronic circuits that have decreased in size by orders of magnitude. One of the main stumbling blocks to reducing the size of micro-batteries is power output due to surface area limitations of micro-scale devices. The sizes of these batteries are very reduced compared to Macro-Scale Batteries, with similar performances. Thick films are on the order of 0.1 mm, but overall thicknesses are minimized by use of three-dimensional structures. While each cell is only rated at 1.5 V, geometries have been duty-cycle optimized to give acceptable power outputs at small overall theoretical volumes (4 mm by 1.5 mm by 0.2 mm) with good durability demonstrated by the electrochemical components of the battery.

- Micro-Fuel Cells: Hydrocarbon based fuels have very high energy densities compared to batteries. For example, methanol has an energy density of 17.6 kJ/cm3, which is about 6 times that of a lithium battery. Like batteries, fuel cells produce electrical power from a chemical reaction. A standard fuel cell uses hydrogen atoms as fuel.

- Micro Heat Engines: At large scales, fossil fuels are the dominant source of energy used for electric power generation, mostly due to the low cost per joule, high energy density (gasoline has an energy density of 12.7 kJ/cm3), abundant availability, storability and ease of transport. To date, the complexity and multitude of components involved have hindered the miniaturization of heat engines and power generation approaches based on combustion of hydrocarbon fuels.

- Radioactive Power Sources: Radioactive materials contain extremely high energy densities. As with hydrocarbon fuels, this energy has been used on a much larger scale for decades. However, it has not been exploited on a small scale as would be necessary to power wireless sensor networks.

Unlike power sources that are fundamentally energy reservoirs we have to recharge our sources in order to continue the normal operation. Power scavenging sources are usually characterized by their power density rather than energy density. Energy reservoirs have a characteristic energy density, and how much average power they can provide is then dependent on the lifetime over which they are operating. On the contrary, the energy provided by a power scavenging source depends on how long the source is in operation. Therefore, the primary metric for comparison of scavenged sources is power density, not energy density. [11]

We can find different methodologies for recharge our batteries [11]:

- Solar Cells: these devices provide a fairly stable DC voltage through much of their operating space. Therefore, they can be used to directly power electronics in cases where the current load is such that it allows the cell to operate on high voltage side of the “knee” in the I-V curve and where the electronics can tolerate some deviation in source volt- age. More commonly solar cells are used to charge a secondary battery.

- Temperature Gradients: Naturally occurring temperature variations can also provide a means by which energy can be scavenged from the environment. A number of researchers have developed systems to convert power from temperature differentials to electricity. The most common method is through thermoelectric generators that exploit the Seebeck effect to generate power.

- Wind/Air Flow: Wind power has been used on a large scale as a power source for centuries. Large windmills are still common today. However, the low efficiency might be not enough to generate power at a very small scale (on the order of a cubic centimetre) from airflow.

- Vibrations: Low-level mechanical vibrations are present in many environments. Examples include HVAC ducts, exterior windows, manufacturing and assembly equipment, aircraft, automobiles, trains, and small household appliances. Several researchers have developed devices to scavenge power from vibrations. Devices include electromagnetic, electrostatic, and piezoelectric methods to convert mechanical motion into electricity.

We could find other units depending on the application of the sensing node. The ideal wireless sensor must be scalable, must consume little power, have programmable software, capable of data acquisition, costs little to purchase and install and requires no real maintenance. [13]

2.2. **Wireless Sensor Network Topologies**

Wireless Sensor Networks can be composed of hundreds of devices and a good communication and data transfer between them is essential. With these requirements, a good design for this network is essential. The most common network topologies used in wireless sensor networks are Peer-to-Peer, Star, Tree and Mesh.

2.2.1. *Peer-to-peer topology*

Peer-to-Peer networks allow each node to communicate directly with another node without needing to go through a centralized communications hub. Each Peer device is able to function as both a “client” and a “server” to the other nodes on the network. One example of this kind of network is showed in Figure 2.3 [14]

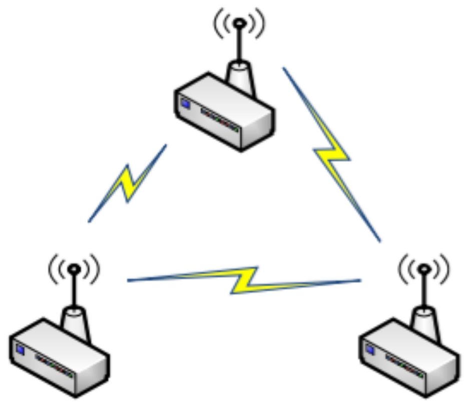

Fig. 2.3 Peer-to-peer network topology [14]

2.2.2. *Star topology*

Star networks are connected to a centralized communications hub. Each node cannot communicate directly with one another; all communications must be routed through the centralized hub. Each node is then a “client” while the central hub is the “server”. One example of this kind of network is showed in Figure 2.4 [14]

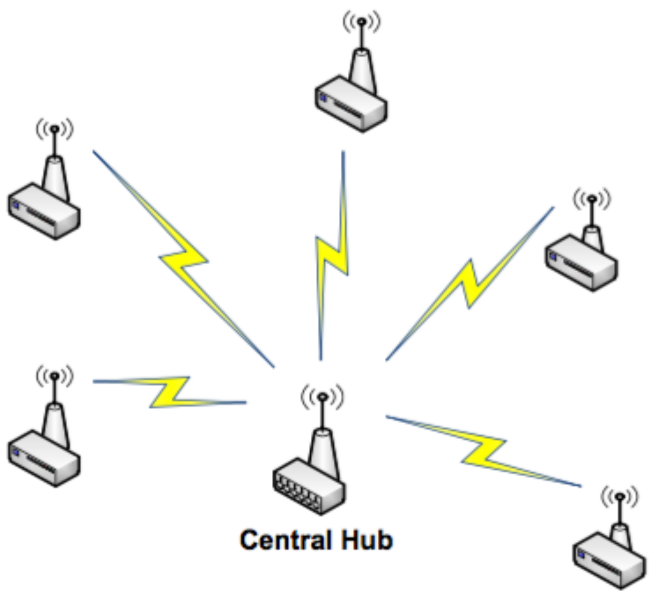

Fig. 2.4 Star netork topology [14]

2.2.3. *Tree topology*

Tree networks use a central hub called a Root node as the main communications router. One level down from the Root node in the hierarchy is a Central hub. This lower level then forms a Star network. The Tree network can be considered a hybrid of both the Star and Peer to Peer networking topologies. One example of this kind of network is showed in Figure 2.5 [14]

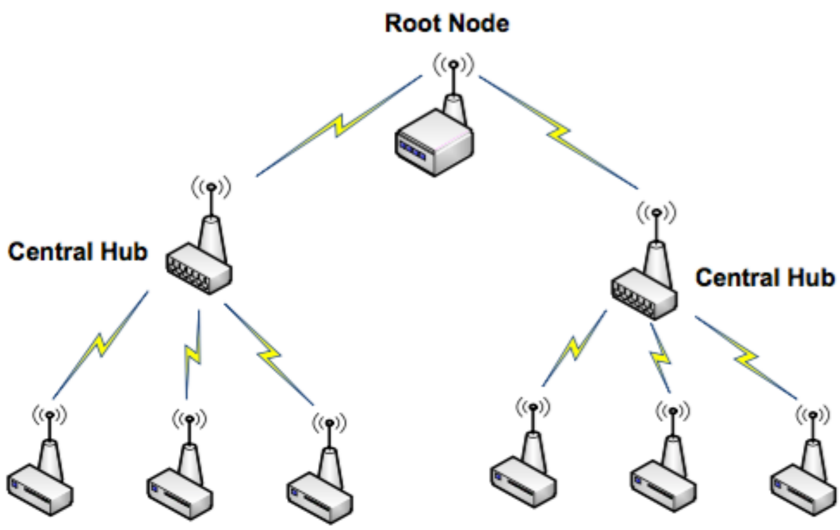

Fig. 2.5 Tree network topology [14]

2.2.4. *Mesh topology*

Mesh networks allow data to “hop” from node to node, this allows the network to be self-healing. Each node is then able to communicate with each other as data is routed from node to node until it reaches the desired location. This type of network is one of the most complex and can cost a significant amount of money to deploy properly. One example of this kind of network is showed in Figure 2.6 [14]

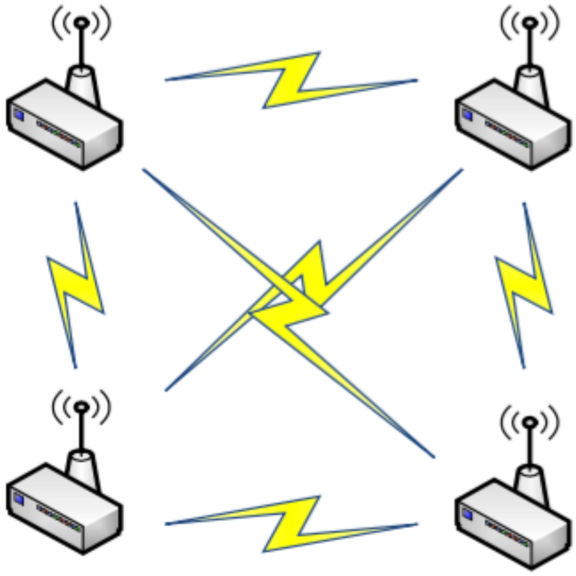

Fig. 2.6 Mesh network topology [14]

2.3. **Wireless Communication Protocols**

For this project we use Wireless communication due to the necessity of communicate some devices in distance. Using wireless we be able to gain flexibility in order to add devices or change the position of the existing nodes, without any extra infrastructure or wires. This technology reduces the cost for the networks infrastructure and avoids the necessity of connect devices physically.

Wireless technology standards are springing up all over the place, with a broad variety of uses. Most of these standards support a number of variants, with different range and throughput characteristics. In our case, taking into account the work’s environment of this network we study 4 different possible communication standards already used in wireless sensor networks. [13]

2.3.1. *Bluetooth*

Bluetooth technology is a short-range communications technology whose robustness, low power, and low cost make it ideal for a wide range of devices ranging from mobile phones and computers to medical devices and home entertainment products. [15]

For industrial applications, a wireless technology has to work well in a noisy environment. Bluetooth operates based on the features of Adaptive Frequency Hopping (AFH) and Forward Error Correction (FEC). It provides a universal short-range wireless capability. It operates in the 2.4 GHz frequency band and the devices within 10m of each other can share the data up to 720Kbps of capacity. This technology is also an authenticated one by sending the acknowledgement from the receiver to the transmitter before making the connection between devices. But its limitation is up to eight devices can communicate in a single network and it asks the confirmation about receiving the each data at every time and also it limits the packet size. [16]

2.3.2. *Bluetooth low energy (BLE)*

BLE is a new technology and responds to the need for a simple communication link between devices that must operate with very low power consumption, and be very low cost. It is not, obviously, a one-for-all technology; instead, it is particularly suited for those devices that need to transfer very small quantity of data, and do so within relatively short ranges. [17]

LE sensor devices are typically required to operate for many years without needing a new battery. LE technology is primarily aimed at mobile telephones, where it is envisaged that a star network topology, similar to Bluetooth, will often be created between the phone and an ecosystem of other devices. [16]

LE is also known as Bluetooth v4.0 and is part of the public Bluetooth specification. As a result of being a standard, LE benefits from all the advantages of conformance and extensive interoperability testing at unplug fests. A device that operates Bluetooth v4.0 may not necessarily implement other versions of Bluetooth; in such cases it is known as a single-mode device. Most new Bluetooth chipsets from leading Bluetooth silicon manufacturers will support Bluetooth and the new LE functionality. [16]

BLE substantially reduces classic Bluetooth’s peak, average, and idle mode power consumption, with energy efficiencies that can be 20 times higher than Classic Bluetooth. The extremely low peak, average, and idle currents of BLE chipsets, shown in Table 1, enable BLE radios to work with very small battery power sources for a year and more. [17]

2.3.3. *Wi-Fi*

Wi-Fi stands for Wireless Fidelity, which refers to wireless technology that allows devices to communicate over a wireless signal. This network is based on the IEEE standard 802.11; including 802.11a, 802.11b, 802.11g and 802.11n, by using the centralized router devices can share the Wi-Fi signal. Wi-Fi networking technology that uses waves to allow high-speed data transfer over short distances. Wi-Fi allows local area networks (LANs) to operate without cable and wiring. It is popular for the home and business networks. Generally, it can be used to provide the wireless broadband internet access for many modern devices such as laptops, smart phones, tablet and computers with authentication. By increasing the number of devices in a single Wi-Fi connection, the strength of the signal provides to each device becomes weak.[16]

The data transfer rate can be higher of 54 Mbps and have a typical transmission range of 35 meters depending on the physical environment and the gain of the performances of the antenna. The operating frequency varies depending on the protocol used, but varies between 2.4 GHz and 5.8 GHz.[18]

2.3.4. *ZigBee*

ZigBee is a specification for a suite of high-level communication protocols used to create personal area networks built from small, low-power digital radios. Is based on the IEEE 802.15.4 standard. The ZigBee wireless networking standard fits into a market that is simply not filled by other wireless technologies. While most wireless standards are striving to go faster, ZigBee aims for low data rates. ZigBee is great at wireless control, where anywhere from two to thousands of nodes are all connected together, in multi-hop mesh network. While other wireless technologies are designed to run for hours or perhaps days on batteries, ZigBee is designed to run for years.[19]

Its low power consumption limits transmission distances to 10–100 meters, depending on power output and environmental characteristics. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device. [13]

2.4. **Proposed Architecture**

Once we have seen the types of networks and communication protocols we can define our case. For the implementation of **our project** we decide to use **Zigbee** standard for the communication network. The main reasons for choose Zigbee instead of the others standards descried previously are the following:

- Easy to implement with Arduino
- Cheap technology
- Simplest technology
- Enough range for the communication between mobile node and the mobile node (100 m with Line of Sight and outdoor)
- Very low power consumption
- Ensures a high quantity of nodes interconnected
- High longevity life

Finally we decide to choose a **Hybrid network** between **Star and Peer-to-Peer** topology. Due to the distance between the tramps (1 tramp / 1 ha), we **cannot ensure communication between the different ground nodes**, because the range for a ZigBee device is 100m with Line of Sight and sometimes we could no ensure this condition. The **communication** will be between the **ground node** and **mobile node**. The sensor node will measure the number of weevils and sent the data only when the node carried on the drone ask for the information. We must take into account that the mobile node must be able to communicate to all the ground nodes and the ground nodes just can talk with the mobile node. This avoids problems of useless communication between ground nodes.

3. DESING AND IMPLEMENTARION

In this chapter we describe the design of the sensing unit, the wireless sensor network an its implementation.

3.1. Sensing unit design

In order to compute the weevils that enter into the tramp, we can use different methods. Here you can find explanation for the most common methods to compute elements in a close space:

3.1.1. *Optical count*

The Optical Count is the most sophisticated method for counting elements into a closed space. At the same time, requires high inversion, due to high cost of the hardware and software to control the sensors. The best two ways to count elements using Optical Technology are explained in [20]:

- Thermal sensors: these thermal sensors are based on thermal cameras and control software to detect the movements of a warm body. But we know that weevils are Cold-blooded animals.
- Movement sensors: uses sensors, other electronics, and software to detect the movements of people through a doorway. Multiple counters can be interconnected into an integrated counting system. The count in the room increases when an element enters and decreases when an element exits and is communicated to a connected computer system via a communication protocol. Each counter has two closely spaced sensors. Normally, the counter increases or decreases the total count by full element-units; however, in some situations the total count may increase or decrease by a half element (presumably when only one sensor detects movement).

These kinds of methods are expensive and needs more technology to develop it. Due to the sensor node must be self and must be low power systems. This solution has very high power consume increasing the size of the batteries. It reduces the lifetime of the batteries, because the system must be connected anytime in order to detect any new weevil.

3.1.2. *Transducers*

Transducers change mechanical movement to electronic signals. Exist two types of encoders, depending of which type of mechanical movement convert to electronic signals:

- Rotatory encoder: converts rotatory position to electronic signal. Converts the angular position or motion of a shaft to voltage or digital output.
- Linear encoder: converts linear position to electronic signal.

The two types of encoders can be sub-divided into two different encoders, depending on how it works:

Absolutes encoders: give unambiguous position between the travel ranges without requiring knowledge of any previous position. This type of encoders can retain the position dates during the loss of alimentation.

Incremental encoders: determines relative position, looking only at the differences between measurements. The encoder engine sends out pulses in channels (called quadratures) and the offsets in these pulses indicate motion. They are limited by only providing change information, so the encoder requires a reference device to calculate motion.

This solution allows us to count one by one the weevils that enter to the trap, using a mechanical gate that the weevils must move. The encoder captures this movement, and we obtain one increment in the weevil's counter. This allows us a high precision, knowing the exact number of weevils in order to realize a very accurate control of them. [21]

3.1.3. *Strain Gauges*

Strain Gauges are devices aimed to measure deformation, pressure, load, etc… over them. These devices take advantage of the physical property of electrical conductance that measures the ease, which an electrical current passes through a conductor. When one electrical conductor is stretched, always within the elastic limit, without permanent deformation, the electrical resistance is changed.

So, using this deformation of the material, we can correlate the load over the gauge with the resistance of this gauge. Gauges have one active axis, normally the longitudinal, where we can find the maximum deformation. The property related to this axis is the gauge length that is the grid length in the sensitive direction. The other axis is practically insensible to the deformation; normally this axis is the transversal. [22]

The Gauge Factor (GF or K) is the relation between the resistance variation in function of the deformation suffered by the gauge:

$$\mathrm{K} = \frac{\Delta \mathrm{R}/\mathrm{R}_0}{\Delta \mathrm{L}/\mathrm{L}_0} = \frac{\Delta \mathrm{R}/\mathrm{R}_0}{\varepsilon} \tag{3.1}$$

Where: ΔL is the length increment due to force applied

L_0 is the original gauge length of the grid
ΔR is the variation of the resistance due to the length variation
R_0 is the original gauge resistance
ε is the strain of the material

Once we have the relation between the deformation of the gauge and the resistance we can readapt the expression in order to obtain the resistance in function of the deformation and the Gauge Factor:

The final electrical resistance of our sensor will be:

$$\mathrm{R} = \mathrm{R}_0 + \Delta\mathrm{R} \quad \textbf{(3.2)}$$

But if we isolate the resistance variation (ΔR) of the Gauge Factor equation we can obtain:

$$\Delta\mathrm{R} = \mathrm{K} \cdot \varepsilon \cdot \mathrm{R}_0 \quad \textbf{(3.3)}$$

Finally we can combine both equations:

$$\mathrm{R} = \mathrm{R}_0 + \mathrm{K} \cdot \varepsilon \cdot \mathrm{R}_0 \quad \textbf{(3.4)}$$

$$\mathrm{R} = \mathrm{R}_0(1 + \mathrm{K} \cdot \varepsilon) \quad \textbf{(3.5)}$$

Now we have the electrical resistance of our gauge in function of the deformation but in order to connect the strain gauge to the microcontroller, we must do conditioning to have a voltage output. For change resistance output into a voltage output we can use different “Wheastone bridges circuits”. These bridges are used when we have small resistance changes in gauges but we can increase precision using more gauges. We can put our gauge, or gauges in the place of the different resistance in the Figure 3.1 (R1, R2, R3 or R4 can be a gauge). We apply one voltage divider between the points A and B and we obtain:

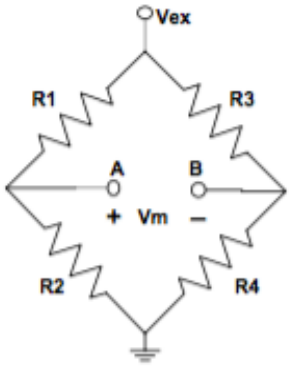

Fig. 3.1 Wheastone bridge [22]

$$V_A = V_{ex}\left(\frac{R_2}{R_2+R_1}\right) \tag{3.6}$$

And for the other node:

$$V_B = V_{ex}\left(\frac{R_4}{R_4+R_3}\right) \tag{3.7}$$

so, Vm, that is the value measured, will be:

$$V_m = V_{ex}\left(\frac{R_2}{R_2+R_1} - \frac{R_4}{R_4+R_3}\right) \tag{3.8}$$

If we want to obtain the maximum sensitivity of the voltage divider must have the same resistance in pairs: R_1=R_2 and R_3=R_4. If we have the same resistance in both positions and both are the same gauge we can say that both resistance have the same R_0. But they change his resistance value when we apply pressure over them:

$$R_i = R_0 + \Delta R_i \qquad \text{for } i \in N\,[1,4] \tag{3.9}$$

If we substitute these equations into the general equation and we operate we obtain:

$$V_m = V_{ex}\,\frac{R_0(\Delta R_2 - \Delta R_1 + \Delta R_3 - \Delta R_4) + \Delta R_2 \cdot \Delta R_3 - \Delta R_1 \cdot \Delta R_4}{4 \cdot \Delta {R_0}^2 + 2 \cdot R_0 \cdot (\Delta R_1 + \Delta R_2 + \Delta R_3 + \Delta R_4) + (\Delta R_2 + \Delta R_3) \cdot (\Delta R_1 + \Delta R_4)}$$

(3.10)

But this solution has no linearity between V_m and the different gauge's resistance variations. In the case of having small variations in the output of the sensor, as our case, we can linearize the voltage measured:

$$\Delta R_i \ll R_0 \quad \text{for } i \in N\,[1,4] \tag{3.11}$$

This assumption is reasonable in the case that we are using gauges, because the voltage variation into the two terminals of the gauges sensors is very small compared with the excitation voltage of the sensors. Normally we have variations of mV and an excitation voltage of 5-12V. With the previous assumption, and despising some small terms and second order terms of the equation, we obtain:

$$V_m = V_{ex} \cdot \frac{\Delta R_2 - \Delta R_1 + \Delta R_3 - \Delta R_4}{R_0} \tag{3.12}$$

Finally we obtain one solution with linearity between V_m and the different resistance variations of the gauges by using 4 sensors. At the same time temperature compensation is not explained. In some cases that the gauges supports heavy load during long period of time, or they measure heavy loads in a small period of time, the gauge can be affected by the self-heating produced by the deformation of the gauge. In our case we use the gauge for low load and for a long period of time but the gauges usually have high temperature sensitivity. In order to compensate this thermal factor we can use one passive sensor, close to the active gauge. Both sensors have the same temperature, but the load does not affect the passive sensor. By this way we can compensate the thermal effect and we can increase the sensitivity. [22]

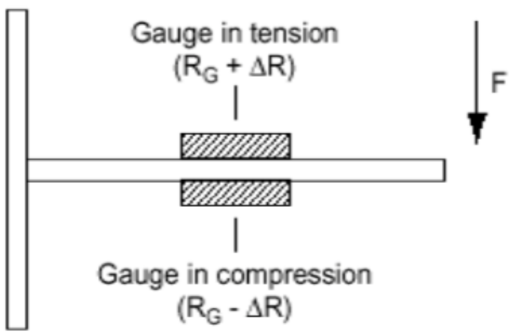

Fig. 3.2 Physical example for a Pull-Push configuration [22]

We can optimize our system adding a Push-Pull configuration. In this kind of configuration we use gauges for measure compression and tension, like in Fig 3.2. We can use 2 sensors or 4 sensors. In the case that we only use 2 gauges we must add 2 resistors, with a resistance value as close as possible to the R_0 of the gauges and have to be high precision resistances and with high stability. With this system we can compensate the temperature effect and if we use 4 gauges we obtain the highest sensibility. Otherwise this 4 gauges system is the most expensive solution. [22]

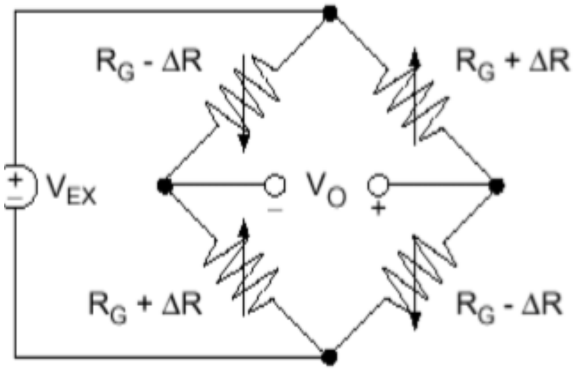

Fig. 3.3 Full bridge for a Pull-Push configuration [22]

3.2. Sensing unit implementation

Finally we decide to use a Strain Gauge as our sensor. This solution, instead of the encoder's solution, is easier: we don't have problems when we have cut off the system alimentation, for example. At the same time, this solution is cheaper, because the price of one Gauge is much low as Encoders. The Optical sensor is discarded due to the high cost and the complexity for the implementation. For the real implementation we will buy a Load cell, where we have the Gauge and all the conditioning circuit so, the output is already tension. So, our technological demonstrator will be done using a load cell, obtained from a bathroom scale.

For our particular case we buy some kitchen scales, showed in Figure 3.4. We do an inverse engineering process. We had to realize some test in order to know which type of gauge we have: single gauge, half bridge, full bride of gauges… After some test we ensure that we had 2 gauges in a push-pull configuration shown in Fig 3.2.

Fig. 3.4 Kitchen Scale used for the project

The next step is to make the conditioning of the signal. The voltages at the output of the load cell are millivolts so we must amplify this output signal in order to be able of detect a good sensitivity of the load cell with the Arduino. For this voltage amplification we use the AD623. This device can provides the difference of both gauges of the bridge of the load cell and, at the same time, amplifies this difference. This amplifier is a Rail-to-Rail device that ensures us an output without high voltage losses. Has high accuracy and work with single supply tension. At the same time is a low cost device with low power consumption.

In our case we would like to adjust the output to 5 volts with a load of 1 kilogram, considering that we do not want to measure more than 1 kg of weevils into the tramp. The output with a theoretical maximum amplification and without load over the cell is 0,285 V, and if we put a load of 1 kg over the load we obtain an output of 1,16 V. The AD623 can provide a theoretical maximum gain of 1000. An external resistance determines this amplification. We can compute the gain with the equation 3.13:

$$G = 1 + \frac{100 \cdot 10^3}{R_G} \tag{3.13}$$

Although the specifications of the amplifier defines a maximum amplification of 1000 using a resistance of 100 ohms, we can obtain more amplification factor using low resistance, but we can saturate the amplifier. After several tests, we can ensure optimal amplification with a 39 ohm resistance without saturate the amplifier, obtaining an amplification approximately of 2500.

Instead of obtain the 5V at the output we realize one limit of the load cell. The maximum output voltage provided for the load cell multiplied with the gain cannot arrive to the 5V. It saturates around 3,5 volts.

The following step is to add a low-pass filter (LPF) in the output of the amplifier. We add a LPF with a cut frequency of 160 Hz, using a resistance of 10 K ohms and a capacitor of 10 microF, enough for the utility of this sensor. We compute the cut frequency using the equation 3.14:

$$f_c = \frac{1}{2\pi RC} \tag{3.14}$$

In Fig. 3.5 you can see the circuit design for the gauge and the signal conditioning:

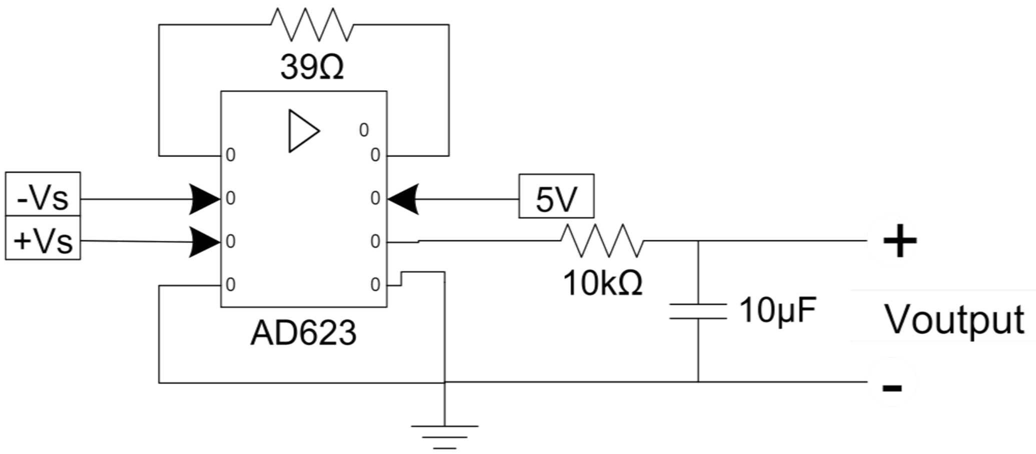

Fig. 3.5 Circuit for the signal conditioning

Finally we have our sensor node finished and we can compute the final resolution of our sensing system. Knowing that one Weevil has a weight around 1 gram, we want to ensure a resolution of 2-3 weevils. We decide to ensure a The Dynamic range can be computed using equation 3.15:

$$DR = \frac{\Delta \text{Weight}}{\text{resolution desired}} = \frac{1000\text{ g}}{1\text{ g}} = 1000 \leq 2^{\text{n-bits ADC converter}} \tag{3.15}$$

$$n \geq 10 \text{ bits} \tag{3.16}$$

With this approximation, we can ensure enough resolution, because the Arduino UNO has a ADC converter of 10 bits.

But the real resolution depends on the output voltage of the sensor:

$$\text{Real Resolution} = \frac{\Delta \text{V sensor} * \text{Gain}}{2^{\text{n bits ADC converter}}} \quad \textbf{(3.17)}$$

Using equation 3.17 we have 3 different resolutions (in Voltage) for each sensor node:

Table 3.1 Different nodes resolutions in voltatge

Node	V sensor (0 weight) *Gain	V sensor (saturation) *Gain	Real Resolution (voltage)
1	0,360 V	3,474 V	0,00305 V
2	1,163 V	3,509 V	0,00225 V
3	0,313 V	3,406 V	0,00302 V

And now, using the different calibration equations we can obtain the final resolution (in weevils):

Table 3.2 Different nodes resolutions in weevils

Node	V sensor (0 weight) *Gain	Bits (0 weight) *Gain	V sensor (1kg) *Gain	Bits (1kg) *Gain	Real Resolution (weevils/bit)
1	0,360 V	73	2,599 V	542	2,13
2	1,163 V	240	3,375 V	703	2,15
3	0,313 V	62	2,662 V	552	2,04

3.3. Wireless Sensor Network Design

In this part you can find all the information, performances and characteristics related to the hardware and software needed to carry out the project: from the technology used in order to enable communication, to the sensors needed to collect all the necessary information. The different interfaces used to program the sensors and the power management for enable a higher lifetime of the sensor are explained as well.

3.3.1. *Arduino Hardware*

Arduino is an open-source prototyping platform based on a simple microcontroller board, and development environment for writing software for the board. This platform allows to sense and control in an autonomous way, once you have programmed it.

Arduino boards are able to read analogue and digital inputs, and turn it into an output in order to control lights, motors or other physical instruments. For programming it can be used the open-source IDE, that you can download for free. The programming language of the IDE is based on C language.

This flexibility and high capacity of customization makes this platform perfect to this project. It offers a variety of digital and analogue inputs, SPI and serial interface and digital and PWM outputs. It is also easy to use and implement, since it connects to computer via USB and communicates using standard serial protocol. The cost of each board costs around 30 euros, so it is not expensive.

Arduino has a high board variety depending on the necessities and performance requirements of the user. Every board has common components, such as a microcontroller, a flash memory, power supply, a USB connection, digital input/output pins, analogue input pins, etc. For this project have been used two different boards.

- Arduino UNO: The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. [23]

Table 3.3 Arduino UNO specifications [23]

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limit)	6-20 V
Digital I/O Pins	14
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

- Arduino MEGA: The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno. [24]

Table 3.4 Arduino MEGA specifications [24]

Microcontroller	ATmega2560
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limit)	6-20 V
Digital I/O Pins	54
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

- Wireless Proto shield: allows an Arduino board to communicate wirelessly using a wireless module. It is based on the Xbee modules from Digi, but can use any module with the same footprint. It can be used as a serial/usb replacement or you can put it into a command mode and configure it for a variety of broadcast and mesh networking options. The shields breaks out each of the Xbee's pins to a through-hole solder pad. An on-board switch allows the wireless module to communicate with the USB-to-serial converter or with the microcontroller. This shield is pin-to-pin compatible with both Arduino microcontrollers used in the project.

  Switch settings:

  - When in the Micro position, the DOUT pin of the wireless module is connected to the RX pin of the microcontroller; and DIN is connected to TX. The wireless module will then communicate with the microcontroller. This switch position is mandatory when we run the program for communication between devices.

- With the switch in the USB position, the DOUT pin the wireless module is connected to the RX pin of the USB-to-serial converter, and DIN on the wireless module is connected to the TX pin of the USB-to-serial converter. This means that the module can communicate directly with the computer. The microcontroller on the board will be bypassed. To use the shield in this mode, you must program the microcontroller with an empty sketch, or remove it from the board. [25]

3.3.2. *Arduino IDE*

The Arduino integrated development environment - or Arduino Software (IDE) - is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. The IDE contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus.

Programs written using Arduino Software are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor. [26]

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier.

3.3.3. *Zigbee*

As has been explained in WSN Chapter, Zigbee is the Wireless Communication Standard used to realize this project. In this section you can find some information about its Communication Architecture, frequency of operation and the different ZigBee devices.

As many other communication Standards, Zigbee uses the OSI 7-layer networking model.The Open Systems Interconnection (OSI) is a reference tool for understanding data communication between any two-networked systems. It divides the communications processes into seven layers, as you can see in Fig 3.6. Each layer both performs specific functions to support the layers above it and offers services to the layers below it. Without this model, the communication between systems would be chaotic and difficult to understand and implement. [27]

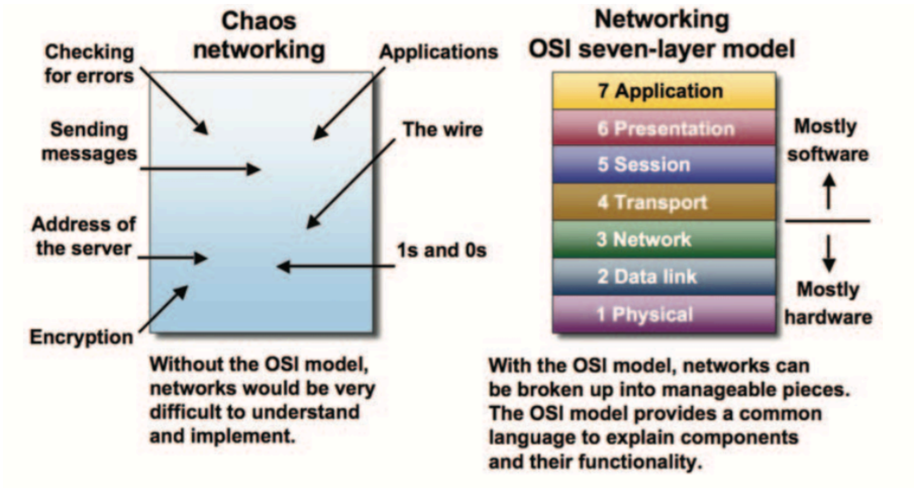

Fig. 3.6 OSI 7-layer networking model [27]

The 3 lowest layers are aimed to passing data traffic between systems, all understood from a physical point of view. In these 3 layers, the Hardware is the most important part over the Software. In the other hand, the 4 top layers come into play in the end system to complete the process. In these layers, the Software is the most important part over the Hardware. [27]

ZigBee does not exactly fit the OSI 7-layer networking model, but it does have some of the same elements. Including the PHY (physical), MAC (link layer), and NWK (network) layers. Layers 4–7 (transport, session, presentation, and application) are wrapped up in the APS and ZDO layers in the ZigBee model. This Architecture can be found in Fig. 3.7.

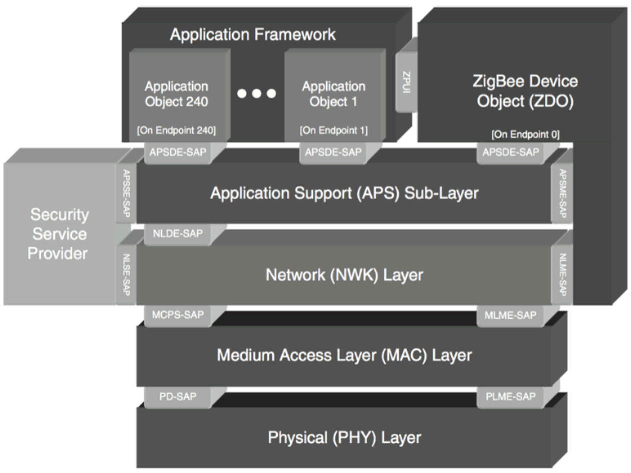

Fig. 3.7 ZigBee architecture based on OSI 7-layers networking model [28]

The two lowest layers, the MAC and PHY, are defined by the IEEE 802.15.4 specification. The PHY layer simply translates packets into over-the-air bits and back again. Defines the physical operation of the ZigBee device including receive sensitivity, channel rejection, output power, number of channels, chip modulation, and transmission rate specifications.[28] Here you can find an explanation for every layer:

- The MAC layer provides interface between physical layer and network layer. MAC layer is responsible for generating beacons and synchronizing devices to the beacon signal in a beacon enabled services. [29]

- The NWK layer is responsible for mesh networking, which includes broadcasting packets across the network, determining routes for unicasting packets, and generally making sure packets are sent reliably from one node to another. The network layer also has a set of commands for security purposes, including secure joining and re-joining. ZigBee networks are all secured at the NWK layer, and the entire payload of the NWK frame is encrypted.

- The APS layer is responsible for application meaning. It acts as a filter for the applications running above it on endpoints to simplify the logic in those applications. It understands what clusters and endpoints mean, and checks to see if the endpoint is a member of the Application Profile and (if present) group before sending the message on up.

- The ZDO layer is responsible for local and over-the-air management of the network. It provides services to discover other nodes and services in the network, and is directly responsible for the current state of this node on the network. [28]

Zigbee utilizes a direct-sequence spread spectrum modulation and operates on a fixed channel. The PHY layer is defined by IEEE 802.15.4 and it defines 16 operating channels in the 2.4 GHz frequency band. This band is a free band of the RF spectrum, where you can use it freely for many radio applications. The channels of this band are numbered from 11 to 26.

For regional operation are defined 2 different frequencies: 868MHz for Europe and 915 MHz for the Americas. Channel 0 works at 868.3 MHz and channels 1 to 10 works between 902 MHz and 928 MHz. All of the channels defined by IEEE 802.15.4 are showed in Fig. 3.8.

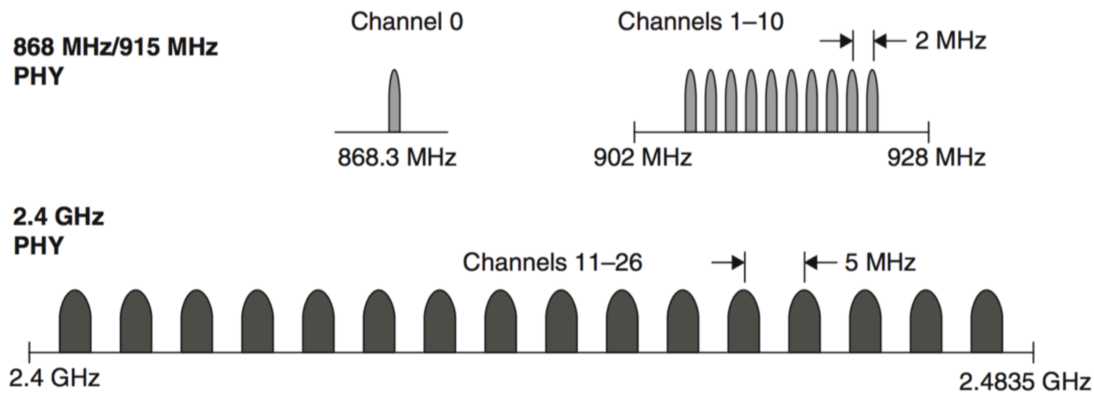

Fig. 3.8 Different channels for ZigBee communiction [28]

There are three different types of ZigBee devices, depending on the use and the type of functions that develop:

- ZigBee Coordinator (ZC): Forms the root of the network tree and might bridge to other networks. Is not needed for normal operation of the network, but is required to allow nodes to join or leave the network, as it contains the Trust Center. Only the Trust Center can decide whether to allow a node on a ZigBee network, or to deny it access. [27]

  There is exactly one coordinator in each network. It is responsible for initiating the network and selecting the network parameters such as radio frequency channel, unique network identifier and setting other operational parameters. It can also store the information about network, security keys. [29]

- ZigBee Router (ZR): Router acts as intermediate nodes, relaying data from other devices. Router can connect to an already existent network, also able to accept connections from other devices and be some kind of re-transmitters to the network. Network may be extended through the use of ZigBee routers. At the same time, the ZR, like ZC, allow other nodes to join the network. [29]

- ZigBee End Device (ZED): End Device can be low-power /battery-powered devices and they can collect various information from sensors and switches. They have sufficient functionality to talk to their parents (either the coordinator or a router) and cannot relay data from other devices. This reduced functionality allows for the potential to reduce their cost. They support better low power models. These devices do not have to stay awake the whole time, while the devices belonging to the other two categories have to. Each end device can have up to 240 end nodes which are separate applications sharing the same radio. [29]

Fig. 3.9 shows an example of WSN with the three types of devices:

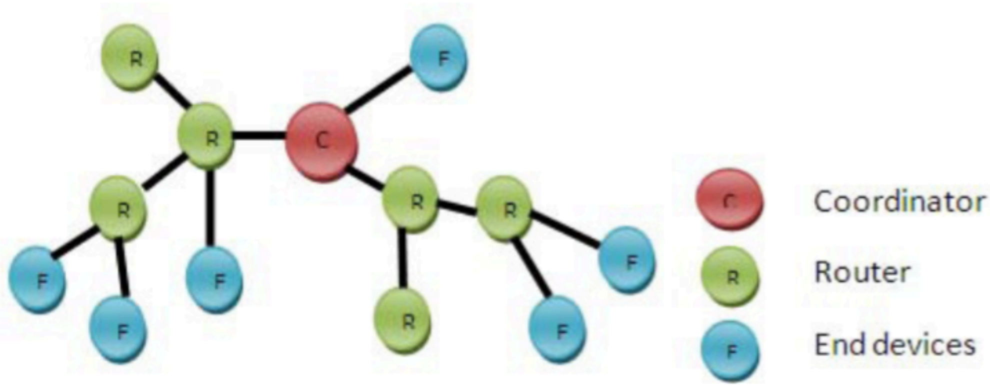

Fig. 3.9 Example of ZigBee network with 3 types of devices [29]

The following table resumes all the functionalities that have the three different ZigBee devices:

Table 3.5 ZigBee Functions for each Device type [30]

ZigBee Network Layer Function	Coordinator	Router	End Device
Establish ZigBee Network	✓		
Permit other devices to join/leave the network	✓	✓	
Assign 16-bit network addresses	✓	✓	
Discover and record paths for efficient message delivery	✓	✓	
Discover and record list of one-hop neighbours	✓	✓	
Route network packets	✓	✓	
Receive or send network packets	✓	✓	✓
Join or leave the network	✓	✓	✓
Enter sleep mode			✓

The XBee/XBee-PRO ZB are RF Modules designed by Digi Ingternational, ables to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. For this project has been choose this RF Modules considering their high performances despite its low cost and its low power consume. Some of their specifications are shown in the following table:

Table 3.6 Xbee Modules Specifications [31]

	Specification	Value
Performance	Indoor/Urban range	Up to 40 m
	Outdoor RF line-of-sight range	Up to 120 m
	Transmit Power Output	1,25-2 mW
	RF date Rate	250000 b/s
	Data throughput	Up to 35000 b/s
	Receiver sensitivity	-96 dBm
Power	Supply Voltage	2,1-3,6 V
	Operating Current (TX)	35-40 mA
	Operating Current (RX)	38-40 mA
	Idle Current	15 mA
	Power Down Current	<1 µA
General	Operating Frequency Band	ISM 2,4 GHz
	Operating temperature	-40 to 85 ^{o}C
	Supported Network Technologies	Point-to-point, point-to-multipoint, peer-to-peer and mesh
	Number of Channels	16
	Channels	11 to 26

The XBee modules support both transparent (AT) and Application Programming Interface (API) serial interfaces.

Transparent Mode: When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DIN pin is queued up for RF transmission. When RF data is received, the data is sent out through the DOUT pin. The module configuration parameters are configured using the AT command mode interface.

API Mode: API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module. [31]

These are the different modes of operation for our RF modules[31]:

- Idle Mode: When not receiving or transmitting data, the RF module is in Idle Mode. The module shifts into the other modes of operation under the following conditions:
  - Transmit Mode (serial data in the serial receive buffer is ready to be packetized)
  - Receive Mode (valid RF data is received through the antenna)

- Sleep Mode (End Devices only)
- Command Mode (Command Mode sequence is issued)

- Transmit Mode: When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data. When data is transmitted from one node to another, a network-level acknowledgment is transmitted back across the established route to the source node. This acknowledgment packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgment is not received, the source node will re-transmit the data.

- Receive Mode: If a valid RF packet is received, the data is transferred to the serial transmit buffer.

- Command Mode: To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands.

- Sleep Mode: Sleep modes allow the RF module to enter states of low power consumption when not in use. The XBee RF modules support both pin sleep (sleep mode entered on pin transition) and cyclic sleep (module sleeps for a fixed time). Remember that this Mode is only available for End Device Modules.

3.3.4. *Xbee external Hardware*

For the XBee configuration we use one USB to serial base unit: SparkFun Xbee explorer USB. This unit works with all XBee modules, and its pin-to-pin compatible with other Wireless devices. By using a mini USB cable you will have direct access to the serial and programming pins on the XBee unit.
The highlight of this board is an FT231X USB-to-Serial converter. That’s what translates data between your computer and the XBee. There’s also a reset button, and a voltage regulator to supply the XBee with plenty of power. In addition, there are four LEDs that’ll help if you ever need to debug your XBee: RX, TX, RSSI (signal-strength indicator), and power indicator.[33]

Be careful, because for control the XBee module with this unit yoy must install to you computer the FTDI drivers. You can download the drivers directly from [34].

3.3.5. *Xbee Software*

XCTU is free software, multi-platform application designed to enable developers to interact with Digi RF modules through a simple-to-use graphical interface. It includes new tools that make it easy to set-up, configure and test XBee RF modules. XCTU includes all of the tools a developer needs to quickly get up and running with XBee. The program allows the RF modules programming in a very visual and easy way. All the parameters of our XBee device can be changed with this tool, like Node Identifier, destination address, operating channel, PAN ID, etc.

We recommend to only use the last software version (6.3.1), so every new version could have problems and in this case, the previous version (5.2.8.4) is more robust versus any inconvenient. [35]

3.3.6. *Power Management*

For this first implementation we are doing one technological demonstrator. We just try to show the capability of the project and we are not going to optimize the systems. For doing this first approximation to the final implementation we supply the Arduino with a 9V cell instead of one Li-Po Battery. This cell ensures us enough power and provides enough autonomy to the system for do several tests.

3.4. **Wireless Sensor Network implementation and programing**

Once we have seen all the background theory of the project, is time to show the final scenario of the project. This scenario has been tested.

As we could saw in chapter 1, the recommended density of the nodes for surveillance is around 1 node/ha or 1 node every 1-2km. For both cases, we cannot ensure communication between the ground nodes, so we must use a hybrid network, between star topology and peer-to-peer topology. The ground nodes do not be able to communicate between themselves, and only can speak with the drone node. The node carried on the drone is travelling all the time. It just sees one ground node, and passed this ground node, they establish communication with other.

The communication between multiple nodes at the same time is not required, so, we decide to use the AT Mode of the XBees for simplicity. All the ground nodes are programed as ZB ROUTER AT, and the drone node is programed as ZB COORDINATOR AT. The Coordinator node is the unique device type mandatory for create the network and is also capable to add devices in the network, so we must use the Coordinator in the drone, to ensure communication between the nodes, once we don’t have communication all the time with all the nodes.

So, for ensure communication between nodes we programed it with the X-CTU software. You can find lots of tutorials for detect and program your XBee devices, for example in [1], but here you can find the most important parameters in order to establish communication between them:

- PAN ID: you must choose the same PAN ID for each node of the network. This parameter is the key for create the network between the devices. The coordinator creates the network with all the nodes that have the same PAN ID. For our project we select the 1001 PAN ID. We can choose one number between 0 and 65535, but this number must be the same for all the network's devices.

- Address: we must add the address of the receiver device, but we have one network, where multiple devices take part, so we cannot put a unique address to transfer data in the coordinator device, so, this device talks and interchange data with multiple devices. For this case we have different options. We have to add the "DH" (Direction high) and "DL" (Direction low). These two parts of the address can be founded in the back of the device, but we don't need this number in order to establish communication in our case: we have the following option: For the Routers, we can change the address of the coordinator with zeros, the device recognizes this two zeros as the Coordinator address, so we just have to program the DL=0 and DH=0 for routers. The coordinator is the unique device that talks with multiple devices and we just can add one unique destination address, but if we put DH=0 and DL=FFFF, we can establish communication broadcast, so, all the devices can receive the packets that we send and we are able to receive packets from all devices.

We send the information via Serial.write() and we read the packets via Serial.read(). The information transferred had 2 parts: the number of palm, and the number of weevils in the tramp. We must take into account that we just can sent packets of information of maximum 8 bits length. So, every parameter higher to 8 bits must be divided in 1 byte packets before sent.

The first parameter is transferred as an 8-bit integer. This allows us to have 256 different palms ID. And the second one is a 16-bit integer, because we want to compute maximum 1000 weevils. The second parameter is divided in two packets. First of all, we sent the 8 higher bits and later we sent the 8 lower bits.
The higher bits are sent by displacing the higher bits to the low bits, and when we sent it via Serial, they just sent the last 8 bits. We use a char type variable, because this kind of variable type just allows store one byte:

```
int weevils=111;
  char b;
  char c;
  char palm=1;
  b=weevils>>8; //Example 0001010010010011 --> 0000000000010100
  c=weevils;
```

```
Serial.write(b);
Serial.write(c);
```

Serial.write sent the data using binary code, but we has been splitted the number. When we receive we must the number we must take into account that we have to reassemble the number again. So we multiply the higher byte to 256 and we add the lower byte.

```
b1=Serial.read();
b2=Serial.read();
rec=b1*256+b2;
```

This is not the unique way to send information via Serial. Other options are Serial.print() and Aerial.Println(), but you must take into account that this two functions use ASCII code to transmit data. If we send the information via Serial.println() we sent two different numbers before the information, because this function prints the package information in a new line text in the serial port receiver, and this change of line add two parameters in ASCII code (corresponding to numbers 10 and 13).

The packets received are stored in the EEPROM memory of the Arduino Mega. This Arduino has a higher EEPROM Memory capacity, for this reason, we use it in the Drone node, for have more memory capacity than the Arduino UNO. The total EEPROM capacity for Arduino Mega is 4KB, and our data is stored in one array, where each column is the position corresponding to palm ID number. For example, the palm number 1 has the position 0 in the array, and the palm 18, has the position 17. Taking into account that we store a 2 byte number for each column of the array, we can store a 2000 column array.

We must take into account that the number of writes/erase cycles for the Arduino EEPROM memory is limited to 100.000 cycles. For this reason, we add a button in the Arduino MEGA shield. We only write into the EEPROM memory when this button is pushed, once the drone, has completed the flight.

The Routers devices are always waiting for the coordinator, and never sleep. Just the End Devices have the capacity of put the node into different sleep modes, but in our case we have several problems working with End Devices and we decide to use the XBees as Routers. We must remember that this project is a technological demonstrator; we just have to demonstrate that the project is feasible; we do not try to optimize every part of the network or the nodes.

In order to avoid desynchronization in data transfer we add to premises to the code:

- The routers just sent the information when they receive information via Serial.
- The coordinator is always sending information via serial, and they just read when they receive data via Serial.

Once we have the network created we assembly and weld the components of the conditioning system of the sensor's signal. We weld everything in the Wireless shield, once we have different solder paths.

The tramp for the Weevils is has been done taking into account the different recommendations exposed in unit 1, where different studies obtain the best tramp model for a higher trapping efficiency. In our case, the tramp is a wireless node, with a weight sensor. The tramp is made with a 5L bucket [2]. At the bottom of the tramp we add the weight balance, covered by one straight plate. This plate has smaller diameter than the bucket, to avoid friction between the plate and the walls.

The Wireless node is placed at the top of the bucket. We add one plate, joined to the top of the bucket. This plate allows space to the wireless node and the battery and can support the wire with the food bait. This plate is joined to the cover, for make easy the maintenance of the trap and for easily handling. The cover of the bucket has one small hole in order to pass of the antenna of the XBee and ensure, as much as possible, the line of sight between XBees and increase the scope of the signal. Remember that the scope is 100m with line of sight and 30 meters without line of sight.

3.5. **Drone**

One important part for the implementation of the project is the Drone technology. This allows us a fast process for the information reuptake. In this chapter you can find all the information about the drone techonolgies, the types and the performances.

3.5.1. *Definition*

The drones or Unmanned Aerial Vehicles (UAV's) are aerial systems remotely controlled. These systems born from military purposes and nowadays the drone technology is used for civil purposes too. Thanks to his high customization capacity, the drones have a very high potential: they can be readapted for work in different environment conditions and for many different finalities. In recent years, the use of unmanned aerial vehicles has been on rise, and the industry is projected to increase very fast in the following decades. Every day we discover new ways to use the UAV's in order to make some works easily or improving the current methods. In this project we do exactly this: we use the drone technology in order to improve the way to do one task. [35]

3.5.2. *Types*

Depending of the purpose of our work we have different drone's performances and characteristics. In our case we are working for civil purposes and we can classify the types of drones by:

- Multi-rotor: This kind of drones is composed by 3 or more motors with propellers. They can turn, and change altitude and direction by regulating the velocity of the rotors. So they are similar to a helicopter..

  They are usually used for applications like: audio-visual, agriculture, construction or for industrial inspection. Their cruising speed is slow and the flight time of the drone is small, due to the consume of the motors is high. They can land and take off in small areas, have low wind resistance and the carrying payload can be high, depending of the type and number of rotors. In Fig. 3.10 you can see an example of multi-rotor drone, with some of their components.

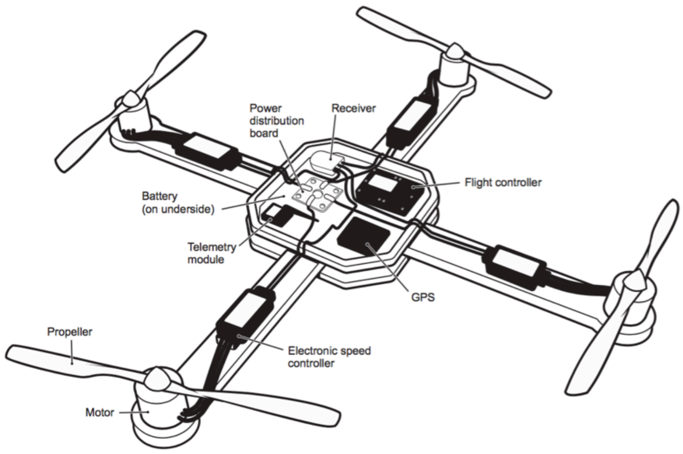

Fig. 3.10 Multi-rotor drone [35]

- Fixed Wing: This kind of drones is composed by just one motor with a propeller. They can turn, change altitude and direction by regulating the velocity of the rotor and thanks to its wings, so they are very similar to an aircraft.

  They are usually used for applications like: land surveying, agriculture, humanitarian or for environmental inspection. Their cruising speed is high and the flight time of the drone is high, because we just use one rotor and its electrical consume is low. They need a large area in order to take off or land and have high wind resistance but the carrying payload is very low. In Fig. 3.11 you can see an example of fixed wing drone, with some of their components.

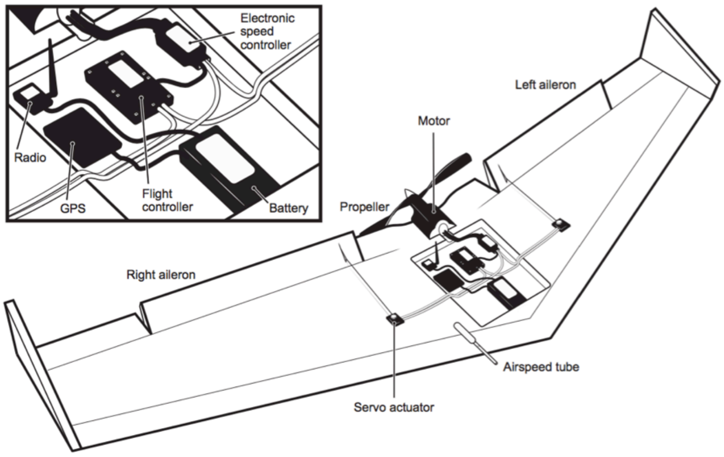

Fig. 3.11 Fixed Wing drone [35]

After knowing all the advantages and disadvantages, we can conclude that the selected platform for this project must be a fixed-wing drone. In this project will be need cover as much area as we can, and the sensor carried on is very light. So in order to have the maximum time of flight and cover the maximum area we must choose a fixed wing drone.

3.5.3. *Tecnology*

The drone technology is a simple but very powerful technology. The brain of one drone is the controller. This controller receives all the inputs and actuates depending on these inputs. The outputs are mainly sent to the rotors. The controller is programmed to allow different functionalities and flight types, for example, an UAV can being controlled using a remote control system based on a RF controller or can fly autonomous. We can program the route and the drone can follow it thanks to the GPS system.

The different inputs that enter to the controller are a GPS signal, provided by a GPS sensor, the RF commands, received from the RF receiver and other sensors inputs that we would incorporate.

The outputs from the controller usually go to the rotor, but controller can control other systems at the same time, like cameras and the camera stabilizers systems. The outputs that go to the rotors must previously pass through the Electronic Speed Controller (ESC's). This device changes the voltage to Pulse Width Modulation (PWM), a type of analogic signal, used to control the motors.
All of this systems and devices are powered by a Li-Po battery. The number of battery cells (2-4-6…) needed depends on the type of the rotors.

3.5.4. *Hemav Fixed Wings Drones*

To bring out the project it has been used fixed wing drones provided by HEMAV SL. HEMAV is a company that designs and commercialize a high added value solutions for industry, thanks to his high capacity for the technical process of the information provided from the drone (RPAS) technology. HEMAV is aimed to develop civil solutions using of UAVs. As we said in section

The current drones are resumed in this section:

- X5: This fixed wing drone is removable drone with and exoskeleton made of EPO foam and with a main crossbar made of carbon fiber. This aircraft can flight with manual mode flight or autonomously.

Table 3.7 Tecnical Specifications for X5 Drone [36]

Technical Specifications	
Wingspan	1180 mm
Endurance (4000 mAh)	40 minutes
Flight Range	N/A
Take Off Speed	N/A
Maximum Cruise Speed	30 m/s
Maximum Take Off Mass	1500 g
Operative Empty Mass	1250 g
Maximum Payload	250 g

- X6-HP1: This fixed wing drone is removable drone with and exoskeleton made of EPO foam and with a main crossbar made of carbon fiber. This aircraft can flight with manual mode flight or autonomously.

Table 3.8 Tecnical Specifications for X6-HP1 Drone [36]

Technical Specifications	
Wingspan	1180 mm
Endurance (5000 mAh)	45 minutes
Flight Range (VLOS)	1,5 km
Take Off Speed	20 m/s
Maximum Cruise Speed	25 m/s
Maximum Take Off Mass	3000 g
Operative Empty Mass	1926 g
Maximum Payload	1074 g

- Skywalker X8: This fixed wing drone is removable drone with and exoskeleton made of EPO foam and with a main crossbar made of carbon fiber. This aircraft can flight with manual mode flight or autonomously.

Table 3.9 Tecnical Specifications for Skywalker X8 Drone [36]

Technical Specifications	
Wingspan	2120 mm
Endurance (N/A battery)	60 minutes
Flight Range (BVLOS)	10 km
Take Off Speed	7 m/s
Maximum Cruise Speed	30 m/s
Maximum Take Off Mass	3500 g
Operative Empty Mass	1950 g
Maximum Payload	1550 g

We must take care that the endurance and the flight range are parameters that may vary depending on the capacity of the battery, the motor-propeller type and configuration. The specifications included in this document are provided by Hemav, and are taken from different tests made by the company. [2]

4. VALIDATION

Finally, we validate the function of all the system, with different experiments and test to ensure the desired specifications.

The **Arduino code** for the different Routers and the coordinator **works correctly**. The Routers send the information only when the Coordinator requires it and the routers send the packages in the correct order and timing, ensuring correct communication and a good data transfer. The Coordinator can talk with all the Routers and the Routers are able to talk with the Coordinator. **The network is created correctly.**

The **EEPROM Memory writes** the **information** when we push the button and we are capable to read the information using another program. All the codes used for Arduino’s Programming, are available in Annex.

The theoretical range for the Xbee device is 100m in outdoor and 30m indoor conditions. We have been done some test to obtain the real range. We realize a test with a LED. This LED is on when the Arduino can read information provided from the other Arduino, and is off when the Arduino is out of the range. At the same time we use this test to know which type of radiation pattern have the Xbee modules.

The results are the following:

Table 4.1 Experimental range for the Xbee devices

Indoor/outdoor	Indoor	Outdoor
Range	24,7 m	12,7 m

This experimental data for the outdoor conditions, **does not adjust to the theoretical ranges**, where the range is higher that Indoor. The maximum range is the theoretical case when we transmit with the maximum power. As a default parameter, the **output power is the minimum**, in order to consume the minimum. In this case we transmit a power around 0 dBm, that is 1-1,25 mW. We can change this parameter with the XCTU software. During this test have been obtained the radiation pattern for the Xbee antennas. **The antennas for the Xbee are omnidirectional for the horizontal plane,** so for ensuring the maximum range we must have Line-of-Sight communication with the horizontal plate of the antenna. If we realize the same test with one antenna in the horizontal plane and the other with the other antenna in his axial plane, we reduce this range to 6,7 meters.

This is a trade off of the RF systems: for obtain higher range, we can send signals with higher power or we can use a more sensibility antenna or with higher directivity, but always we have a higher consume. In this case, we want the lowest consume, for obtain the higher lifetime of the sensor node. For this project we don't need to change this parameter, because have enough range for ensuring communication. The drone do not need higher ranges, because it can fly with security with a 10 meters distance between drone and obstacle. The date palm is the most common palm is Saudi Arabia, and the favourite palm for the RPW, can be 23 m tall, and the drone just needs to fly 10 meters above the ground.

In the following figure you can se the **total distance where the drone node can establish communication** with the ground node for both cases:

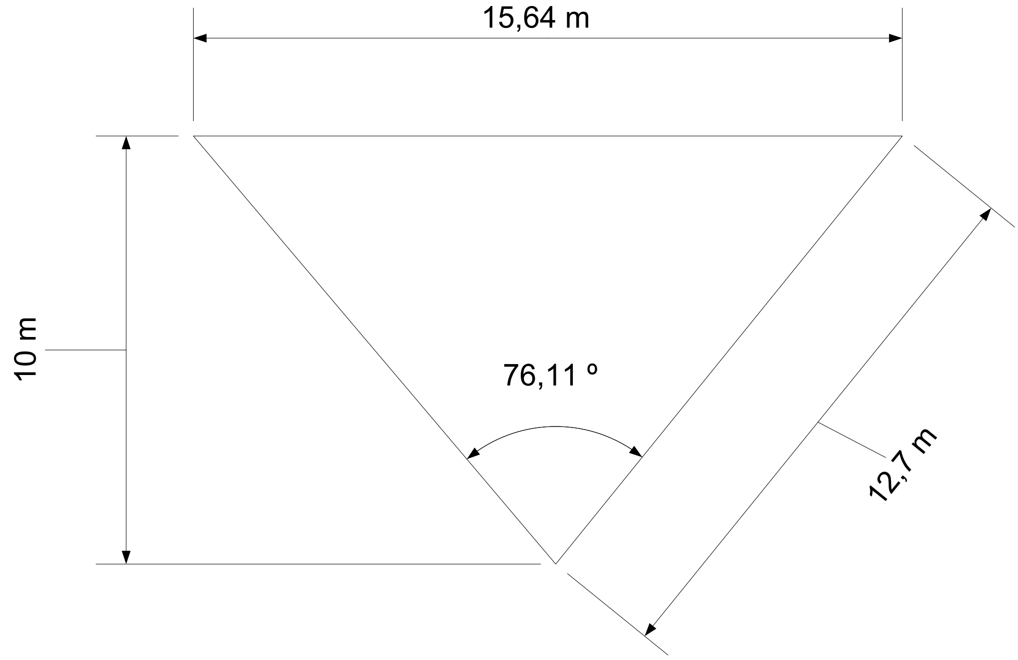

Fig. 4.1 Flight distances for coverage, FL 10 meters

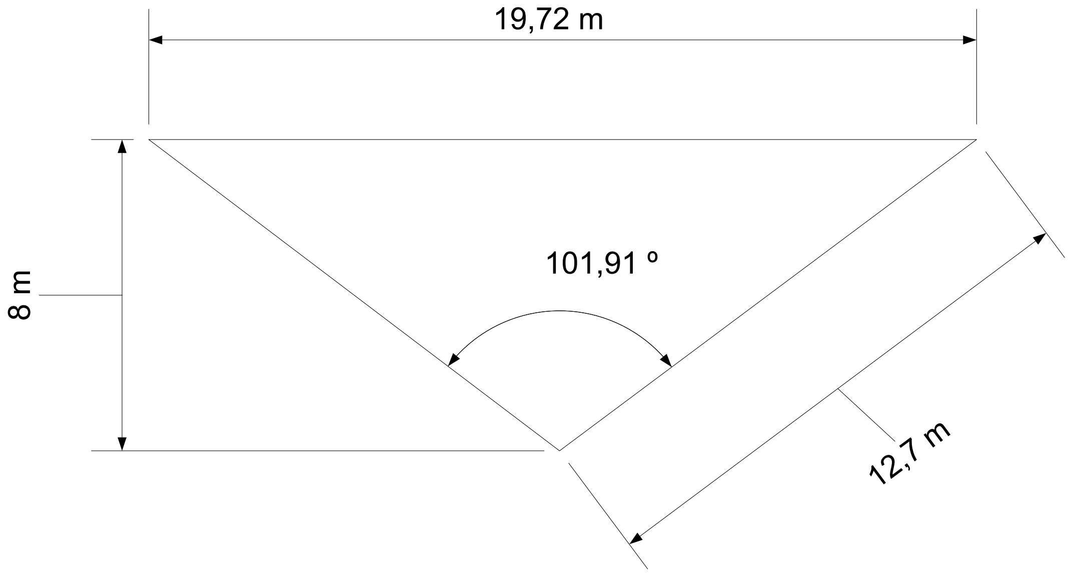

Fig. 4.2 Flight distances for coverage, FL 8 meters

The study has been done for two different flight levels. In the case for a flight level of 10 meters, the distance with coverage is 15,64 meters, and for the second case, where the flight level is 8 meters above the ground node, we can obtain 19,72 meters with coverage.

The time to begin the transmission, once the mobile node enters in the coverage zone of the ground node is 1 second, and the **experimental data throughput is 84.000 b/s**. Taking into account that we transmit 3 char variables plus another char for begin the communication, and each char variable has 8 bits. We want to transmit 24 bits, so we just need 0,2 ms in order to transmit all the information packages.

At the same time, we use another char variable to establish communication, so the total time required for the transmission is 0,38 ms. The **total time required** for communication and **data transfer** is 1,00038 seconds.

Once we have the time sequired for the data transfer, we just have to adapt the drone velocity. This table resumes the maximum velocity for each case:

Table 4.2 Maximum Cruise velocity for ensuring communication

Flight Level	10	8
Velocity (m/s)	15,63	19,71

For both cases, the drone velocity is assumable, and we don't need higher range for establish communication. But if higher ranges are required we can change the RF device. Other options for wireless communication are available, but the technology that ensures higher range is **LoRa**. This technology ensures a **range around 16 km** with line-of-sight, but with a low data throughput, around **0,3 – 3,5 Kbits/s**, but this is still enough for our application. This type of device allows us to connect to the same network all the devices, at the same time, and they can speak to each one, so the drone's utility is reduced, we could do a **wireless network without a mobile device.**

The tramp is adapted with the characteristics exposed in chapter 1.1.3, but the resolution of the sensor is lower to the desired one. Primary, we wanted to obtain la resolution to 1 gram, because this is the approximated weight of 1 weevil, but the **real resolution** is of **2 grams.** The system is able to known the number of weevils wit a resolution of 2 weevils. In order to **improve** this **resolution**, we have two options: We can **change the voltage reference** of the **ADC**, or we can **change the amplifier**.

The problem is that we do not use the **full scale range** of the **ADC**, because the Vref of the Arduino is **predefined** to **5 V** and the amplifier saturates at 3,5 Volts. The Arduino board has one pin called AREF, where you can change the voltage reference. If change the **voltage reference** to **3,3 V** we can use a higher scale range for our ADC, improving the resolution. Doing this change we obtain the following results:

Table 4.3 Final resolution

Node	V sensor (0 weight) *Gain	Bits (0 weight) *Gain	V sensor (1kg) *Gain	Bits (1kg) *Gain	Real Resolution (weevils/bit)
1	0,345 V	104	2,584 V	797	1,44
2	1,163 V	436	3,375 V	929	2,02
3	0,302 V	90	2,662 V	819	1,37

As you can see, we **improve the resolution**, and now we have higher resolution. Now we are able to differentiate changes of **1,5 grams**. We do not have enough, because we cannot adapt the 1 kg voltage to 3,3 volts. For this reason we need to change the amplifier in order to obtain higher gain.

The different electronic components will be able to put up with the meteorological conditions. The Arduino, Xbee, resistances, capacitors and amplifier can work with temperatures higher than 40 ºC. If we talk about the electrical junction, we are not always transmitting data, and the power used to transmit is low, so, the **temperature** of the system is **not a limiting factor** of the project.

Finally we can conclude that the project accomplishes all the requirements proposed in the initial chapter.

5. CONCLUSIONS

Wireless Sensor Networks are a very powerful technology, with multiple utilities, and for this project is a **feasible solution**. With this method we improve the actual surveillance method, making it faster. This type of networks allows **multiple solutions and configurations**. If the problem varies, we can readapt the solution to the new conditions. In this case we use **hybrid network**, between a peer-to-peer and a star topology.

The **ZigBee** communication protocol is a **perfect solution** for the project. Is a easy technology with multiple capabilities in order to adapt the performance of the device and the network to the project requirements. In our case we adapt the Xbee devices for establish communication with the Coordinator and we decide to program it in AT mode for simplicity, but exist to many options. For this project we send information via Serial, but Xbee allows libraries for send packets in a better way, with responses when the information is received correctly.

Control the number of weevils, **measuring the weight** of the tramp is a **feasible solution**, once we have enough resolution. At the same time is the easier and cheapest solution.

The **Drone** is an **excellent device to create a network** and permit us to change the network. At the same time allows us to have a network with the ground nodes far away from the other ground nodes and **ensures communication and data transfer between nodes without coverage.**

The **network** and the **counting solution** is **compatible** with **other wireless protocols and applications**. By using the Arduino Hardware, we can change the device, because some of them are **pin-to-pin compatibles**, like Bluetooth. Other ones use different shields, compatibles with the Arduino board, and they can **send data via Serial.** So we can use the **same network** type **with other protocols and devices**, readapting the solution with the current necessities.

At the same time we are able to **incorporate other sensors**, like **temperature** sensors in the palm tree to detect the health of the palm before the infestation affect the palm. So, the method can be changed, instead of weevils, we can control the **palms physiognomy**, **health** or so many other parameters. The created infrastructure can be used for **multiple purposes.**

The **future works** before the real implementation of the projects are aimed to improvements in consume of the devices, a final design for the tramp and try to program the Xbee with the existing libraries.

Now, the nodes that contain a sensing unit are programed like a Router. With this type of device, we cannot **put the Xbee into Sleep Mode**. This function is only available for the **End Devices**. This increases the consume of the node, with a reduction of the battery lifetime. With the actual XCTU software we had lots of problems in order to program the devices in End Device mode, so you will have to use the previous version that is more robust again different errors and compatibilities. The Xbee has different **libraries for Arduino's software**. These are open code software, so, for future improvements, we must be careful when we use it, because we can find some problems not discovered yet. This software are available for the API mode and for Java programming. This avoids to send information via Serial, and this libraries allow the user to know if the information package arrives to the desired destination.

The actual **Power system** is a simple battery, used for the technological demonstrator. For the Real implementation we will need to change the battery, and add a **recharge system**. We must choice the **primary battery** and the **secondary**. The theory of this part can be found in section 2.1.

Improve the **trap**. The entrance for the weevils must be covered to **avoid** the **sand** of the desert to enter inside, but must allow the weevils to enter with facilities. This sand can produce a **big error in the measure**. The sand that enters can fell into the space between the floor of the tramp and plate where there are weevils. This big space can be for sand store. The plate must have fewer diameter than the tramp. If the diameter is the same we have friction between the plate and the walls and we don't measure correctly the number of weevils and the sand is placed in the plate.

For obtain the **resolution desired** we must **change** the **amplifier**, to obtain a output signal that uses the **full scale range** of the Arduino's **ADC**, or **change** the **Voltage reference** for the **ADC**. The resolution is enough for the final objective, but the signal from the strain gauges fluctuates. For avoid this we must change the gauge for another with more stability. We must take into account that the gauges used are low cost and aimed to measure with a resolution of 5 grams.

Every time that we work with Arduino, we must take into account the **electromagnetic compatibility.** For avoid interferences, all the **pins** of the Arduino programed as **Inputs**, has to be **turned off**. In the desert you do not have interferences from other signals or near systems, but you can create an auto-interference and the Arduino board is susceptible to have different couplers.

BIBLIOGRAPHY

[1] Giblin-Davis R. M., Romero Faleiro, J., Jacas, J. A., Peña, J. E. and Vidyasagar, P.S.P.V, "Biology and Management of the Red Palm Weevil, Rhynchophorus ferrugineus", *CAB International,* 1-16, (2013)

[2] Murphy S. T. And Briscoe, B. R., "The red palm weevil as an alien invasive: biology and the prospects for biological control as a component of the IPM", *Biocontrol News and Information*, Vol. 20 No. 1, 8-9 (1999)

[3]Karnataka, J., "Studies on the biological aspects of red palm weevil, Rhynchophorus ferrugineus(Oliv.)" *J. Agric. Sci*., 22(3), 732-733 (2009)

[4] El-Faki Mozib, M. and El-Shafie H. A. "Effect of the red palm weevil, Rhynchophorus ferrugineus(Olivier) infestation on temperatura profiles of date palm tree", *Journal of Entomology and Nematology,* Vol 5(6), 77-83, (2013)

[5] Datasheet report for Rhynchophorus ferrugineus (red palm weevil) www.cabi.org/isc/datasheet/47472

[6] Kaakeh, W., El-Ezaby F., Aboul-Nour, M. M., Khamis, A. A., "Management of the Red Palm Weevil, Rhynchophorus ferrugineus(Oliv.) by a Pheromone/Food-Based Trapping System" *Department of Plant Production, United Arab Emirates University*, 325-340

[7] Zohraby, K., Minoli, D. and Znati T., "*Wireless Sensor Networks: Technology, Protocols and Applications*", John Wiley and Sons, New Jersey, (2007)

[8] Kemapech, I., Duncan, I. and Miller, A., "A Survey of Wireless Sensor Networks Technology", *School of Computer Science St Andrews*, 1-3, (2002)

[9] I. Akyildiz y Su, W., "Wireless Sensor Networks: a survey", *Computer Networks,* no 38, pp. 393-400, (2001)

[10] Erazo Villegas M. A., Qian, Y. and Seok Yee, T., "Wireless Sensor Network Communication Architecture for Wide-Area Large Scale Soil Moisture Estimation and Wetlands Monitoring", Network Communications Infrastructure Group Department of Electrical and Computer Engineering University of Puerto Rico, 1-5.

[11] Frechette, L., Rabaey, J., Roundy, S. and Wright, P., "Sources for Wireless Sensor Networks", Cap 1 in "*Wireless Sensor Network*", Karl H., Willing, A. and Wolisz, A., 1-18, Springer, Berlin, 2004

[12] Alabesa, J., Gasulla, M. And Penella M. T., "Powering Wireless Sensor Nodes: Primary Batteries versus Energy Harvesting", *Instrumentation, Sensors and Interfaces Group, Electronic Engineering Dept. Escola Politècnica Superior de Castelldefels*, 1-2, 2009.

[13] Rodríguez Pallares, J.E, “Wireless Sensor Network Implementation with Arduino and Xbee”, *Escola d’Enginyeria de Telecomunicació i Aeroespacial de Castelldefels*, 4-23, 2015.

[14] Kosmerchock, S., “Wireless Sensor Network Topologies” 1-3.

[15] Gaia Zanchu, M., “Bluetooth Low Energy”, *Litepoint Whitepaper*, Vol. 1,1-3 ,2012

[16] A. Jayan and V. Abinayaa, “Case Study on Comparison of Wireless Technologies in Industrial Applications*”, International Journal of Scientific and Research Publications,* vol. 4, no 2, 2014.

[17] Smith, P., “Comparing Low-Power Wireless Technologies”, *Digi-Key Electronics*, 2011.

[18] Rabbit, *“An Introduction to Wi-Fi, Product Manual”*, Digi International, USA 2008

[19] Gislason, D. *“Zigbee Wireless Networking”,* Newnes, 2002

[20] Fisk, D. and Sullivan, D., “Optical People counting for Demand Controlled Ventilation: A Pilot Study of Counter Performance”, *PIER Subject Area: Buildings End Use Energy Efficiency Program*, 2-6, 2009

[21] “Encoder Application Handbook”, *Danaher Industrial Controls*, 2-5,

[22] Moreno, I., “Sensores resistivos y su acondicionamiento”, *Universidad de Burgos, Area de Tecnología Electrónica*, pp 10-23.

[23] https://www.arduino.cc/en/Main/ArduinoBoardUno

[24] https://www.arduino.cc/en/Main/ArduinoBoardMega2560

[25] https://www.arduino.cc/en/Main/ArduinoWirelessProtoShield

[26] https://www.arduino.cc/en/Guide/Environment

[27] Simoneau. P., “The OSI Model: Understanding the Seven Layers of Computer Networks”, Global Knowledge, 1-6, (2006)

[28] D. GISLASON “Zigbee Wireless Networking”, Newnes.

[29] Somani. N.A. and Patel. Y., “Zigbee: a low power wireless technology for industrial applications”, *International Journal of Control Theory and Computer Modelling (IJCTCM),* Vol.2, No.3, 1-7, (2012)

[30] Tomar. A., “Introduction to Zigbee Technology”, *Global Technology Centre*, Volume1, 1-15, (2011)

[31] Digi International, “ZigBee RF Modules: XBEE2, XBEEPRO2, PRO S2B. User Guide”, 2015.

[32] https://www.sparkfun.com/products/11812

[33] http://www.ftdichip.com/FTDrivers.htm

[34] http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

[35] Kakes, K., Greenwood, F., Lippincott, M., Dosemagen, S., Meier P. and Wich, S., *“Drones and aerial observation: new technologies for property rights, human rights and global development”,* New America, 2015

[36] Fixed Wing Drones Specifications, HEMAV SL.

ANNEX

Routers Code:

```
int sensorPin=A0; //pin where the sensor is connected

void setup() {
Serial.begin(9600);
analogReference(EXTERNAL); //We use an analog reference of 3.3V instead of
5V
}

void loop() {
  while(Serial.available() >0)
  {
  int sensorValue = analogRead(sensorPin); // read the value from the sensor
  int numberofbits=sensorValue-104;//number of bits of initial offset
  int weevils=numberofbits*1.44; //with the sensibility we change the bits to
weevils
  char b;
  char c;
  char palm=1;//ID of the palm
  b=weevils>>8;//high part of the int number
  c=weevils;//low part of the int number
    Serial.write(b);
    Serial.write(c);
    Serial.write(palm);
    delay(1000);
  }
}
```

Coordinator Code

```
#include <EEPROMWriteAnything.h>
int rec;
int numpalm;
char palm;
char b1;
char b2;
int information [3];//Array where the information is stored, the lenght is changed
when the number of nodes in the network vary
const int buttonPin=2;
const int ledPin=13;
```

```
int buttonState=0;

void setup() {
Serial.begin(9600);//Remember that the baud must be the same on both
arduinos
pinMode(ledPin,OUTPUT);
pinMode(buttonPin,INPUT);
}
void loop() {
Serial.write("Hello");//He sends always a message, for beggin the
communication
buttonState=digitalRead(buttonPin);
digitalWrite(ledPin,LOW);
while(Serial.available() > 0) {//While he recieves information....
        Serial.write("Hello");//The Router send information if he recieves
information via serial!
         b1=Serial.read();
         Serial.println(b1, BIN);// For test, prints the information in Serial
         b2=Serial.read();
         Serial.println(b2, BIN);// For test, prints the information in Serial
         palm=Serial.read();
         Serial.println(palm, BIN);// For test, prints the information in Serial

        rec=b1*256+b2;// reensamble the number of weevils
        numpalm=palm-1;
        Serial.print("número d'escarabats: ");// For test, prints the information in
Serial
        Serial.println(rec);// For test, prints the information in Serial
        Serial.print("número de palmera: ");// For test, prints the information in
Serial
        Serial.println(palm);// For test, prints the information in Serial

        information[numpalm]=rec;
        Serial.println(information[numpalm]);// For test, prints the information in
Serial
delay(1000);
}
if(buttonState==HIGH){//When the button is pushed...
  digitalWrite(ledPin, HIGH);//We turn on the LED, for ensure that we enter into
this IF
  EEPROM_writeAnything(0, information);//We write the information in the
EEPROM Memory
}
}
```

Read EEPROM Memory Code:

```
#include <EEPROMWriteAnything.h>
int information [3];
void setup() {// put your setup code here, to run once:
 Serial.begin(9600);
 EEPROM_readAnything(0, information);// We read the information just one time
}

void loop() {// put your main code here, to run repeatedly:
Serial.print("Dada 1:  ");
Serial.println(information[0]);//We print the information
Serial.print("Dada 2:  ");
Serial.println(information[1]);//We print the information
Serial.print("Dada 3:  ");
Serial.println(information[2]);//We print the information

delay(8000);
}
```