Bird Patrol - Using birds as sensing units

Rodrigo da Silva Freitas Rocha

Thesis to obtain the Master of Science Degree in

Telecommunications and Informatics Engineering

Supervisors: Prof. Duarte Nuno Jardim Nunes
Prof. Marko Radeta

Examination Committee

Chairperson: Prof. Ricardo Jorge Fernandes Chaves
Supervisor: Prof. Duarte Nuno Jardim Nunes
Member of the Committee: Prof. Hugo Miguel Aleixo Albuquerque Nicolau

November 2019

Acknowledgments

I would like to thank my parents for their friendship, encouragement, patience, support, and care over all these years, for always being there for me through thick and thin and without whom this project would not be possible at all, for education is the basis of an individual in society.

To my grandmother for her patience, friendship and her kind words and education throughout life.

To my grandparents and my uncle which were not able to see me becoming an engineer, I'm sorry I wasn't able to complete the journey on time, however, I know you would be so very proud to witness this moment, and I will do my best to honor your memory.

To my sister, for the ongoing encouragement to proceed with my education, and for always being able to find a way to motivate me and above all for being a constant inspiration in my life.

To my brother and his family, for their friendship throughout all these years may we set each other back on track when needed for many more years to come.

To my girlfriend for her patience, friendship, love, affect and utmost dedication throughout this journey. May the future years shine on us

To Eng. Jorge Lopes for his competence and thorough analysis of problems and above all, for his friendship, dedication and ongoing advisement throughout this journey.

To the future Ph.D.'s Dinarte Vasconcelos and Miguel Ribeiro for their support and ongoing advisement, actively helping me with the tests carried on, may you have outstanding careers in the field.

I would also like to acknowledge my dissertation supervisors Prof. Nuno Jardim Nunes and Prof. Marko Radeta for the possibility of being part of M-ITI and also for their valuables insights, support, and sharing of knowledge that has made this Thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and are always there for me during the good and bad times in the journey of life. To each and every one of you, who without your contribution this work wouldn't be possible, for that I'm grateful and from the bottom of my heart,

Thank you all.

This study is part of LARGESCALE project with grant no. 32474 by Fundação para a Ciência e a Tecnologia (FCT) and Portuguese National Funds (PIDDAC). It is also supported by the funding from Projeto Estratégico UID/EEA/50009/2019 (ARDITI/LARSyS). It is also supported by Programa de Cooperación INTERREG V-A España-Portugal MAC (Madeira-Azores-Canarias) 2014-2020, throughout project INTERTAGUA (Interfaces Aquáticas Interativas para Deteção e Visualização da Megafauna Marinha Atlântica e Embarcacoes na Macaronésia usando Marcadores Rádio-transmissores), with Ref: MAC2/1.1.a /385. Also, this thesis has been supported by the regional funding from MITIExcell - EXCELENCIA INTERNACIONAL DE IDTI NAS TIC (Project Number M1420-01-01450FEDER0000002), and project SUBMARINE (Simulators for Understanding the Biodiversity using Mobile and Augmented Reality as Interactive Nautical Explorations), provided by the Regional Government of Madeira.

Abstract

While global warming remains a pertaining problem, scientists are ever more interested in understanding the human impact on the ecosystem. Current technologies applied in ecology for understanding and sensing the marine species remain costly, providing opportunities for low-cost microcontrollers and worldwide IoT communities. This study provides the experimental apparatus of a bio-tag, capable of collecting the environmental telemetry, intended to be used on marine fauna focusing on seabirds. A contribution is in providing the feasibility study, using the remote sensing through LoRa protocol and modeling the geolocation position while maintaining a competitive frequency of message transmissions when compared with technologies employed nowadays.

Keywords

IoT; Wildlife Monitoring; Ecology; LoRaWAN; LoPy; Bio-telemetry; RSSI; Ranging; Multilateration; Battery optimization;

Resumo

Embora o aquecimento global continue sendo um problema pertinente, os especialistas estão cada vez mais interessados em entender o impacto humano no ecossistema. As tecnologias atuais aplicadas em ecologia para entender e detectar as espécies marinhas permanecem muito caras, oferecendo oportunidades para microcontroladores de baixo custo e comunidades de IoT em todo o mundo. Este estudo fornece um aparato experimental de uma biotag, capaz de captar telemetria ambiental, destinada a ser utilizada na fauna marinha com foco em aves marinhas. A contribuição está na execução do estudo de viabilidade, usando o sensores remotos através do protocolo LoRa estimando a sua localização, mantendo tanto a longevidade como uma frequência competitiva de transmissão de mensagens quando comparada às tecnologias empregues atualmente.

Palavras Chave

IoT; Monitorização de vida selvagem; LoRaWAN; LoPy; Bio-telemetria; RSSI; Multilateração; Eficiência de bateria;

Contents

List of Figures

List of Tables

Listings

Acronyms

ABP	Activation By Personalisation
AP	Access Point
AoA	Angle of Arrival
CSS	Chirp Spread Spectrum
DBPSK	Differential Binary Phase-Shift Keying
EKF	Extended Kalman Filter
FSK	Frequency-Shift Keying
GFSK	Gaussian Frequency-Shift Keying
GLS	Global Location Sensing
GPS	Global Positioning System
GSM	Global System for Mobile Communications
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
ISM	Industrial Scientific and Medical
IST	Instituto Superior Técnico
IoT	Internet of Things
LAN	Local Area Network
LC	Location Class
LIFO	Last In First Out

LPWAN	Low Power Wide Area Network
LoS	Line of Sight
M-ITI	Madeira Interactive Technologies Institute
MAC	Media Access Control
MCU	Microcontroller Unit
OTAA	Over the Air Activation
PHY	Physical Layer
PSAT	Pop-up Satellite Archival Tags
PTT	Platform Transmitter Terminals
QoS	Quality of Service
QoS	Quality of Service
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SD	Secure Digital
SPI	Serial Peripheral Interface
TDoA	Time Difference on Arrival
TTN	The Things Network
ToF	Time of Flight
UART	Universal Asynchronous Receiver/Transmitter
UHF	Ultra High Frenquency
UTM	Universal Transverse Mercator
VHF	Very High Frequecy
WAN	Wide Area Nework
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

1

Introduction

Contents

Recently, the viability and widespread of the Internet of Things (IoT) became possible along the years with the ongoing extension of Internet connectivity beyond standard devices. Nowadays, these devices are used mostly for sensing and actuating in urban environments, allowing them to shift the traditional non-internet enabled objects into microelectronics, communication and sensing units. The biggest contribution of these IoT devices is that they can be easily and remotely monitored and controlled while maintaining low-cost and low-power policies. These devices, represent the edge or end-nodes of the IoT ecosystem and fuel the idea of collaborative effort networks among nodes through production, transmission, and processing of the widest variety of environmental parameters. All of these, provide novel insights and more expanded knowledge in remote sensing, e.g. in safety applications measuring CO_2 levels [3], monitoring and understanding of crops [4], etc.

In the case of this thesis, the focus will be on seabirds and specifically, in the region of Madeira, as there are numerous endemic species of seabirds in that region (e.g. Madeiran Zino's Petrel[1]). This thesis contributes to building a prototype and a pilot study, paving the way for future use on local seabirds, with the purpose to collect more environmental data in this region.

1.1 Motivation

Up to the present time, technological improvements made during the last decades are ending what the industrial revolution began, creating economies based on information technology [5]. It is of common knowledge that the industrial era, along with the technology produced as a consequence of it, played a considerable part in changing the day-to-day lifestyle, while it is continuously increasing the human impact on nature on a global level (e.g, biodiversity loss, climate changes, deforestation, among others) [6].

The urgency of detecting and responding to this environmental discrepancy is one area of research with the extensive potential to the scientific communities around the world. While causing an impact on the way of living, collaborative networks continuously contribute to the production of data and monitoring the impact of the human footprint in the ecosystem. Moreover, in the region of Madeira, there are an estimated 45 species of birds[2] which fly over the oceans as well as urban environments. Also, pollution remains a big concern as more seabirds are globally threatened because of the human impacts both at sea and at breeding sites. Due to the Moorse law, as electronic components keep getting smaller and more accessible, more IoT devices are produced, this pilot study intends to use the accessibility of IoT coupled with seabirds. This study foresees the impact of novel monitoring tools which could help scientists, ornithologists, marine biologists, and other experts, to formulate a better view of the problem.

[1] https://www.madeirabirds.com/zinos-petrel-pterodroma-madeira

[2] http://aprenderamadeira.net/aves/

1.2 Research contributions and research questions

In this study, the usefulness of the seabirds to be used as bio-monitors is a strong point and it is important to refer that this hypothesis has been exploited in the past [7]. Why is it important to monitor sea birds? Sea birds due to their biology, maintain a close relationship not only with the environment in the mainland but also with the vast open sea. In summary, due to their conspicuous nature, seabirds reveal themselves as a very good choice as a monitor, as changes in variables like population size, breeding and survival rates may provide an alarm indicating a problem related to the health of our environment[3]. This thesis proposes a system to close monitor seabirds, while passively monitoring the surrounding environment. Since changes in the environment will surely affect sea birds first, it is of utmost importance to monitor, develop and deploy accessible tools for the better conservation of species and the identification of the problems mentioned that should be posteriorly addressed. In [8], a quantitative analysis directly relates seabird mortality with marine debris ingestion as 250 000 tonnes of marine debris float in the oceans which presents a significant threat to marine wildlife.

The close monitoring of the seabird population is an area of interest not just for ornithologists but as well as all the experts that will deal with the problems encountered by the studies in this area. However, ornithologists find themselves limited by constraints when trying to monitoring taxa through sensors, such as price or the flexibility of currently existing technologies. These sensors acquire data such as temperature, pressure, altitude, etc, which is data relevant to develop knowledge about the behavior of a species, but also to develop knowledge about the environment surrounding the species itself. As the current development in electronics as was mentioned and the current development of Low Power Wide Area Networks (LPWANs) to cope with the needs of a sensor network, has brought new possibilities in terms of price and flexibility of operations the questions that we pretend to answer with this study are the following:

1. **[RQ1].** Is it possible to advance the state of the art in software for tagging sea birds using existing technologies?
2. **[RQ2].** How does distance estimation behave in ocean settings?
3. **[RQ3].** What's the overall level of accuracy in position estimation that is possible to achieve with the system developed?

[3] http://seabirdyouth.org/seabird-science/

1.3 Objectives

The aforementioned provides an opportunity for collecting and understanding the concentration of noxious elements using seabirds, which might seem to be an appealing and reasonable idea. However, several constraints prevent the IoT application of low-cost sensors to fulfill this objective. Current sensory equipment remains at the high cost due to their complexity, while some require additional intensive calibration methods which precede an execution phase [9]. Another major challenge is the continuity and longevity of execution, with the purpose to gain the understanding of a given situation. In mobile and low power systems, such requirements are often not achievable. As sensor longevity remains a major restriction, it can lead to imprecise data extraction, as the amount of energy remains a very limited constraint.

Thus, this project poses a novel approach tackling the environmental issues from the seabird perspective. By studying the software of implementing and in the future, coupling a sensor to a seabird, it is possible not just to sense the local environment but also to map and classify species behavioral patterns, understanding their breeding, feeding habits and migration flows caused by humans. All of these can provide the opportunity to assess possible deviations from normal behaviors, caused by external human factors (e.g. marine litter pollution, noise pollution, etc). Leveraging this potential, the focus of this project is to study the feasibility of using remote sensing through the LoRa protocol. Taking into account three major requirements that were initially established being, (i) development of a prototype tag using a Microcontroller Unit (MCU); for (ii) low power and impact on battery autonomy; while (iii) sensing the wide variety of environmental parameters (temperature, luminosity, pressure, etc) using diverse sensory-based data and payloads.

1.4 Organization of the Document

This study is realized in cooperation with Instituto Superior Técnico (IST) and Madeira Interactive Technologies Institute (M-ITI), and is organized as follows: chapter 2 provides the related work in the ecology field as well as an overview and comparison in IoT technologies, chapter 3 presents an implemented solution for the system proposed. Chapter 4 presents the results obtained in the tests performed, as well as a thorough analysis of these, while chapter 5 concludes the study, outlining several limitations and future improvements.

2

Related Work

Contents

In the following chapter, it is presented to the reader, part of the related work that was considered relevant throughout this project. In the State of the art section, it is presented the several types of wildlife trackers used nowadays, as well as their characteristics in terms of power-management, weight, lifespan, range, location method, accuracy in the position obtained and the price of conducting a project of this nature. In section 2.1.3 it is presented more in-depth research about other location estimation techniques in radio transmission. Subsequently, it is also presented a section about the state of technology were the options in the market for off the shelf MCUs are explored. The available choice available in LPWANs and Wireless Local Area Networks (WLANs) is considered, as well as a set of considerations about radio wave propagation and propagation models for evaluating the presented results.

2.1 State of the art

Bio-loggers are devices transported by animal subjects to collect data. Applications of such devices are denominated and known as bio-logging [10]. However, when doing the literature survey, bio-logging differs from the definition of bio-telemetry [11]. Bio-loggers are referred to as devices with storage capacities to record measurements to create locally stored logs. These devices have limited or no transmission capabilities when affixed to the subject species. Moreover, their recorded logs are collected after the end of the experiment and are mostly used for laboratory processing. Bio-telemetry devices, on the contrary, provide experts with access to real-time data. These devices may have storage capacity, while their primary focus is the real-time transmission of the acquired data to a remote server. Nevertheless, both bio-logging and bio-telemetry are equally important in the realm of ecology. Both definitions may refer to the direct and indirect studies of the marine taxa. While the former focuses on their physiological or behavioral aspects, the latter deals with the impact of the surrounding environment on marine taxa. Moreover, they both focus on transportation, and that is why they are commonly referred to in the literature as bio-monitors [10]. Such devices may be affixed to the subject in numerous ways; (i) attached directly to the animal skin or carapace using glue (as in the case of sea turtles, e.g. Caretta caretta[1] commonly known visitor of Madeira); (ii) carried as collars or bracelets (as in the case of seabirds or with sea lions); and (iii) pierced to the animal, using harpoons (as commonly used with pilot-whales, or other cetaceans visiting the Madeira region or other vertebrates such as the hammer-head sharks). In this section, the focus will be instead of marine megafauna in the Macaronesian region, it will be on local seabirds while understanding the techniques and procedures of tagging the seabirds with bio-monitors.

First seabird tagging techniques dates to the early 1800s, when John James Audubon, a French-American ornithologist conducted the first known bird-banding experiment in North America. He found that a bird in the region would return to its nesting site after its migratory season (see fig. 2.1). This

[1] https://www.lobosonda.com/caretta-caretta-part-1/

coined the term and provided the meaning of philopatry [12]. Since that time [13], and with the increase of technology, sensing devices, and techniques in ecology, experts started to analyze larger corpora across different spatial and temporal scales. This provided the realistic data collection which would be otherwise very challenging or inappropriate to obtain in terms of the deployment environment. E.g, in [14], the study underpinned the general and constant circulation of elephant seals in southern oceans where researchers successfully deployed and collected the data from tagged bio-monitors.

Figure 2.1: Bird banded for experiment

2.1.1 Bio-logging Systems

Archival Tags are of most rudimentary forms of bio-logging systems [11]. As typical geolocation tags do not transmit data remotely, they do however require retrieval when the experiment is performed. This allows the successful retrieval of sensing logs from the on-board memory. Conversely, when applying bio-logging techniques to the sea birds, these tags provide valuable information in contrast to aforementioned tagging techniques, which were reported in [15], as being a very limited method and with ethical concerns. These limitations have been argued that it only depicts that a bird traveled from point A to B, without gaining insight about where and how their time has been spent in between. Location in archival tags is usually estimated with Global Location Sensing (GLS), however as studied in [16], the use of positions obtained through light-based estimation presents a mean error that can range significantly from 40 to 380 km for free-ranging animals (animals living outside of captivity). Such inaccuracies are due to passing storms or clouds and also due to large-scale movements that occur during their flights. Currently, the biggest advantage of data-loggers is the low-price that can go from US$ 12 to US$ 600, with devices weighing from 0.3 to 3.3 g. These devices are usually equipped with batteries providing longevity as low as 5 days to as high as 2 years [2]. Other considerations regarding the usage of archival

[2] http://www.migratetech.co.uk/

tags are the visibility of the device while being attached to the animal. Considering that this kind of system is planned to be successfully retrieved, these tags must be visible and identified with contacts and reward information (e.g. information provided by the local fisherman). This additional information should understand the geographical area where the tag may be located and must be made public via campaigns or any other channels available.

Although restricted to marine research, Pop-up Satellite Archival Tags (PSAT) remains a technology that stands between the aforementioned archival tags and the satellite systems that will be mentioned in the bio-telemetry section. Satellite and ground-based waves present high attenuation in marine research due to the presence of water, providing a big constraint. PSAT [17] is in its essence an archival tag with the buoyant add-on allowing the successful detachment from the studied marine taxa and allowing it to emerge at the ocean surface for successful retrieval. These detachments are programmed, and once these tags are at the sea surface, PSAT can relay data directly through the expensive ARGOS satellite system[3]. By employing this PSAT system, one can favor the simplicity of an archival tag, to the ease of transmission by satellite system, setting aside the problem of retrieving the device [18, 19].

2.1.2 Bio-telemetry Systems

In this section, it will be discussed the currently available bio-telemetry systems. In general, there are three types of distinct systems used in radio-telemetry [20]:

1. Very High Frequecy (VHF) radio tracking;
2. Satellite tracking;
3. Global Positioning System (GPS);

Radio-tracking uses a radio signal to provide information about the marine taxa to marine biologists. Typically, a radio-tracking system is any kind of communication system which allows transmitting the data using a sender and receiver. Thus, transmitting systems consist of: (i) a radio transmitter; (ii) power source; and (iii) a propagating antenna. On another hand, a receiving system includes (i) a power source, (ii) receiving antenna, and (iii) a signal receiver with the reception indicator. These two parts are commonly used in-situ, by tuning the transmitters to different frequencies and allowing the proper link identification between the sender and receiver. It's also important to reflect on which sender-receiver system is best suited for the diverse experiment [11]. Thomas [21] follows three criteria that can lead an expert to the best choice available to achieve adequate communication: (i) specification of the data required to meet the project; (ii) understanding the constraints imposed by the species and locations of the study; and (iii) calculation of the cost of the various tracking methods available. Based on these

[3] http://www.argos-system.org

criteria, one can evaluate the technologies to use that will provide a basis for the proposed solution in this study.

VHF

Regarding the ranges, located in a band between 30 to 300 MHz, VHF transmitters emit radio-frequency signals which are received by a receiving antenna with a ground-to-ground range of 5-10 km and an air-to-ground range of 15-25 km [22]. VHF enables animal tracking using two main methods, homing and triangulation. As explained in [23], the former technique involves following a signal towards its strength. This assumes that experts are in-situ, while they are tracking the signal using the receiving equipment. As experts are getting close to the transmitter, which translates in increasing signal strength, the receiver gain is successively reduced to get a better direction of the signal. This process undergoes constant iterations and is repeated until finally the subject of the experiment is located. Latter one, known as triangulation, is the process of two receiving antennas extracting two bearings[4] from the received signals. Sunny day scenario would be at angles of 90º, which would be plotted and intercept at transmitter location like in [24]. Logically, the accuracy of measurements increases as more bearings are extracted, consequently translating in more antennas in the range of the transmitter, which increases the costs of the system. In ornithology, this is presented as a low accuracy method, due to large-scale movements of birds which will introduce significant error in position estimation. However, VHF reports a precision between 200-600 m for the location estimation through triangulation and homing, and has proven effective in the studies of species with low levels of movement with few up-front costs and transmitters going from US$ 180 to US$ 300 [25]. When understanding their battery autonomy, these systems can rely primarily on lithium batteries [22] which have a longer life or can rely on a combination of solar cells plus rechargeable batteries ensuring 24 hours signal output until other components failure. However, power source consideration is a critical planning point, as heavier power-sources will increase the device lifespan while affecting the device weight. This weight should be properly assessed and the risks of infringing the 3-5% rule of transmitter weight [26] when deployed on taxa must be understood. In general, VHF transmitter can weigh as little as 0.2 g [27] with a lifespan of 18-22 days or can weigh as much as 100 g with a lifespan up to 4 years [5]. In any case, radio-tracking through VHF, demands the placement of the transmitter to be in a geographical area within the perimeter of deployed antenna, so that the signal is successfully transmitted, or by constantly following the signal in the case of mobile antennas (see fig. 2.2).

[4] A bearing is a measurement of direction between two points.

[5] https://atstrack.com/tracking-products/transmitters/product-transmitters.aspx?serie=A1500

Figure 2.2: On-field expert homing on VHF transmitter

VHF is a very effective way of tracking animals, although when studying migratory birds or far-ranging animals [28], the usage of these systems does not provide the necessary needed range, allowing the coverage of the area traveled by the species. Satellite telemetry establishes a trade-off between spatial resolution and range and maybe the best option to consider real-time monitoring on a global scale, although remaining expensive both in price and in battery autonomy.

ARGOS

ARGOS is a satellite-based system which began in the 1978 [29], and has global coverage through three subsystems:

1. the Platform Transmitter Terminals (PTT);
2. the space segment;
3. the ground segment;

PTT attached to animals transmit radio signals in the Ultra High Frenquency (UHF) band which is detected by ARGOS satellites located on a polar orbit at 850 km above the earth. Argos satellites cover 100% of the earth's surface with a visibility diameter of 5000 km. When a transmitter begins to be in range of a passing satellite, it has approximately a window of 10 to 12 minutes to send the frequency data and the required timestamps (Doppler effect)[6]. These data are then downloaded and distributed to the ARGOS processing centers, where the location is calculated. Also, these centers process the information received by the satellites and the location is assessed trough a least-squares analysis and assigned to one of the several Location Class (LC) which represents the range in which the position estimate is inserted. According to Nicholls, [30] ARGOS satellites can locate any transmitter between 1

[6] when a satellite approaches a transmitter, the frequency of the transmitted signal measured by the onboard receiver is higher than the actual transmitted frequency, and lower when it moves away

to 14 times per day with a high position accuracy of 1 km of standard deviation. However, the standard deviation of positional error in latitudinal and longitudinal axes is claimed to be 250 m for LC 3, between 250 m and 500 m for LC 2, between 500 m and 1500 m for LC 1, and more than 1500 m for LC 0 [7]. When three or fewer messages are received by satellite, the accuracy levels are LC A and B (without any estimation accuracy) or LC Z (providing invalid location). Finally, the location is transferred to the researcher using the ARGOS system. When applicable to ornithology, ARGOS PTT can be powered by solar cells which can weigh a total of 2 to 50 g having a base price of 2900 to 4450 US$, plus a monthly fee is charged when the platform transmits at least once during a given calendar month, with an estimated lifespan of 2 - 3 years. When powered by batteries ARGOS PTT can weigh from 45 to 105 g costing between 2550 to 2950 US$ with a lifespan between 40 days up to 3 years. [8]. In this study, the low-cost approach places the sensor two orders of magnitude lower than an ARGOS PTT.

Nevertheless, the ARGOS system can provide experts with information about marine species that live in harsh or inaccessible environments, which is impossible to perform with conventional VHF telemetry. However, the Doppler-based positioning can be also inaccurate, and may not provide the spatial fine detail needed for the experiments as argued in [31].

Global Positioning System

Conversely, GPS tracking devices provide fine-scale location data about a marine species, by receiving transmissions from a total of 24 satellites orbiting earth at 20000 km. When at least four satellites are in the range of the transmitter GPS provides a location accuracy below 30 m using trilateration measuring the signal distances as opposed to the measured angles in VHF triangulation. The data concerning location is transferred by associating GPS with a data-download method such as ARGOS, Global System for Mobile Communications (GSM), VHF or by simply storing the information on-board for later retrieval. However, in the ecology field, GPS telemetry systems have major disadvantages related to the costs. A single collar can range from around 2000 to 8000 US$ depending on the features of the unit itself [32]. Being a very expensive system, GPS has a direct influence on sample sizes of marine taxa to be used in studies, due to the trade-off between the number of units in field and cost per individual unit. Another problem related to GPS is the lifespan of the device, as GPS is a power-hungry location method. To mitigate this problem, many devices have programmable duty-cycles to take only a few locations per day and proceed to sleep, achieving the conservation of power. Typically, these devices can be supported by either batteries or solar cells. However, depending on the environment of the study setup and deployment, even solar cells may turn out to be non-viable options [21]. Currently,

[7] www.argos-system.org/manual/

[8] https://www.microwavetelemetry.com/avian_transmitters

Parameters	Archival Tags	VHF	ARGOS PTT	GPS
Power Source	Batteries	Batteries/ Solar Cell	Batteries/ Solar Cell	Batteries/ Solar Cell
Weight (g)	0.3 to 3	0.2 to 100	2 to 50/ 45 to 105	17 to 50
Lifespan	5 days to 2 years	Few days to 4 years	2 to 3 years/ 40 days to 3 years	up to 3 years
Range (km)	N/A	5 to 25	Global	Global
Location Method	GLS	Triangulation/ Homing	Doppler Effect	GPS
Positioning Accuracy	Low	Medium	High	Very High
Price US$	12 to 600	180 to 300	2900 to 4450/ 2550 to 2950	2000 to 8000

Table 2.1: Comparison of different technologies used in the ecology field

solar combined GPS/ARGOS systems weigh between 17 to 50 g with lifetime up to 3 years [9]. Table 2.1 summarizes the information found between the systems that are currently employed in monitoring the wildlife, concluding that each system presents advantages at the level of spatial detail given to experts and disadvantages related to cost, weight and lifespan constraints which condition the range of species for which they are available.

2.1.3 Location Estimation Techniques

To open the topic on estimating the emitting node's position, it is first needed to clarify that node position while transmitting will be relative to the gateways which are static and with known positions. Two popular techniques usually used to locate a device, which is triangulation, and multilateration. Bear in mind that each technique uses itself a distance estimation technique, like Angle of Arrival (AoA) for triangulation or alternatively Time of Flight (ToF) of the emitted signal, Time Difference on Arrival (TDoA) based distance or Received Signal Strength Indicator (RSSI) based distance for multilateration. Note that by multilateration we're specifying a lateration made by more than one gateway, as some scientific papers [33, 34] tend to classify bilateration, trilateration, and multilateration as different estimation techniques.

Triangulation

The angulation technique as explained in the VHF section (see section 2.1.2) makes use of the AoA to determine target location by intersecting pairs of angle direction lines. Understanding that the AoA on more than one antenna it is then a problem of solving the trigonometric relationships from the intersection of the two bearings formed by a radial line to the receiving gateways (see fig. 2.3).

[9] https://www.microwavetelemetry.com/avian_transmitters

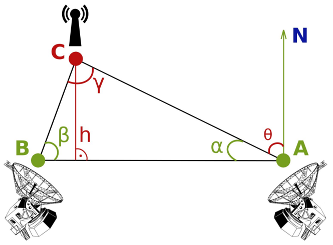


Figure 2.3: Enunciation of the triangulation technique

When in Line of Sight (LoS), the antennas between points A and B are adjusted to the point of highest signal strength. At this point, the antennas can directly be used to determine the angles of incidence β and α. This technique has the advantage of not needing any type of synchronization. Apart from the complex hardware setup, it is reported to only produce satisfactory results in situations with LoS conditions although accuracy and precision decrease when the signal is submitted to large distances and multipath (signal reflections). However, since no way to have an accurate and inexpensive way of finding the angle available at the time, for range, the multilateration technique was explored.

Multilateration

Multilateration is a technique that stands on the principle that a gateway will generate a circle with radius R, forming an area where signal transmission is possible (inside the circle) and forming an area where it is not (outside the circle). Ranging, how is commonly referred in some scientific papers [33–35], is the process of inferring the distance *d* that separates the emitting node from the receiving gateway (see fig. 2.4), thus the node will be located at any point on a circle of radius *d*.

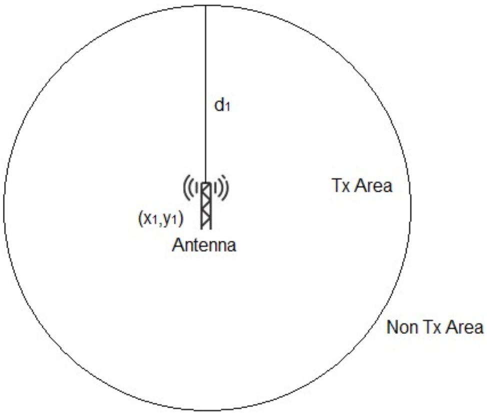


Figure 2.4: Antenna lateration principle

When applying the lateration to several gateways, i>1 we get several circle intersections equal to the number of gateways that are receiving in an instant *t*, note that more gateways mean more intersections which will improve the accuracy of the estimated point. Mathematically, and first referring to a situation wherein an instant *t* a signal is simultaneously received by three gateways i.e i=3 (trilateration), geometrically we have 3 gateways each forming a circle with known centers C$_1$=(x$_1$,y$_1$) C$_2$=(x$_2$,y$_2$) and C$_3$=(x$_3$,y$_3$) with radius R$_1$, R$_2$ and R$_3$ respectively, which intersect at the unknown point P(x,y), and which corresponds to the node's location (see fig. 2.5).

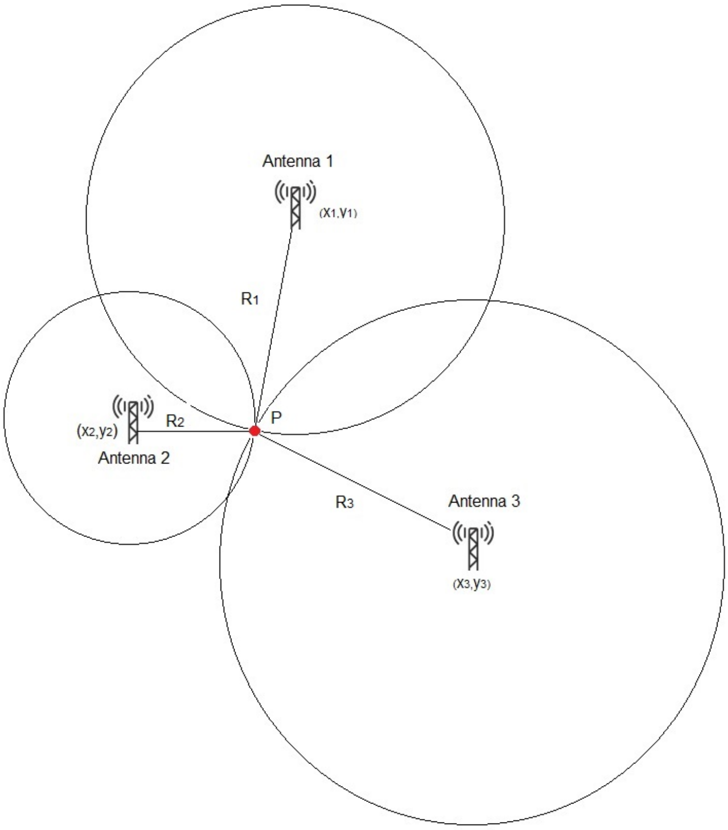


Figure 2.5: Antenna trilateration principle

Using basic mathematics, upon the transformation of the geodetic coordinates (lat, long) in cartesian coordinates (x, y), in 2-D space these circles need to satisfy the equation 2.12 as described in [35]:

$$(x - x_i)^2 + (y - y_i)^2 = R_i^2 \quad \text{(i=1,2,3,...,n)} \tag{2.1}$$

Similarly, if we want an understanding of the problem in terms of 3-D space, as it is used by the GPS, we pass the representation to 3-D space turning the geodetic coordinates (lat, long, alt) into cartesian coordinates (x, y, z), the previously obtained circles become spheres and the problem needs to satisfy the equation 2.2.

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = R_i^2 \quad \text{(i=1,2,3,...,n)} \tag{2.2}$$

Although for simplicity of reading and later working the data, the coordinates are only going to be handled in 2-D space as this study checks the feasibility of the method, however, this should be implemented in future studies.

The solution for the multilateration problem using a number of gateways, i=3 is the following:

starting from the equation for antenna 1 and expanding:

$$(x - x_1)^2 + (y - y_1)^2 = R_1^2 \quad \Leftrightarrow$$

$$x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 = R_1^2$$

applying the same method for 2:

$$(x - x_2)^2 + (y - y_2)^2 = R_2^2 \quad \Leftrightarrow$$

$$x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 = R_2^2$$

and for 3:

$$(x - x_3)^2 + (y - y_3)^2 = R_3^2 \quad \Leftrightarrow$$

$$x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 = R_3^2$$

(2.3)

at this point it is possible to compare all the three expressions, two equations at a time, subtracting the second equation to the first,

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = R_1^2 - R_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$$

Likewise, subtracting the third equation from the second,

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = R_2^2 - R_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2$$

reaching a linear system, with two equations and two unknowns,

$$Ax + By = C$$

$$Dx + Ey = F$$

which solving in order to x and y gives,

$$x = \frac{CE-FB}{EA-BD}$$

$$y = \frac{CD-AF}{BD-AE}$$

As previously mentioned, the solution presented in equation 2.3, will improve its accuracy once more gateways are added, although using a more refined method will probably translate in a more simple methodology to reach a solution. As an alternative to the solution presented above, one can use the iterative Gauss-Newton algorithm for non-linear systems, which will find the point (x, y) that minimizes residual errors in the equation 2.12 as explained in [33]. Note that the solution presented or the usage of the Gauss-Newton algorithm comes as a suggestion as different papers about the subject present different approaches [1, 33, 36, 37]. However, is this study no iterative methods for calculus were used given that our equations remained linear.

Considering now a situation where only two gateways receive a message i.e bilateration being i=2, as depicted in fig. 2.6 we have the following solution where:

- If d $>$ R$_0$ + R$_1$ then there are no solutions, the circles are separate.
- If d $< |R_0 - R_1|$ then there are no solutions because one circle is contained within the other.
- If d = 0 and R$_0$ = R$_1$ then the circles are coincident and there are an infinite number of solutions

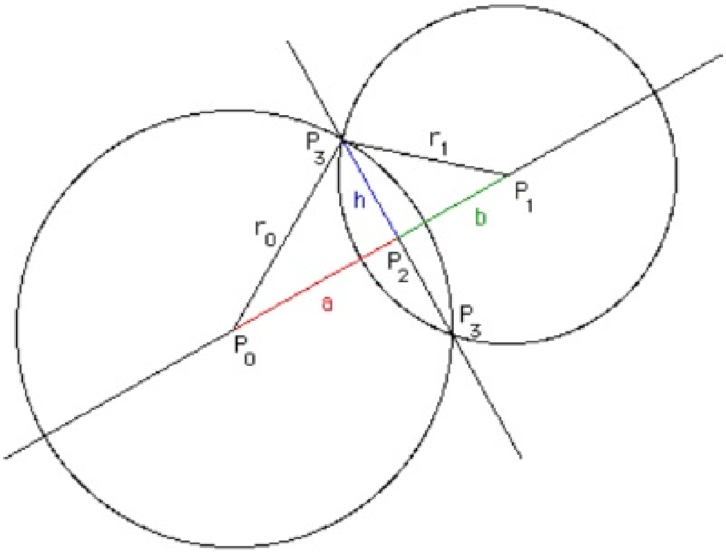


Figure 2.6: Antenna bilateration principle [1]

For the triangles formed by $P_0P_2P_3$ and $P_1P_2P_3$ we can write:

$$a^2 + h^2 = R_0^2$$

$$b^2 + h^2 = R_1^2$$

Using d = a + b we can solve for a:

$$a = \frac{R_0^2 - R_1^2 + d^2}{2d}$$

It can be readily shown that this reduces to r0 when the two circles touch
at one point, i.e.: d = $R_0 R_1$.Solving for h by replacing a into the first equation, (2.4)

we get $h^2 = R_0^2 - a^2$ so,

$$P_2 = \frac{P_0 + a(P_1 - P_0)}{d}$$

And finally, P_3 = (x_3,y_3) in terms of P_0 = (x_0,y_0), P_1 = (x_1,y_1) and P_2 = (x_2,y_2), is:

$$x_3 = x_2 \pm \frac{h(y_1 - y_0)}{d}$$

$$y_3 = y_2 \pm \frac{h(x_1 - x_0)}{d}$$

When the two circles do not intercept, we only know that the solution is along the line perpendicular to P_0P_1, with its center in the point P_2. The solutions for multilateration are at this point presented to the reader for i=1,2,3,...,n. In this bottom-down approach of the problem it's presented a guide on how to perform multilateration of a position, however, following section a research on how to get the radius for the circles referred above is performed.

Time of Flight - ToF

ToF is a ranging method discussed in this thesis which makes use of time synchronization taking as a principle the linear relation between the time of propagation, i.e ToF, and the speed of propagation *c*, to determine the distance traveled *d*, as shown in equation 2.5.

$$d = c \cdot \text{ToF}, \quad c = 3.0 * 10^8 m/s \tag{2.5}$$

When using the ToF method, the node measures how long it takes for the radio signal to travel from the tag to the antenna, satellite or Access Point (AP) and back to the tag. Making it then able to calculate the distance from the tag to each gateway. As previously mentioned, once the distance is estimated for at least 3 gateways, it's possible to calculate an exact position P(x,y) of the tag.

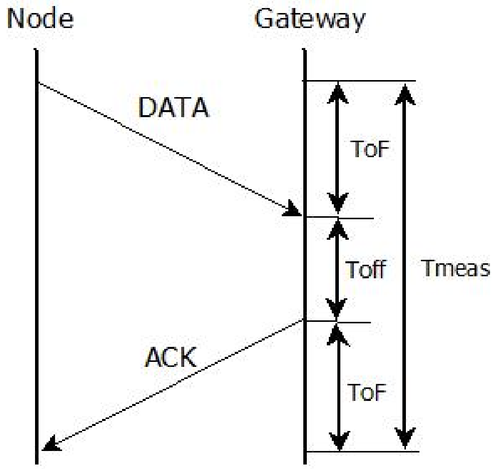


Figure 2.7: ToF ranging process [2]

Figure 2.7 depicts what is the process in ToF ranging. Relying upon DATA/ACK frames exchange between the node and gateways, it is possible to measure the time passed since the DATA transmission to the end of the acknowledgment reception, where T$_{meas}$ is given in equation 2.6.

$$T_{meas} = 2 \cdot \text{ToF} + T_{off} \tag{2.6}$$

Being the ToF the time of traveling of the signal and T$_{off}$ the time offset which corresponds to the time between frames of DATA and ACK. Assuming that one knows the T$_{off}$ for each frame exchange one can approximate the ToF as depicted in equation 2.7.

$$\text{ToF} = \frac{T_{meas} - T_{off}}{2} \tag{2.7}$$

Time Difference on Arrival - TDoA

When analyzing the TDoA method for estimation of position, it makes the use of the same principle shown in equation 2.5, with the particularity of synchronization being needed in the gateway side. When using this method, the node sends out data packets. All the nearby gateways will pick up these messages, however, as opposed to the ToF method, there is no need for ACK messages to be sent. Being the gateways positioned in different positions at different distances from the node, the message does not reach every gateway at the same moment in time. This method explores the differences in time between the gateways as a basis for multilateration calculation, and to determine the cartesian coordinates of an unknown point P. The key point in this methodology is that the gateways need to maintain synchronization, otherwise, its impossible to achieve an acceptable degree of confidence in the answers obtained. By using the time difference between gateways, it exists one measurement for each possible pair of gateways represented by Δt_{ij} = t_j - t_i, where t_i and t_j represent different timestamps from different receiving gateways. Distance then can be calculated using equation 2.5. By having the distance calculated, it becomes a problem of multilateration of the position as shown above.

Usually, methods that rely on synchronization report better accuracy in position estimation with relative complexity, requiring more sophisticated hardware which translates in more costs. These applications are usually more power-hungry requiring more messages exchange to achieve time values on which conclusions can be drawn, and for these reasons were left out of this study.

Received Signal Strength Indicator - RSSI

Another research perspective has focused on RSSI-based ranging. Instead of using a time-dependent method that will increase costs as well as the need of using more hardware and complexity, one can use RSSI measurements to estimate it's distance by using a radio wave propagation model or path-loss model. A path-loss model, explained in more detail below, explores the reduction in power density of an electromagnetic wave as it propagates through space, i.e the loss of signal strength, as a function of antenna properties, the frequency of emission, but mainly the distance between the emitter and the receiver. In theory, the best-estimated distance will just be a matter of choosing the best model, however, a model itself is subject to noise due to interference and obstruction, e.g. weather conditions. Therefore, a specific model will predict different distances for signals penetrating obstacles or for signals transmitted in LoS.

2.1.4 Radio Signal Propagation

Properties of radio signal propagation directly limit the performance of communication in Wireless Sensor Networks (WSNs). These same properties vary depending on the environment that the signal is subjected. Thus, a propagated radio signal will not behave the same in a situation in which it travels in LoS, or a situation with obstructions. When traveling through a wireless channel, a signal is generally affected by several propagation phenomena, such as absorption, reflection, diffraction, and scattering, which in turn will translate in propagation loss. When in the presence of buildings, trees or mountains large scale distortions are expected to occur. Reflection, absorption, and diffraction may translate in small scales distortions of the signal due to the presence of specific surfaces.

Free-Space Path-Loss Model

In telecommunications, when addressing radio signal propagation the free-space model is one reference in prediction models. This model works by assuming a clear LoS with no obstructions between emitter and receiver. This model is usually employed in wireless channels such as microwave or satellite transmissions. The path-loss is referred to in free space path loss, does not take into account any obstructions the path-loss L_{Pfree} is given in equation 2.8:

$$L_{Pfree} = [\frac{4\pi \cdot d}{\lambda}]^2 \tag{2.8}$$

where *d* represents the distance between emitter and receiver in km, and λ represents the wavelength given by equation 2.9

$$\lambda = \frac{c}{f} \tag{2.9}$$

Where *c*=3.0·10^8 m/s and *f* represents the frequency in Hz. By replacing the former wavelength frequency in equation 2.8 one can achieve the relation:

$$L_{Pfree} = [\frac{4\pi \cdot d \cdot f}{c}]^2 \tag{2.10}$$

And finally by expressing it in dB:

$$L_{Pfree[dB]} = 32.44 + 20log(f) + 20log(d) \tag{2.11}$$

As one can deduce, modeling a radio wave propagated signal without taking into account any type of terrain, and considering that no obstructions will be found, is at very best an optimistic hope that will only fit a very particular set of solutions such as satellite transmission or over the sea communications [38]. However, throughout this work, free-space model will serve as a basis, as we deal with estimating the sea birds on sea, for comparison for the results obtained with more refined models detailed in the sections below.

Antennas

One of the most important factors to take into consideration when building a system of this type is the choice of antennas on the market. It is useful to understand what types of antennas exist, and what are their basic properties [39]. Different antennas propagate signals in different patterns. Dipole antennas, used in this project, have omnidirectional properties, i.e radiates equally in all directions. By analyzing fig. 2.8 the resulting pattern in 3-D coordinates will resemble a donut shape, with the antenna in the center radiating energy outwards. Given this pattern, one can see that dipole antennas should be mounted so that is vertically oriented relative to the ground. The ”null” in this pattern i.e, the point in which almost no radio waves are radiated, is along the z-axis. This type of antenna will give the best performance and Radio Frequency (RF) range because of its dimensions and 3-D exposure.

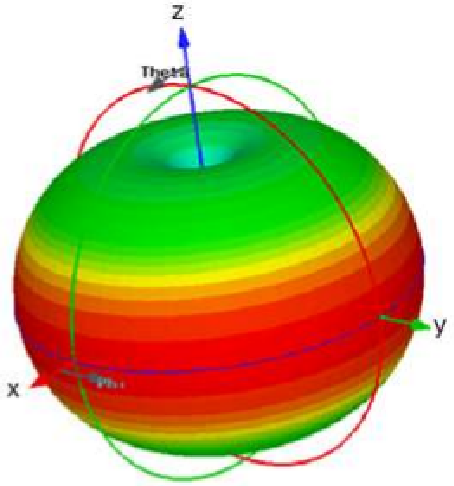


Figure 2.8: Dipole antenna radiation pattern

Another type of antennas, to consider are directional ones. Directional antennas are used for coverage point-to-point links. A good example of one of these antennas are dish antennas commonly seen in satellite systems, the goal is to radiate energy in a particular direction. Another example of directional antennas is the Yaggi antenna commonly used in VHF projects. These antennas are formed by driving a simple antenna and by shaping the beam using a chosen series of non-driven elements whose length and spacing are controlled. Yaggi antennas are directional antennas, instead of radiating energy in a circular pattern, energy is only radiated in a specific direction as can be seen below in fig. 2.9.

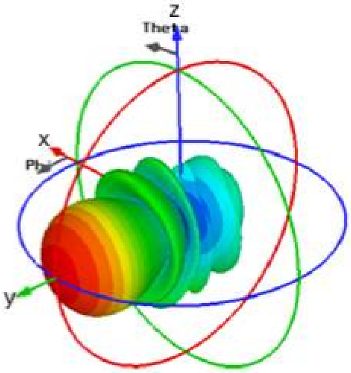


Figure 2.9: Yaggi antenna radiation pattern

To cope with the focus of this study, an omnidirectional antenna was used. A dipole was the elected choice, given that its radiation pattern results in the maximum amount of energy radiating outwards, thus maximizing the coverage area.

Polarization

Antenna polarization or polarization state is another concept that can influence transmission path. Antennas create both electric and magnetic fields, which are orthogonal between them forming a 90º angle with each other. These two fields create an electromagnetic wave that spreads out or propagates in a certain direction at speed of light *c*. At the point of reception, the wave hits the antenna which resonates at a specific frequency generating an electric current. During the transmission, electrons flow through the antenna and change direction depending on the signal frequency, creating the magnetic field. Antenna polarization is determined by the plane of the electric field. This implies that antennas are sensitive to types of electromagnetic waves. Although the concept to retain, is the practical implication of this concept which is that antennas with the same polarization will provide the best transmission/reception path. Often, simply by physically rotating the antenna used, the polarity is changed causing losses to the signal.

Signal Disturbances

When a signal propagates through the atmosphere, it suffers immediately the disturbances with the surrounding environment. Phenomenons like reflection, diffraction, refraction, absorption, and scattering are surely going to impact the propagation model [40]. Reflection can occur to a propagated signal in the presence of a variety of surfaces. When signal reflection occurs, there is usually some loss of the signal translated into the models, either due to absorption or by changing the medium. In long-distance transmissions, reflections may occur due to the presence of water. In contrast, the presence of buildings also provides very good reflectors. When obstructed by an object, diffraction occurs causing the waves to bend around the obstacle. This is known as shadowing and is caused by waves propagating into the shadowing region. Since only a part of the waves propagates into the shadowed region, the strength of the signal experiences rapid decays when propagating. Scattering occurs when the reflected energy spreads out in different directions, this can occur due to roughness in surfaces. This is usually the case when the signal is obstructed by an object smaller than the wavelength of the signal λ. Finally, Multipath propagation, a well-known phenomenon in propagation which will cause fluctuations as multiple waves will reach the receiver with a difference in phase. Multipath will result in fading of the signal [41].

Radio Wave Propagation Models

There is an extension of models to choose when working with transmission and trying to assess signal path-loss to improve performance, while also elaborating an estimation of the financial feasibility. As previously mentioned, a signal can suffer from different types of disturbances. Furthermore, each model adds a factor to represent the importance of it in the path-loss. The free-space model, as explained above, represents path-loss as a function of the frequency of transmission, and distance between emitter and receiver. Although further factors are added to other models like the Okumura/Hata model such as antenna height and base station antenna height to achieve a more accurate estimation [42]. Furthermore, branches are added to the equation to model propagation in different environments, such as rural, and urban environments. In summary, one has to classify the environment surrounding the transmissions to choose one of the several existent models.

In this study, the path-loss equation is manipulated to derive the distance *d*, instead of the value of path-loss, given that both transmitted and received powers are known. Moreover, the position estimation used in this thesis is located at sea, as our project places gateways on land receiving telemetry of a non-static emitter located at sea. In the literature [38], the scenario that characterizes the best states that transmissions that take place over water, or at sea with LoS conditions, are best described by the previously explained free space path-loss model and by the log path-loss model.

Log-distance Path-Loss Model

The Log-distance path-loss model presents itself as a practical and simple way of estimating the signal strength as a function of distance. The equation that defines the model is,

$$PL = PL_{d_0} + 10 \cdot \gamma \cdot log\frac{d}{d_0} \tag{2.12}$$

where the γ parameter represents the path loss exponent, and d_0 is the reference distance in meters, usually at 1 m, for the path-loss PL_{d_0} reference and *d* the distance of interest. This model is used throughout the chapter 4 and provides a practical way of adapting to each environment, although some references, of path-loss exponents, are given [33], a good practice is to determine the exponent for the model to be best adapted to the surrounding environment.

2.2 State of Technology

To satisfy the aforementioned constraints and provide solutions to expensive tagging techniques, several trade-offs should be taken into consideration. In this study, the focus will be on using IoT solutions as low-cost alternatives. The process of choosing the most adequate MCU for prototyping needs to pass rigorous steps of evaluation of weight, size, and price, to aim for the lightest and cheapest solution achievable. Although weight was considered at this phase, it was discarded at the end as the final prototype could not be attached to a bird as-is. Also, more considerations should be taken regarding the sensing shields/parts available on the market, allowing us to sense the parameters. Finally, communication protocols supported by each of these boards with the least possible hardware should be analyzed. To select a proper low-cost technology for communication, in chapters below it will be depicted the process of choosing which ran by evaluating the LPWAN available in the IoT area, including the main constraints on ranges, payloads, and prices.

2.2.1 Single-board Microcontrollers

The market options for development boards have been increasing in the last years. In this chapter, a comparison was made by investigating the most prominent three IoT ready boards. The Particle Photon, given it's usage in previous projects within M-ITI; the Arduino UNO, given it's popularity between developers and hobbyists, and the Pycom LoPy 4, given it's recent launch and interest in the market.

Arduino

Arduino is an open-source tool for developing systems for sensing and actuating in the physical world. Since 2005, Arduino has been creating low-cost and easy to use, open-source physical computing platforms based on a simple MCU board, as well as a development environment for writing software for the board. The software is written in C or C++ programming language. Between the range of choice, one can do in Arduino boards, it is highlighted the Arduino UNO since it is the most used board within the Arduino spectrum for in-situ IoT solutions. UNO (see fig. 2.10) is based on the ATmega328P running at 16 MHz with an 8-bit core and has a limited amount of available memory with 32 Kb of memory and 2 Kb of random access memory [10]. In terms of size, it has a length of 68.6 mm and a width of 53.4 mm, while having a total weight of 25 g. It consumes around 50 mA when active, and around 35.0 mA when in sleep mode with no additional external outputs.

[10] https://store.arduino.cc/arduino-uno-rev3

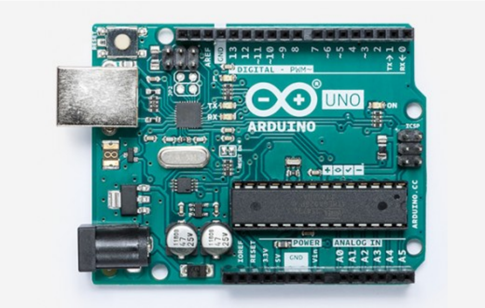


Figure 2.10: Arduino UNO

Particle Photon

Particle Industries Inc. provided a IoT device called Photon, which is 37 x 20 mm and 5 g open-source module combining an ARM MCU integrating the WiFi chip. With its 18 GPIO peripherals, Photon makes a great IoT device (see fig. 2.11), allowing remote controlling and data gathering from multiple connected sensors with 1 Mb flash, and 128 Kb of random access memory (RAM). Particle photon is programmed in C and has an active current of 80 mA and a deep sleep current of 80 μA, with a clock running at 120 MHz. Particle intended to combine their solutions into a fully integrated IoT platform that offers the hardware, paving the way for the constant connection to the internet, over WiFi, cellular, or mesh networks. It also has the growing community, offering the software through extensive libraries and support teams, and most importantly, free-of-charge connectivity solutions in their Particle Cloud. With this connectivity in mind, Particle allows every device to be easily connected to the cloud. Once connected, the devices can be accessed and it's possible to push code to a device and flash the firmware using WiFi [11].

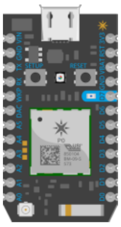


Figure 2.11: Particle's Photon representation

[11] https://store.particle.io/collections/mesh

Pycom LoPy4

Given its recent launch in the market, this study will also consider the Pycom LoPy 4. The Lopy 4 (see fig. 2.12) is a 55 x 22 mm and 7 g quadruple bearer MicroPython enabled development board (LoRa, Sigfox, WiFi, Bluetooth) With the latest Espressif chipset the LoPy4 offers a combination of power, and flexibility [12]. The Lopy 4 runs at 32 kHz and has 4 Mb of random access memory and 8 Mb of flash memory. The LoPy 4 is of particular interest due to its low power base for IoT solutions.

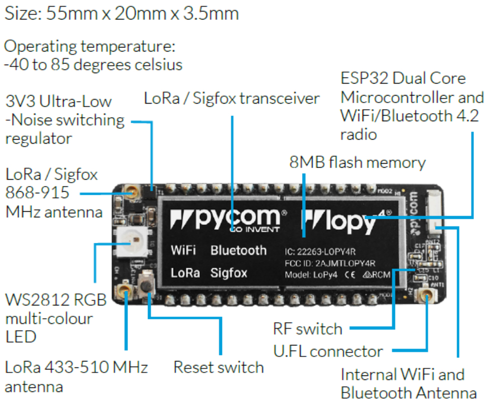

Size: 55mm x 20mm x 3.5mm
Operating temperature: -40 to 85 degrees celsius
3V3 Ultra-Low -Noise switching regulator
LoRa / Sigfox transceiver
ESP32 Dual Core Microcontroller and WiFi/Bluetooth 4.2 radio
8MB flash memory
LoRa / Sigfox 868-915 MHz antenna
pycom GO INVENT
lopy4
WiFi Bluetooth
LoRa Sigfox
IC: 22263-LOPY4R
FCC ID: 2AJMTLOPY4R
Model: LoPy4
WS2812 RGB multi-colour LED
RF switch
U.FL connector
LoRa 433-510 MHz antenna
Reset switch
Internal WiFi and Bluetooth Antenna

Figure 2.12: Pycom Lopy 4

When active, and in full capabilities, the LoPy 4 generates approximately 20 mA, while in deep sleep can go as low as 25 μA. These characteristics suggest this MCU to be ideal for the deployment in this study.

2.2.2 Low Power Wide Area Networks

IoT applications require constant connection and transmission of data among devices. Conversely, LPWANs are projected as the connectivity solutions to support a major portion of the billions of devices forecasted for the IoT [43]. These applications are designed to meet specific requirements such as increased battery autonomy, adequate capacity, long-range, and low-cost. Although there is many emerging LPWANs, in this review we will consider only LoRa and Sigfox.

Sigfox

Sigfox is a LPWAN candidate that provides global wireless networks to enable an end-to-end IoT connectivity solution in 60 countries, by December 2018 [13] Sigfox uses an IP-based network to connect their base stations to the back end servers [44]. Its approach is similar to cellular network operators

[12] https://pycom.io/product/lopy4/

[13] https://www.sigfox.com/en

which provide services to devices however Sigfox has the advantage of providing low power consumption and consequently low-cost performance [45]. When transmitting from devices to base stations (uplinks) Sigfox uses Differential Binary Phase-Shift Keying (DBPSK) modulation and for downlink transmissions Gaussian Frequency-Shift Keying (GFSK) technique, which are less frequent [45]. Sigfox uses part of the unlicensed Short-Range Devices (SRD860) and Industrial Scientific and Medical (ISM) frequency bands, for example, 868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia. The use of unlicensed frequency bands is cost-free but duty cycle and maximum transmitted power are limited due to regional restrictions which affect the latency and downlink transmission [45]. Because Sigfox uses an ultra-narrow band, this allows the efficient usage of the frequency band. This means that Sigfox is resilient to interference and noise levels, leading to very low power consumption, high receiver sensitivity, and low-cost antenna design. Every message requires only 100 Hz to be transmitted, leading to a data rate equal to 100 bps which can range up to 10 km in urban areas and up to 40 km in rural areas. The low data rate is considered to be the main drawback of Sigfox which limits its operability in high data applications [44].

Initially, Sigfox only supported uplink communication, but later evolved to bidirectional technology with a significant link asymmetry [44]. Now the uplink transmissions are limited to 140 messages of length equal to maximum 12 bytes each, transmitted per day. Downlink transmissions have a limit of 4 messages of length equal to a maximum of 8 bytes each, transmitted per day. There is an asymmetric relation between uplink and downlink in Sigfox technology, therefore there cannot exist downlink acknowledgment for every single uplink message. To compensate for the lack of acknowledgments for each uplink message, Sigfox takes advantage of frequency and time diversity. The device transmits the same message 3 times using 3 different time slots, each having a different frequency [45]. Competing with the Sigfox solution there is the solution brought by LoRa which will be compared below.

LoRa

LoRaWAN standard is offered by Semtech, first released in 2015 and developed by LoRa Alliance as a wireless communication standard operating within the unlicensed bands. The name stands for Long Range Wide Area Network.

When speaking of LoRa it is important to distinguish between LoRaWAN and LoRa, as they are distinguishable terms. LoRa defines the modulation in the physical layer and LoRaWAN defines a Media Access Control (MAC) protocol that supports the low power, long-range and high capacity in LPWAN [45]. LoRa, which is part of the physical layer, creates the value for the above layers (see Fig. 2.13). Describing the modulation is what maintains the long-range feature in LoRaWAN.

LoRa is based on Chirp Spread Spectrum (CSS) modulation, which offers the same low power characteristics as a Frequency-Shift Keying (FSK) modulation. Moreover, it significantly increases the communicating range. Other solutions featuring CSS modulations may be found in military and space applications, due to the long distance of communication that can be achieved. LoRa is a pioneer low-cost implementation for commercial usage [43]. The spread spectrum separates signals by exploiting the orthogonality between them by using a unique spreading factor to the individual signal [46]. This method gives advantage in terms of data rate, as there are six orthogonal spreading factors in the range of 7 to 12, which provide different data rates. The bit rate is inversely proportional to the spreading factor and the higher the spreading factor, the longer is the communication range. Although higher spreading factors mean more power consumption. Depending on the spreading factor and channel bandwidth that can be 125 kHz or 250 kHz, the data rate can range from 300 bps to 50 kbps, with a maximum payload length per message of 243 bytes, that can range from 5 km in urban areas to 20 km in rural areas [44].

Also, the MAC layer provides different classes of device options to provide flexibility to meet the requirements of IoT applications. The device classes tradeoff network downlink communication latency versus battery lifetime [43]. These will be explained below and later on, in the sections below we'll compare both technologies in terms of IoT factors. Quality of Service (QoS), payload length, network coverage, and range should always be compared when studying different LPWANs.

LoRaWAN

LoRa Alliannce defines and offers the following LoRaWAN specification and regional parameters [43].

LoRaWAN is a network protocol in the LPWAN category optimized for battery sensitive devices. The network has a star topology composed of end-devices, gateways, and a network server. The end-devices make use of a single-hop LoRa to one or multiple gateways that in turn are connected to the network server using standard IP connections as depicted in fig. 2.13 [46].

The communications between these devices are spread in different frequencies and data rates, which can range from 0.3 kbps to 50 kbps. Although end-devices should respect the following two rules to be allowed to transmit in a channel:

- The end-devices should respect the maximum transmit duty cycle relative to the sub-band used and local regulations. An example of this usage is, if a device transmits a 0.5 seconds frame on one of the default channels, in order to comply with the 1% duty cycle, that device cannot transmit on that whole sub-band (868-868.6MHz) during $0.5 * 99 = 49.5$ seconds.
- The end-device respects the maximum transmit duration relative to the sub-band used and local

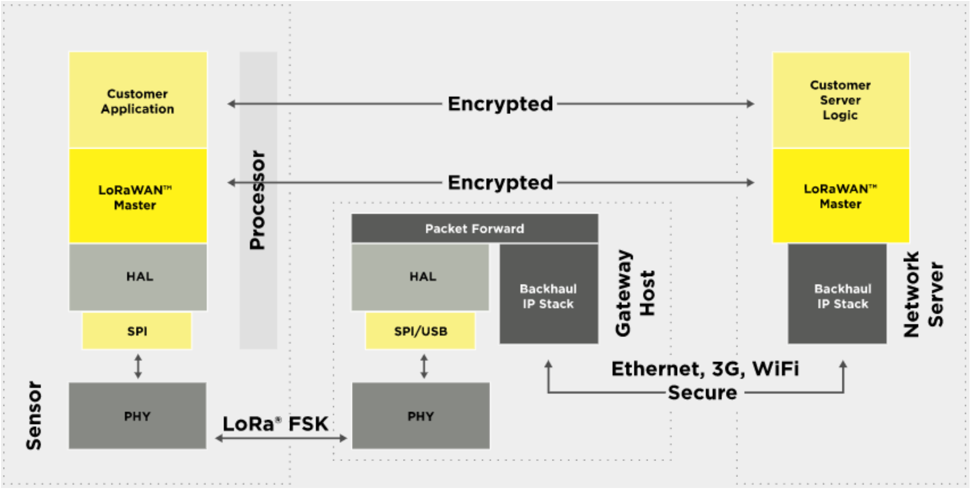
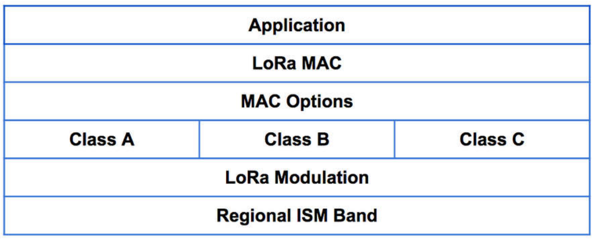


Figure 2.13: LoRaWAN architecture

regulations. In our case, there is no dwell time limitation imposed by ANACOM for the EU863-870 ISM band, as can be seen in https://www.anacom.pt/render.jsp?contentId=1380834

Device Classes

LoRa networks distinguish end-devices based on a configurable parameter called the device class, from which one can configure it as Class-A (default), Class-B, and Class-C (see fig. 2.14).



Application		
LoRa MAC		
MAC Options		
Class A	Class B	Class C
LoRa Modulation		
Regional ISM Band		

Figure 2.14: LoRaWAN classes

These classes can be explained as follows:

- Class-A devices allow bidirectional communication; for each device uplink transmission there's two short downlink receive windows (see fig. 2.15). This class represents battery powered sensors and is the mandatory class for every device used, being the most energy efficient class by requiring short downlink communications e.g ACK message [44].
- Class-B devices, in addition to the downlink window of class-A, opens extra receive windows at scheduled times. In order for the window to be open at the scheduled time, the device receives a time-synchronized beacon from the base station. This allows the network server to know when

the end-device is listening. Class-B is mainly used by battery powered actuator, where energy consumption is still efficient although with latency controlled downlink.

- Class-C devices are used by main powered actuators, that is, devices that can afford the energy consumption in order to be always listening for downlink messages which don't have to wait until the next transmission slot available. This class is the most power demanding, however offers the lowest latency in the device to server communication.

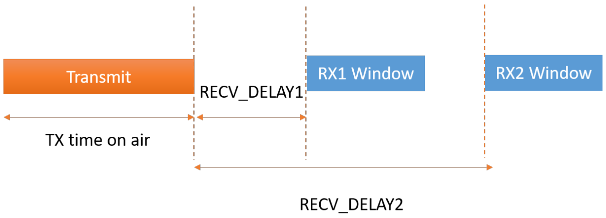


Figure 2.15: LoRaWAN Class-A

Regional Parameters

As a matter of ensuring interoperability between the different networks, the LoRaWAN specification defines parameters that should be followed for each different region. This project makes use of the 868MHz frequency which allows for different data rates depending on the spreading factor used.

Security

The LoRaWAN protocol offers security by using three different 128 bit keys:

- Application key which is generated using AES-128. Furthermore this key is also used to generate the remaining keys.
- Network session key which is derived from the AppKey, and known to the network in order to assure the integrity and authenticity of a packet.
- Application session key which is also generated from the AppKey, and ensures end to end encryption for the application payload.

These keys can be used in two different methods. Activation By Personalisation (ABP) and Over the Air Activation (OTAA). In ABP the join procedure is skipped, remaining the two session keys (NwkSKey and AppSKey) static and personalized by the user. On the other side if OTAA is activated the end-device performs a join procedure being these keys re-generated on every session. Concluding, integrity is ensured only within the network, although confidentiality is ensured throughout the whole process.

2.2.3 Wireless Connectivity

The options available for wireless IoT connectivity continues to grow. Popular connectivity methods like Wi-Fi and Bluetooth are specialized by technologies that push the limits of what is possible. These types of wireless connections like Wi-Fi, Bluetooth, and Cellular are very popular choices even among the IoT field, although depending on the requirements of the project most of these lack an adequately long range of coverage or they require too much power input to operate consistently. The reason why Wide Area Neworks (WANs) hasn't been mentioned is that the requirements fulfilled by these don't cope with the requirements like long-range coverage and low power requirements.

Local Area Network (LAN) technologies, like WiFi Bluetooth or Zigbee, have all well-established standards, and provide a good basis for indoor devices although, the short-range, and poor battery longevity are features that would make the communication in a project like these impossible.

Cellular networks, like GSM, 3G, 4G and LTE like LANs have well-established standards while providing a long-range with good coverage at high data rates, although the power required as input makes battery critical sensors impossible to deploy as no longevity would be achieved. fig. 2.16 depicts the comparison in range and data rate between several wireless technologies.

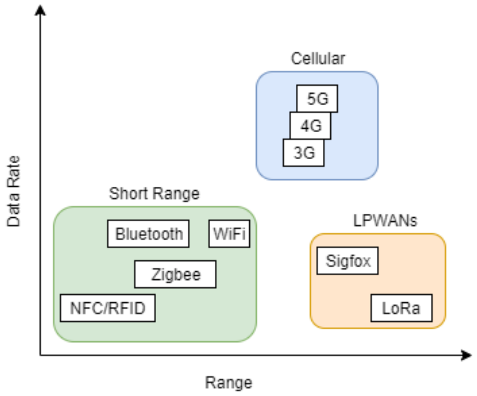


Figure 2.16: Wireless connectivity compared between data rate and range

2.2.4 Opportunities

The subject of IoT has been long championed. Although, previous research did not identify a IoT solution which can be applied to the ecology field, to be used for monitoring the marine taxa or monitoring the environment surrounding the marine taxa carrying sensors. Previously, in table 2.1, it has been highlighted some of the differences with a focus on the costs of employing the current state-of-the-art technologies when studying a species or species behavior. In the related work section and supported by table 2.1, the reader may notice that leaving aside the right technology for the right experiment, there is not a lot of flexibility when choosing between a technology. The costs sometimes can be too high for gaining little conclusions due to equipment failure or permanent loss. With the development in the IoT field, using a hardware solution described in the section of single-board MCU, it is possible to achieve a simpler and cheaper solution. Table 2.2 summarizes the boards reviewed in the previous section.

Board	Arduino UNO R3	Particle Photon	Pycom Lopy 4
Weight (g)	25	5	7
Size (mm)	68.6 x 53.4	37 x 20	55 x 22
Language	C/C++	C	MicroPython
Clock (MHz)	16	120	32
Power Consumption (active/sleep)	50mA/35mA	80mA/80μA	20mA/25μA
Price EUR	20.00	16.58	59.90

Table 2.2: Comparison of the different boards reviewed

Another possibility would be, to have a custom board made to cope with the solution proposed in the next sections. Taking into account all of the constraints embracing processing power, autonomy, power consumption, it could be possible to build better hardware to fit the solution. Although, such work is out of the scope of proposed work in this thesis, as it requires more time for prototyping the boards and more electrical engineering knowledge.

Many factors should be considered when choosing the appropriate LPWAN technology for a IoT application. Factors as QoS, battery life, latency, scalability, payload length, network coverage, range, and cost. These factors for Sigfox and LoRa are compared, as well as their technical differences. In terms of QoS, both Sigfox and LoRa use unlicensed bands and asynchronous communication protocols. They can reject interference, multipath, and fading. Although QoS is not excellent, it seems adequate for this kind of application, where sporadic packet loss does not translate in major impact as in hard-time systems. As Sigfox and LoRa channels are not used 99% of the time, meaning that end-devices are in sleep mode most of the time, reducing the amount of consumed energy, and eventually lead to important cost reduction. Therefore, both can afford lifetime battery up to 13 years [45]. Sigfox and class-A LoRa are the best options for applications that are insensitive to the latency and do not have a large amount of data to send, which makes both good choices for this project. Nevertheless, if technical specifications on both are compared, class-A LoRa seems to be the best choice for this project since it can transmit unlimited messages per day, unlike Sigfox. One of the key features of Sigfox and LoRa is the support of a massive number of devices. Both technologies work well with the increasing number and density of connected devices. LoRa allows a maximum of 243 bytes of data to be transmitted. On the contrary, Sigfox proposes the lowest payload length of 12 bytes [45], which limits its utilization on any IoT application that needs to send large data payloads. In terms of coverage, a wide area can be covered with just a few base-stations/gateways with these technologies. For example, an entire city can be covered by one single base station and a country like Belgium can be covered with only seven base stations [46]. Nevertheless, the key to long coverage of Sigfox and LoRa is the physical layer. These challenges are overcomed in different ways for both technologies. LoRa modulates a CSS which performs on large distances and is robust against noise and interference, and Sigfox uses an ultra-narrow band with slow modulation scheme BPSK [45]. Also, various cost aspects need to be considered such as spectrum cost (license), network/deployment cost, and device cost. Sigfox and LoRa are very cost-effective. Although Sigfox deployment cost is $\geq$ 4000 $ per base station, while the end-device cost is only $\leq$ 2 $ and has a nonexistent spectrum cost. LoRa deployment cost is $\geq$ 100 $ per gateway, $\geq$ 1000 $ per base station and the end-device cost is 3 to 5 $ [44]. The previous comparisons are summarized in table 2.3 and will be mentioned again in the following section. As Sigfox is also for-profit and being LoRa an open platform, the LoRa technology was chosen.

Parameters	LoRaWAN	Sigfox
Modulation	CSS	BPSK
Frequency	Unlicensed ISM bands	Unlicensed ISM bands
Bandwidth	250 Hz and 125 Hz	100 Hz
Max data rate	50 kpbs	100 bps
Bidirectional	Yes	Limited
Max msg/day	Unlimited	140 (UL), 4 (DL)
Max payload	243 bytes	12 bytes (UL), 8 bytes (DL)
Range	5 km (urban), 20 km (rural)	10 km (urban), 40 km (rural)
Standardization	LoRa-Alliance	Sigfox

Table 2.3: Summarization of the discussed LPWANs

3

Implementation

Contents

3.1 Project Proposal

To face the already existent systems in the market that make use of VHF, ARGOS or GPS and following the original proposal of developing a wildlife tracker to be used on birds, it was seen as important to summarize this project proposal.

3.1.1 Project Requirements

By the meetings that took place in Madeira Island with Dr. Francis Zino, a renowned ornithologist, a study with birds conducted by a team of ornithologists would have to endure a minimum of three months period for gathering data. The systems used by Dr. Zino and his team currently employ ARGOS devices, however, the cost per tag and per download upon information treatment leads to a situation where it becomes impossible to expand the experiences to more subjects due to financial costs. GPS trackers are avoided at all costs due to the financial cost of the project and also because one can not often achieve the longevity necessary for these experiments to take place due to the battery drained by the GPS. GPS and ARGOS have both high accuracy in location estimation as referred in chapter 2, however this degree of precision is sometimes unnecessary, so it is possible to achieve a reduction in accuracy when estimating the position while developing a cheaper system.
Data accessibility is another problem encountered due to the fact that when ARGOS trackers are not employed due to their cost the usual system used are archival tags that, as said before, produce telemetry and store it in a Secure Digital (SD) card. The difficulty that this method poses is directly related with the retrieval of the device, as normally birds nests are in inaccessible terrains which make up the need to gather experienced climbers for the retrieval of the device, a situation that not only poses with financial costs but also can be potentially dangerous.

This project proposal intents to tackle the enunciated points above, so as a requirement the sensor should:

- Weigh as least as possible (2-3% of the total bird weight);
- Be efficient in terms of battery consumption in order to withstand a period of 3 months producing telemetry, with a satisfactory frequency of transmissions (when compared to the technologies in chapter 2);
- Produce a satisfactory position estimation based on RSSI values (when compared to the technologies in chapter 2);

3.2 Research Apparatus

After the elaborated study on the presented technologies in the section of related work, this thesis provides an approach for a ground-based system to be used on marine fauna focusing on seabirds. This solution proposes a remote sensing system which has not been yet employed in the ecology field. By using a low-cost development board and a LPWAN, this solution will broaden horizons in the field, by providing a low-cost, low power and long-range system, to face the technologies used nowadays. It was seen that the technologies in use are either ground- or satellite-based, which underline high costs, as well as that they are seen as a big disadvantage in the ecology field. Currently, ecology is paying too much for solutions based in Argos and GPS. This field can, therefore, benefit with a solution based on the ecology knowledge with the application of the IoT field. For that reason, this solution thesis focuses on prototyping a solution with a low-cost development board, using a low power and long-range LPWAN.

In this section, we describe the apparatus used for testing in both the coastal and the sea vessel endpoints and the used LoRa settings. The deployment consisted of 4 coastal based nodes (however 2 failed) and 1 sea vessel node attached to the mast of the ship for the duration of 1 hour, allowing to test and determine the location of the ship. It is important to consider that the development of the final prototype as the result of this thesis work and the prototype for testing have undergone different algorithms and different setups in testing, and these will be described as separate schemes in the following sections.

3.2.1 Hardware

The development board chosen among the previous comparisons was the recently launched Pycom Lopy 4 (see fig. 2.12). This choice was due to its low power consumption which was seen as a most valuable feature among devices of his kind. Also, Pycom offers a set of shields to which the Lopy 4 can be attached, and so for the ease of it, we'll also be using the Pycom Pysense. Although this decision made the size and weight to be secondary, the latter allows with no additions, to work with a set of radio technologies (WiFi, Sigfox, Bluetooth, LoRa) as well as an initial large array of environmental telemetry parameters from the start, incorporating the following sensors:

- Acceleration (LIS2HH12 sensor) measured in G (9.806 m/sec^2).
- Light (LTR-329ALS-01 sensor) measured in lux.
- Humidity (SI7006A20 sensor) measured in percentage values (%).
- Temperature (SI7006A20 sensor) measured in Cº.
- Air Pressure (MPL3115A2 sensor) measured in mbar or hPa.

When working in the Pycom environment, one can also use the Pytrack shield which is also a shield developed by Pycom enhanced for localization systems equipped with an Inertial Measurement Unit (IMU) and a GPS module to provide a ground truth position to be used later for evaluation. However this shield was tested *in situ* and the ground truth GPS reading performed poorly, due to inaccuracies in obtaining a reading of position with a satisfactory error relative to the real position of the sensor. Due to this fact using the Pytrack shield was early discarded. Instead it was seen as a more useful approach to go with the Pysense (see fig. 3.1) in order to produce telemetry and couple an external GPS module which was chosen the ADH-tech GP-735 (see fig. 3.2) due to features related with size, weight, high performance readings, and above all to low-power consumption which is critical feature when working with any GPS module and also trying to prolong the lifetime of the device. For transmission and to make use of the Semtech LoRa transceiver SX1276 radio, an SMA Tilt Swivel 1/4 Wavelength Dipole antenna was externally attached and lastly for power supply it was used an 1800 mAh LiPo battery.

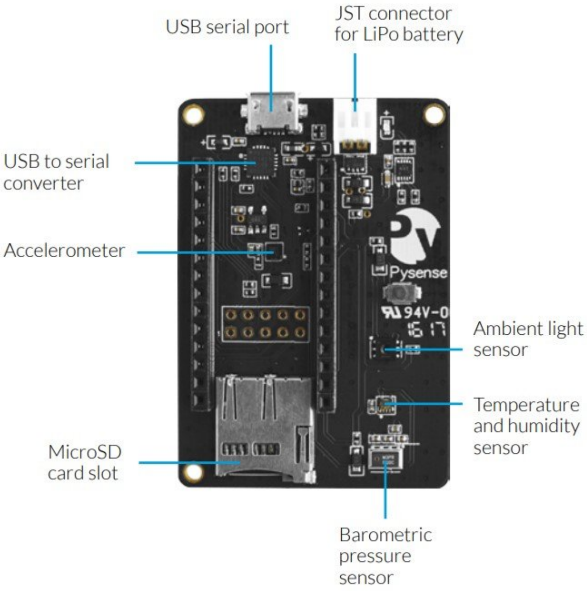


Figure 3.1: Pycom Pysense Shield (11g)

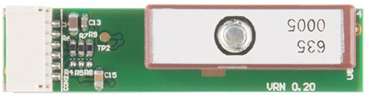


Figure 3.2: External GPS module GP-735 (3g)

In this way the system was implemented using the hardware described above, and relating to practicality this system brought the following advantages:

- Programs written in the MicroPython[1] programming language, allowing rapid prototyping and de-

[1] https://micropython.org/

velopment, with minimum constraints, e.g: massive library imports.

- A system capable of acting both as a device and as a gateway, allowing the code to be reused between both sides of the system.
- Possibility of producing telemetry through the Pysense shield with no further complexity related with electronics and wiring.
- Possibility of producing a system with precise GPS readings while not leaving aside the requirement of battery optimization.

Relating to hardware both emitter and gateways are identical, except for the fact that no external GPS module was attached to gateways, due to the fact that when experimenting with the purpose of testing and estimating positions there is no real need for this module since the positions of the coastal gateways are static and known beforehand.

3.2.2 Wiring

This section will explain how the wiring was made and wrapped to place the wired sensor inside of a plastic protective casing. The first consideration in terms of hardware wiring was transmitting in LoRa connecting the SMA Tilt Swivel 1/4 Wavelength Dipole antenna[2] to the 868 MHz port while leaving the 433 MHz port free of connections. This choice has been done because the relation made in equation 2.9 using lower frequencies would result in the need for using different antennas, which were not available at the time.

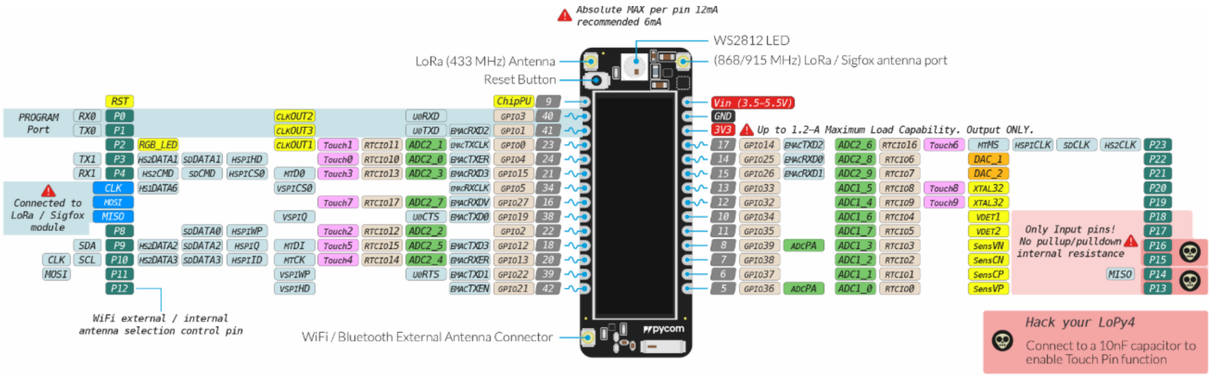
Relating with the ease of operating the sensor, it was added a toggle switch connected with jumper cables between the battery and the JST connector present in the PySense shield (see fig. 3.1), this was added also to the fact that operating the sensor and connecting batteries by opening the case and connecting it directly to the JST connector would sometimes result in disconnecting other wired cables, compromising functions of the sensor.

LoPy 4 provides the flexibility of 25 I/O pins to work with, ranging from Universal Asynchronous Receiver/Transmitter (UART), Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) pins (see fig. 3.3).

Although working with these I/O pins was not as simple as the documentation led to believe. This was because it was already impractical to connect a pin and not compromise the internal functions of the MCU. Further, adding the Pysense shield resulted in more occupied I/O pins for the good function of its sensors (see fig. 3.4).

The 6-pin interface of the GP-735 can be seen in fig. 3.5. Initially, experimental tests were conducted with just the 1 to 4 pins connected i.e, the VCC, GND and UART TXA and RXA pins working. Although

[2] included in the kit when ordering



Absolute MAX per pin 12mA
recommended 6mA
WS2812 LED
(868/915 MHz) LoRa / Sigfox antenna port
LoRa (433 MHz) Antenna
Reset Button
RST ChipPU 9
PROGRAM Port RX0 P0 CLKOUT2 U0RXD GPIO3 40
TX0 P1 CLKOUT3 U0TXD EMACRXD2 GPIO1 41
P2 RGB_LED CLKOUT1 Touch1 RTCIO11 ADC2_1 EMACTXCLK GPIO0 23
TX1 P3 HS2DATA1 SDDATA1 HSPIHD Touch0 RTCIO10 ADC2_0 EMACTXER GPIO4 24
RX1 P4 HS2CMD SDCMD HSPICS0 MTDO Touch3 RTCIO13 ADC2_3 EMACRXD3 GPIO15 21
CLK HS1DATA6 VSPICS0 EMACRXCLK GPIO5 34
Connected to LoRa / Sigfox module
MOSI Touch7 RTCIO17 ADC2_7 EMACRXDV GPIO27 16
MISO VSPIQ U0CTS EMACTXD0 GPIO19 38
P8 SDDATA0 HSPIWP Touch2 RTCIO12 ADC2_2 GPIO2 22
SDA P9 HS2DATA2 SDDATA2 HSPIQ MTDI Touch5 RTCIO15 ADC2_5 EMACTXD3 GPIO12 18
CLK SCL P10 HS2DATA3 SDDATA3 HSPIID MTCK Touch4 RTCIO14 ADC2_4 EMACRXER GPIO13 20
MOSI P11 VSPIWP U0RTS EMACTXD1 GPIO22 39
P12 VSPIHD EMACTXEN GPIO21 42
WiFi external / internal antenna selection control pin
WiFi / Bluetooth External Antenna Connector
Vin (3.5-5.5V)
GND
3V3 Up to 1.2-A Maximum Load Capability. Output ONLY.
17 GPIO14 EMACTXD2 ADC2_6 RTCIO16 Touch6 MTMS HSPICLK SDCLK HS2CLK P23
14 GPIO25 EMACRXD0 ADC2_8 RTCIO6 DAC_1 P22
15 GPIO26 EMACRXD1 ADC2_9 RTCIO7 DAC_2 P21
13 GPIO33 ADC1_5 RTCIO8 Touch8 XTAL32 P20
12 GPIO32 ADC1_4 RTCIO9 Touch9 XTAL32 P19
10 GPIO34 ADC1_6 RTCIO4 VDET1 P18
11 GPIO35 ADC1_7 RTCIO5 VDET2 P17
8 GPIO39 ADCPA ADC1_3 RTCIO3 SensVN P16
7 GPIO38 ADC1_2 RTCIO2 SensCN P15
6 GPIO37 ADC1_1 RTCIO1 SensCP MISO P14
5 GPIO36 ADCPA ADC1_0 RTCIO0 SensVP P13
Only Input pins! No pullup/pulldown internal resistance
pycom
Hack your LoPy4
Connect to a 10nF capacitor to enable Touch Pin function

Figure 3.3: Lopy 4 pinout diagram

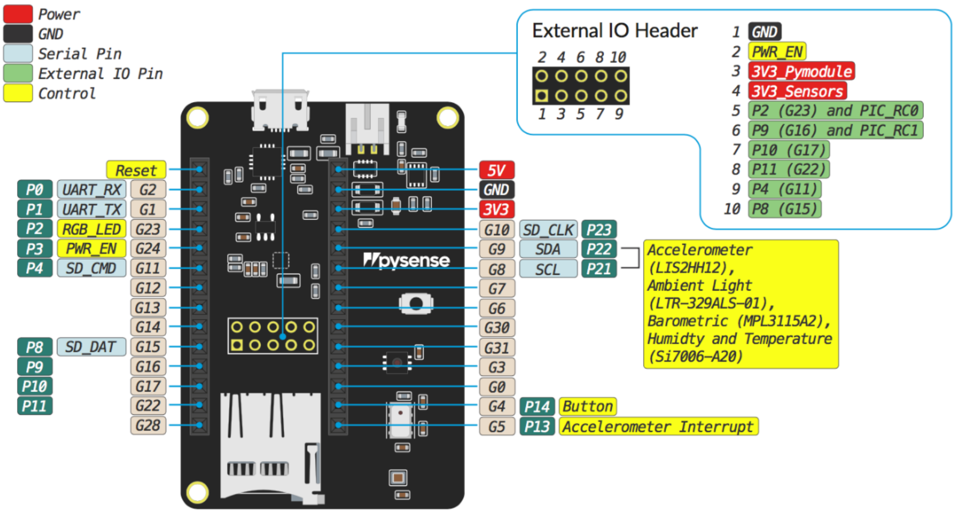


Figure 3.4: Pysense pinout diagram

this option was quickly discarded when evaluating under the oscilloscope, and due to the high readings of current being drawn from the battery, it was concluded as impossible to work with the GPS module as-is as no significant longevity would be achieved regarding battery lifetime.

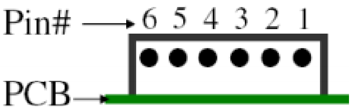


Figure 3.5: GP-735 6-pin interface

Taking this learned fact into account, it was then added a wire to pin 6 to the interface acting as a power control pin, being that GP-735 supports the easy power saving control mechanism. To control the power of the module, pin 6 was connected to the MCU GPIO. In this way, when pulling the pin to zero, it was possible to maintain the module to a minimum level of battery drain, while only keeping RTC and RAM. Moreover, this allowed a normal run when pulling the pin at one or by simply leaving it floating to which situation becomes fully powered.

The GP-735 module connected to the sensor via UART TTL serial and taking into account the pin assignment in the module datasheet, the connection to the module was done as described in the table 3.1.

Pin	Name	Function	I/O	Lopy4 Pin
1	GND	Ground	Input	GND
2	VCC	3.1 ∼ 5.5 V power supply	Input	3V3
3	TXA	Port A serial data output (from GPS)	Output	P20
4	RXA	Port A serial data input (to GPS)	Input	P19
5	V_BAT	Backup power (1.5 ∼3.5 V)	Input	Not used
6	PWR_CTRL	Module power control High or Floating: power ON Low: power OFF	Input	P11

Table 3.1: Connection of GP-735 to the Pycom Lopy4

Additionally, the RGB led present in the LoPy 4 which was initially used as a control led, was turned off to give place to a control pin P2. This choice was carried out so it would be possible to visualize the different stages of the run when optimizing the battery with the oscilloscope. Nevertheless, some considerations about battery optimization will be further explained in the manuscript.

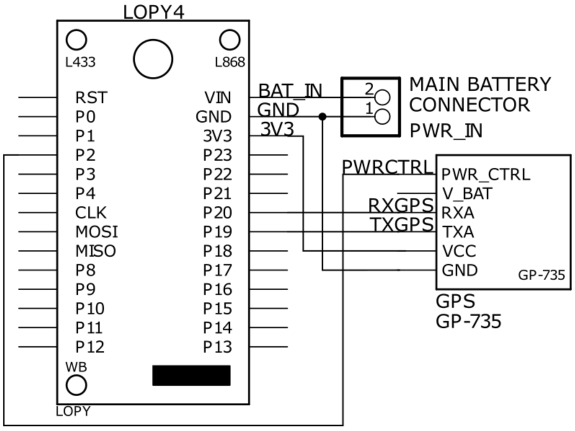


Figure 3.6: Final wiring scheme

The node wiring scheme as explained above and can be seen in fig. 3.6.

3.2.3 Software

Once assembled, the hardware was mounted, where both the node and the gateways used throughout this thesis were constantly flashed and updated with the latest firmware provided in `https://docs.pycom.io/gettingstarted/installation/firmwaretool` since PyCom is constantly making improvements and adding new features to the devices. Both Lopy devices were programmed in MicroPython programming language which is an efficient implementation of the Python 3 programming language, ensuring to include a small subset of the Python standard library while being optimized to run on MCUs. While developing the software that was later implemented on the nodes and gateways, used libraries can be classified into system libraries and support libraries. Former ones were concerning with the internal function of the MCU and the Pysense shield while the latter concerning with support instructions. The most relevant libraries used were the ones that provided the basic functions for the boards, e.g: pycom, pysense, machine, etc... These are seen as the most relevant because without them all basic functions would have to be programmed from scratch being the rapid development and deployment impossible to achieve. Relating with support libraries used, the most relevant was the ubinascii[3] and the struct[4] libraries since these libraries provided the basis to the encoding of the payload and to maximize its efficiency in terms of size. JavaScript was used punctually in a script to decode the received payload in the The Things Network (TTN), allowing a possible stakeholder to use an identical solution, and visualize information in a clear way instead of sole HEX values. Additionally, Python 3 was used to process the information collected with the transmissions made between sea node and the coastal gateways to process several ranging methods and estimating the final estimated coordinates.

3.2.4 Algorithm

Throughout this section, the algorithm developed for the development of the node and coastal gateways are described, as well as the different considerations that took place in this phase.

In fig. 3.7, the full fluxogram used in the development of the node is shown. Inside the board, there are two important files that MicroPython will look for in the root of its filesystem. These files contain MicroPython code that will be executed whenever the board is powered up or reset. Thus, in every execution the board boots by running the *boot.py* file this file should contain low-level code that sets up the board to finish booting. In the Lopy, 4 the boot file was used to turn off some of the functions enabled by default, that would not be used and would inevitably drain battery later. Thus, the RGB led, WIFI and the Bluetooth peripherals were turned off. Furthermore, when booting the Lopy, it either creates or reads from the internal memory, the ID of the current execution. Performing the setup now in the *main.py* file where the main script is located, relates to the need of importing libraries, declaring global variables and performs the LoRa setup. The setup of LoRa consisted of initializing the LoRa in LoRaWAN mode, creating the OTAA authentication parameters and defining the channel frequency, lastly joining a network using these same parameters while defining a timeout for the join to take place. Also, a non-blocking LoRa socket defined to send confirmed messages, the LoRa setup used both for tests and final proto-

[3] Ubinascii module implements conversions between binary data and various encodings of it in ASCII form (in both directions).
[4] Struct module performs conversions between Python values and C structs represented as Python strings.

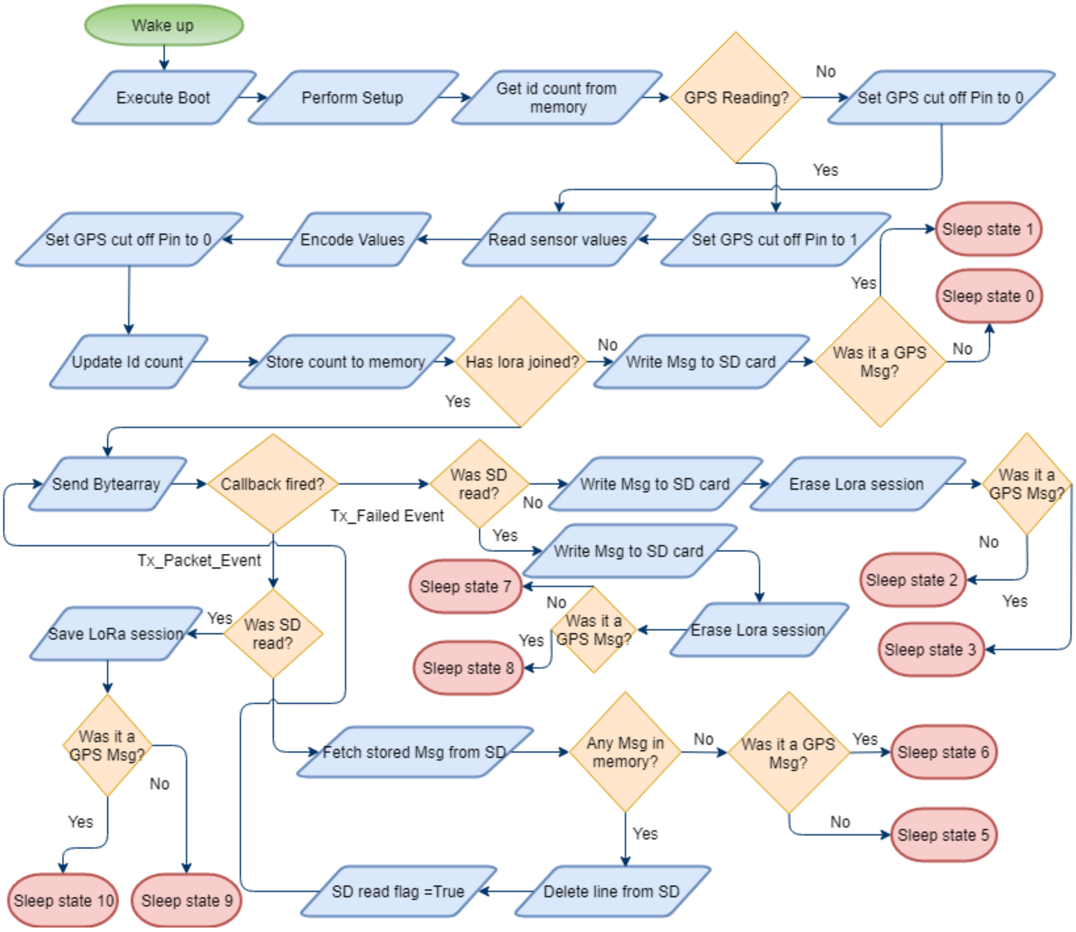


Figure 3.7: Final node fluxogram

typing is further explained in the study setup section.

From this point, the main cycle reads the ID count from the internal memory and checks if it is going to take a regular telemetry message or a GPS message, this is based on counting the ID as a result of the battery optimization section below. Once the values are read and encoded in a byte array the GPS pin is cut off so no unnecessary battery is drained from it. After the message is prepared, the LoRa network status is re-checked again, at this point if LoRa has successfully joined a network the message is sent or stored to the SD card otherwise. Upon sending a message, LoRa class-A device opens two receive windows at specified times (1s and 2s) after an uplink transmission. As a matter of serving the purpose to handle ACK messages, a callback handler for the LoRa radio was implemented.

Lora Callback

After sending one message and receiving a LoRa.TX_PACKET_EVENT trigger, the callback proceeds to check if the SD card has already been read, if not, and since the sensor is in range of a gateway, the message in the SD card is read, deleted from the SD card and sent. Finally, after the send instruction, another callback in the queue is fired and the message is deleted. Should the transmission fail, the system proceeds to store the message to the SD card, otherwise, the sensor goes to deep sleep.

Message Payload

Relying entirely on telemetry, and estimating the position and multilateration, might work when the sensor is near an on-shore gateway, i.e near the coast of the island. However, there is still the need to locate or at least, know the position of the sensor when out of range of a gateway. Although relying entirely on GPS readings would solve the problem of accurately locating the positions where the sensor has been, this would approach would rapidly drain the battery. To cope with this situation, the sensor forms two types of payloads to be sent to the coastal gateways. One payload is formed by the telemetry given by the PyCom and PySense shield, and the ID of the message. The other payload is formed by the latter integration with a GPS reading, adding latitude, longitude, altitude and time (hh:mm:ss). This way, rearranging a different payload to be sent was inspired by the various bio-telemetry systems[5], which add GPS corrections to the telemetry produced.

Encoding the payload was done using the *struct* library given by MicroPython, which was extremely helpful when maximizing payload efficiency. One approach initially performed, was to append a header byte before a field, to give the knowledge to the server how the information processed was structured, and to be able to decode the payload. This header would dedicate 4 bits to indicate the type of information and 4 bits to indicate the size of the information to decode, as depicted in fig. 3.8.

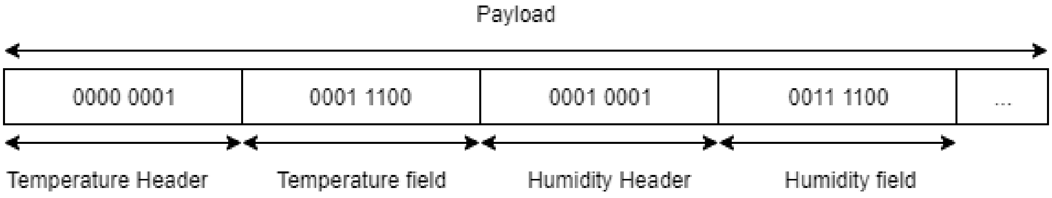


Figure 3.8: Initial approach to payload encoding

This method improves the robustness of the whole system, by removing the need of the server to know beforehand the format of the payload and to be able to decode it. However, as tested, this method represented a drastic increase in complexity in the sensor side. Encoding each field would result in a dynamic payload in every transmission, thus draining the battery as a result of the increasing amount of made calculations. The priority was to leave the critical side, i.e the sensor, as ”light” as possible in terms of calculations made, and time spent for these. To cope with the previous experience, the payloads were

[5] https://www.microwavetelemetry.com/avian_transmitters

encoded into a byte array using the *struct.pack* method, using a static format given both to the sensor and the server. This method packs all the fields into bytes of a given size according to a given format. This way, each payload will have the same size, 32 bytes for a payload with GPS reading and 17 bytes for a payload with regular telemetry. This method has brought advantages by reducing the time and complexity of encoding the payload, while also reducing the size of the message for transmission. The resulting payloads can be seen in fig. 3.9.

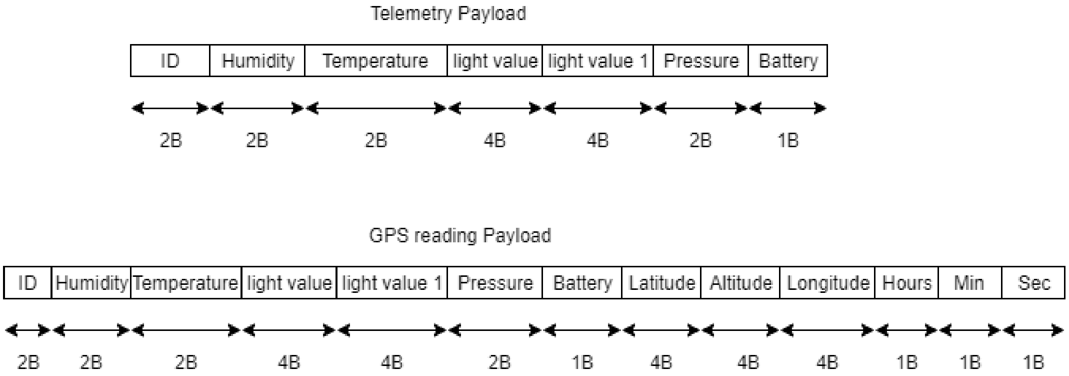


Figure 3.9: Resulting message payloads

Adaptive sleep

The concept behind all the terminating states depicted in 3.7 was to create a sensor with the capability of adapting it's deep sleep time to any group of events occurred while executing the main cycle. Thus, each state represents a different collection of events, depending on the LoRa joining procedure, the payload type, the events received by the callback on the first and second transmission, and lastly, by checking if there's any message stored in memory. These collections can be seen below in table 3.2 and serves as a way to clarify each process forming a state.

State	Lora Joined	Reading	1st Transmission	2nd Transmission	Messages in mem
0	No	Telemetry	N/A	N/A	N/A
1	No	GPS	N/A	N/A	N/A
2	Yes	Telemetry	Tx_Failed_Event	N/A	N/A
3	Yes	GPS	Tx_Failed_Event	N/A	N/A
4	Yes	Telemetry	Tx_Packet_Event	N/A	No
5	Yes	GPS	Tx_Packet_Event	N/A	No
6	Yes	Telemetry	Tx_Packet_Event	Tx_Failed_Event	Yes
7	Yes	GPS	Tx_Packet_Event	Tx_Failed_Event	Yes
8	Yes	Telemetry	Tx_Packet_Event	Tx_Packet_Event	Yes
9	Yes	GPS	Tx_Packet_Event	Tx_Packet_Event	Yes

Table 3.2: State specification

Each state, depicted above, served also as a basis for the battery optimization section and illustrates that each behavior will have different impacts on battery consumption.

Timeouts

Regarding the timeouts in the code, several tests took place to achieve the best timeout possible for the LoRa to join a network. Also, this assured the amount of time that GPS needs to acquire a position. These tests were performed due to the fact that it would not be possible to be constantly trying to join a network (even when out of range from a gateway) or trying to acquire a GPS position (when not possible, e.g. due to obstructions) as these are two of the most battery consuming points in all parts of the algorithm. As a result of a 24h test running the sensor, joining a LoRa network while marking these operations with the RTC, the best timeout found for joining the network was determined to be 15 seconds. In the case that it did not join, all of the telemetries are stored on the SD card for later transmission. The GPS module was given 30 seconds of time-to-fix, which corresponds to the cold start time given by the manufacturer in `https://cdn.sparkfun.com/datasheets/GPS/GP-735T-150203.pdf`. This ensured that the module is capable to acquire a position. The reason why the cold start time was followed was given to the fact that this algorithm is at the end of each execution placed into a deep sleep state for the duration, which was set previously and a sensor wake-up will result in a system restart which will cause the GPS to loose its RTC and RAM states.

Storage

For this sensor, a 16 GB SD card was used. Furthermore, the messages stored in the SD card were GPS plus telemetry with a total of 34 bytes and sensor telemetry ones with a total of 19 bytes leaving aside the need to create an efficient way to manage storage memory for a period of 3 months of execution (see section 3.1.1). The accesses made to the SD card followed a Last In First Out (LIFO) policy, giving the priority of transmission to the oldest messages.

3.2.5 System Architecture

Once the node's software was implemented, the development of the gateway began as well as establishing the conditions for the network server to receive the messages and forwarding them to the TTN API.

Gateways

For the gateways used in this project, the Pycom Lopy was once more used. Pycom gives multiple libraries related to LoRa in `https://docs.pycom.io/tutorials/lora/`. The LoRaWAN nano-gateway library allows the Lopy to connect to a LoRaWAN network such as the TTN. The code present in the Gateway is split into 3 files, *main.py*, *config.py* and *nanogateway.py*. These are used to configure and specify how the gateway will connect to a preferred network and how it can act as a packet forwarder[6].

Each gateway on TTN has a Gateway ID and a sample code is provided that creates a unique Gateway ID using the unique identifier (MAC) that's built into the Wi-Fi network adaptor on all Pycom devices. The resulting ID will be e.g. 240ac4FFFE008d88 and will be used in the network provider configuration. After calculating the ID, the *main.py* runs and calls the library *config.py* file which contains the settings for the server and network it is connecting to. Depending on the nano-gateway region and provider, these configurations may vary. In this project, the configuration was tested using the TTN router.eu.thethings.network in the European 868Mhz, region to initialize the gateway. Once the gateway is started, the nano-gateway library controls all of the packet generations and forwarding for the LoRa data. This does not require any user configuration and the latest version of this code should be downloaded from the Pycom GitHub Repository[7].

Registering the gateways in TTN

At this point, it is necessary to register the gateways configured in the TTN. After creating an account, one registers a gateway with the Gateway EUI which is the Gateway ID obtained above, defines the frequency plan (Europe 868 MHz) and defines the router for the gateway to establish the connection and transfer the data to the rest of the network through the UDP protocol. Once this configuration is performed, the gateway is now configured and the node was registered and configured to connect to the TTN using OTAA as it provides a more secure approach because the activation of the device is confirmed, and also because the session keys are negotiated with every activation. At this point, the remaining task was to create an application inside the TTN. The application generates all the above security keys and dictates what happens to the LoRa data once it is received by TTN. Some several different setups/systems can be used, however in this project persistence of data was seen as a priority. This is in order to be able to access the data, study integrated the data storage service given by TTN, which allowed to store the data in JSON format for a period of 7 days, an example of one of this uplink messages can be seen in Appendix A. The architecture of the system may be seen in fig. 3.10.

[6] A LoRa packet forwarder is a program running on the host of a LoRa gateway that forwards RF packets received by nodes to a server through an IP/UDP link, and emits RF packets that are sent by the server.

[7] `https://github.com/pycom`

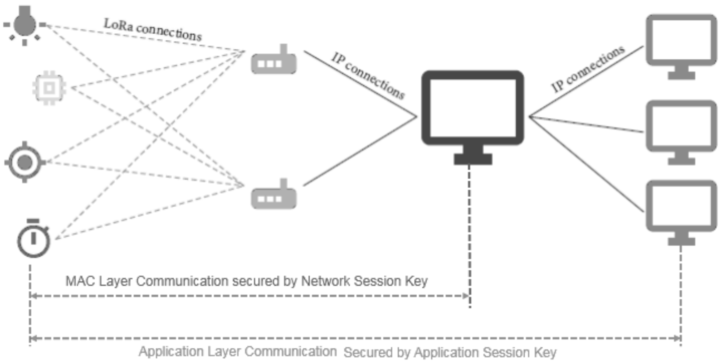

LoRa connections
IP connections
IP connections
MAC Layer Communication secured by Network Session Key
Application Layer Communication Secured by Application Session Key

Figure 3.10: System architecture

From the architecture, the node broadcasts the previously shown payloads at a low frequency over a determined distance. The LoPy gateway was used to receive LoRa packets from the end node, acting as a forwarder to the network server through IP connections. The network server will manage the entire network. When a packet is received, the redundancy is removed and a security check is performed. Lastly, the network server delivers the data to the API where data can be decoded and processed.

The communication in LoRa protocol is best depicted in fig. 3.11 consisting the LoRaWAN protocol of a MAC layer that serves an application layer based on the LoRa physical layer.

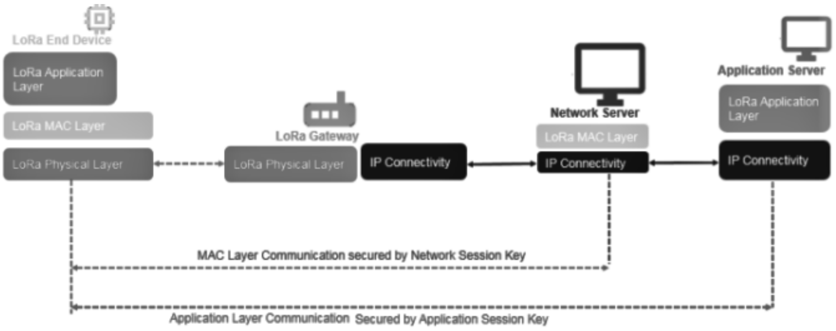


Figure 3.11: Lora architecture

Additionally, below, in fig. 3.12, it is provided an overview of the LoRa frame structure, to give perception where the different LoRa setups are used in the frame structure.

In the physical layer, the LoRa frame starts with a preamble. This defines the packet modulation scheme being modulated with the same spreading factor[8] which was configured in the code. While in the Physical Layer (PHY), the header contains information such as payload length and the data rate. The PHY Payload contains the whole MAC frame. The packet that reaches is processed in the MAC layer consists of a MAC header, a MAC Payload and a message integrity code. The MAC header contains information regarding protocol version, type of frame (data or management), uplink or downlink

[8] The spreading factors represent the duration of the chirp. SF7 represents the shortest time on-air, SF12 will be the longest

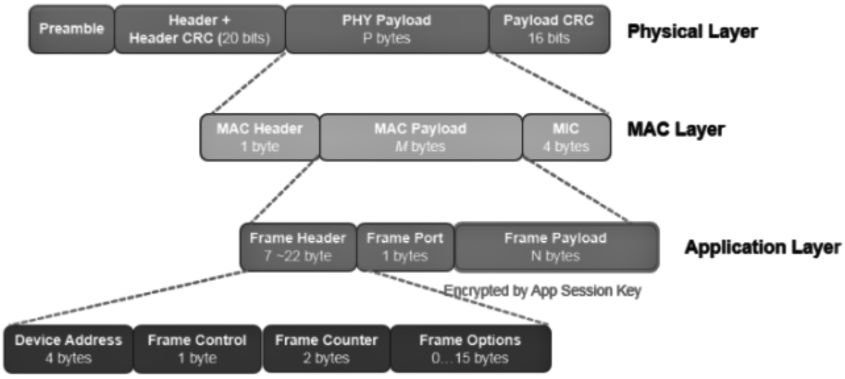


Figure 3.12: Lora packet structure

transmission, and the need for the frame to be acknowledged. The payload is in the MAC layer initially, and the join procedure is composed of a join request message and a join accept messages. Then, in the application layer, The MAC Payload consists of a frame header, port, and payload. The frame header contains fields of control, such as a device address to identify the network and the device, the frame counter for sequence numbering, and the frame control along with the frame options. Although in this project, the adaptive data rate was not used, as it lets the LoRa server control the data rate and transmission power, for a reason of stability in the transmissions these fields were of no use. Lastly, the frame payload is where the previous payloads sent by the end-node are located and encrypted with the previously explained App Session key.

Decoding the payload

Once the payload was successfully received, it was presented in the TTN console, depicted in the hexadecimal representation of the decrypted binary data. Given that humans cannot easily parse hexadecimal values, the last step was done to program a script inside of the TTN to decode and reverse thee encodings used by the node. By knowing the formats used in the encoding, it became relatively simple to return to the original values taken by the node. For easy understanding, the used decoder is presented in appendix A.

3.3 Study Setup

To test our project for battery optimization and position estimation, two separate tests and setups were done. To test for power consumption, the final prototype was used along with all the architecture above explained using the oscilloscope as a resource for measurements and calculations. For position estimation, a raw LoRa setup was used, instead of using LoRaWAN. Being the objective, gather data on the received power per transmission there was no need for an internet connection and use of encrypted messages. Thus a simple node-to-node communication was used by reading and transmitting values from the node to the 4 gateways, storing the values in the SD card for post-processing.

3.3.1 Battery optimization tests

To optimize the battery drained by each execution and to calculate an appropriate sleep time for each state depicted in table 3.2, the node was connected in series with a 1 Ω resistor (see fig. 3.13), to obtain an average value of the current drained per execution.

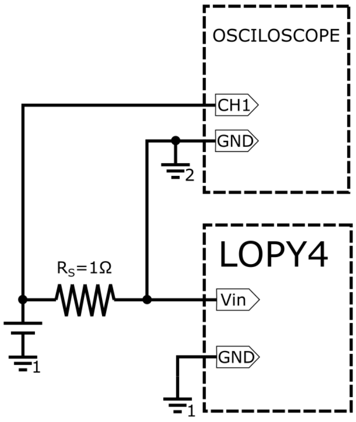


Figure 3.13: Battery optimization circuit

The LoRaWAN setup used in the battery optimization tests was the setup intended for deployment of the final version of the prototype, tested with OTAA and configured for acknowledgment of every uplink message (see table 3.3).

Parameter	Value
Region	EU868
Frequency	868 MHz
Bandwith	125 KHz
Transmitted Power	14 dBm
Spreading Factor	7
Device Class	CLASS_A
Security	OTAA

Table 3.3: LoRaWAN Setup

Furthermore, the workflow of these tests consisted of connecting the node to the oscilloscope, initiate execution with a preconfigured state, and execute transmissions by controlling the power of the gateway. This was to control the transmission status of each message and to allow the current to be visualized in each state. Further calculations and assumptions are explained in the evaluation chapter.

3.3.2 Position estimation tests

For estimating positions, as previously mentioned, there was no use for encryption, or an API, as these tests were done with the single interest of obtaining the RSSI values per transmission. 4 gateways were placed on top of a PVC pole with an average height of 3 meters. The node was attached to the mast of a vessel transmitting the GPS, used for obtaining the ground truth for later processing. Both emitter and receivers were configured with the LoRa setup depicted in table 3.4.

Parameter	Value
Region	EU868
Frequency	868 MHz
Bandwith	125 KHz
Transmitted Power	14 dBm
Spreading Factor	7
Device Class	CLASS_A

Table 3.4: LoRa Setup

After the planning and configuration phase took place, the computing on the RSSI values was performed offline. Because the RSSI values are in a logarithmic scale, two approaches can be taken to derive a linear equation of the data: 1) turn the RSSI into a linear scale or 2) turn the distance into a logarithmic scale. The first approach was used and the distance *d* and is calculated with the equation 3.1 presented below [47].

$$d = 10^{(RSSI/10)*n} \tag{3.1}$$

This equation uses the RSSI values, the distance d in meters and a tuning parameter n. The workflow in the location estimation tests was done as follows:

- Apply an average of the RSSI values per position as RSSI values can be influenced by the surrounding environment (even the ground truth position can suffer signal variation). This action smooths the data for modelling and increases the precision of the ranging method.
- As for ranging methods, both the free-space and log path models were used. In addition, linear regression was used to compute the distance vs linear RSSI.
- Use linear regression with the RANSAC method, given its easy implementation and improved robustness against outliers in the data.
- Apply bilateration using the best model for the circles calculated in the ranging phase.
- From the two possible bilateration solutions, choose the location calculated at sea, discarding the on-shore one.

As stated in the related work chapter, higher degree multilateration would result in better location estimation. This is usually true, however, the use of multilateration was not possible due to the gateways being aligned in an almost straight line with small curvature in-land. When ranging, and if the circles do not intersect, this causes the multilateration equations to determine positions equally apart from the circles, although with symmetrical orientation relative to the straight line. This increases the error of the model. Stated this, we chose to use a less accurate solution although more stable: bilateration. In fig. 3.14 one can observe the final setup of both node and gateways.

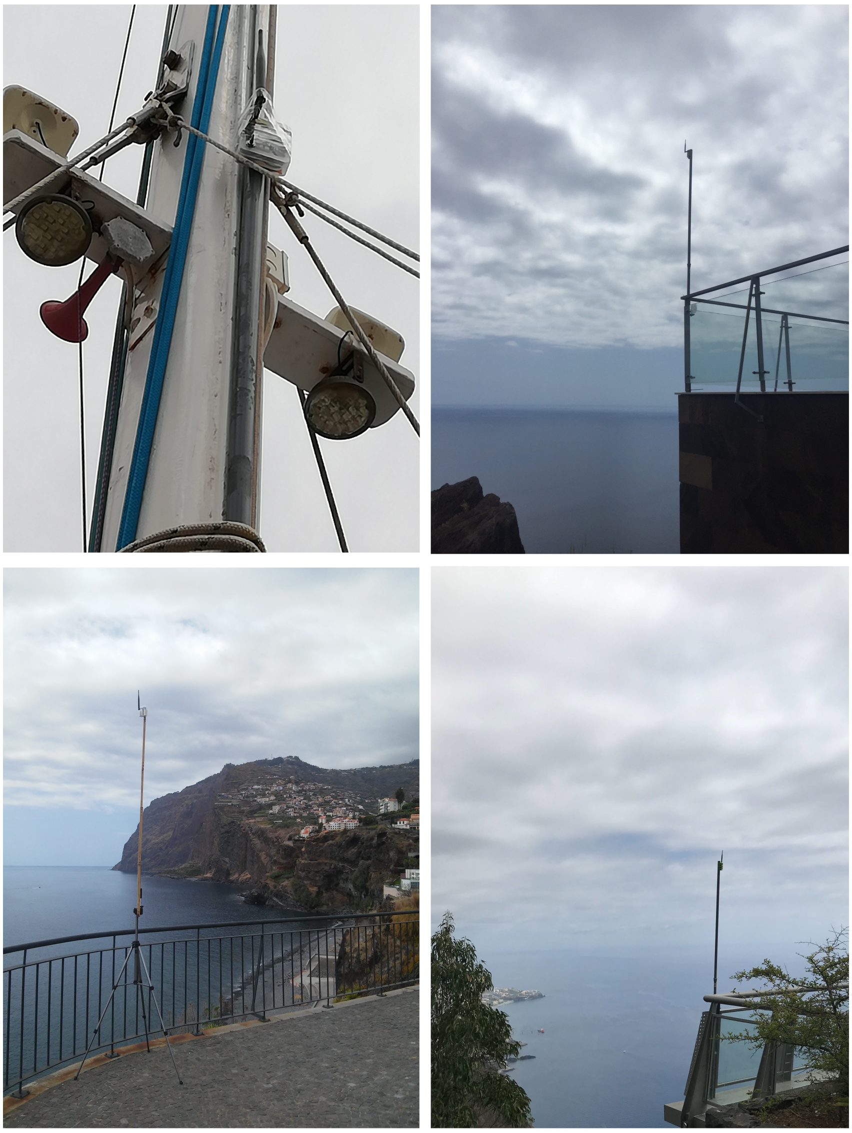


Figure 3.14: Upper left: node; Upper right: gateway 1;
Bottom left: gateway 2; Bottom right: gateway 3;

4

Evaluation

Contents

The following section presents the results as well as an overview of the experiments made with the aforementioned system. These results were obtained empirically evaluating the localization system in-situ, while also testing the node to maximize its lifespan.

4.1 Power Consumption

In terms of achieving the best levels of power consumption and consequently the best longevity, one has to compromise the number of cycles that produce telemetry, and GPS readings in favor of adequate deep sleep time, reaching a balance resulting in the best duty cycle time.
What was observed, and as expected, is that GPS readings, as well as procedures relating to LoRa, were responsible for the major part of the current consumption. However, being able to operate in the 3 months threshold, this was achieved with a very satisfactory frequency of readings and transmissions. Furthermore, one of the reasons that were in the root of working with LoPy 4 was the announced low consumption currents. The latter was proven to be very effective in maintaining the balance between active and sleep current, and there is nothing that would stop this system to be designed for a threshold superior to 3 months for a very affordable price.

This test was performed for the smallest battery available at the moment, which was an 1800 mAh LiPo battery. Furthermore, tests were made to fit the states presented in the algorithm (see fig. 3.7), with the behavior presented in table 3.2. The first criteria were to leave a security margin for the available battery levels. As a consequence of this criteria, the system was designed to use 80% of the battery, as a result, the calculations made fit the equivalent to a 1440 mAh battery. As 3 months, represent 2160 hours of consumption, the average current drained by the system must correspond to 0.66 mA. By measuring, the deep-sleep which corresponded to 60 μA and by establishing this balance, the values were related in equation 4.1 formed by the values of current consumption and to discover the percentage of the time, both in current consumption as well as in sleep consumption, forming the whole duty cycle.

$$0.66 = \overline{I}_{State_i} \cdot t + \overline{I}_{Sleep} \cdot (1 - t) \tag{4.1}$$

Furthermore, each measurement related to the 10 states previously depicted in table 3.2 can be seen in fig. 4.1 being the top left figure a depiction of state 0 and the below figures the following states. Furthermore, in the image below the y-axis represent the levels of measured current in mAh, and the x-axis the divisions of time for each cycle.

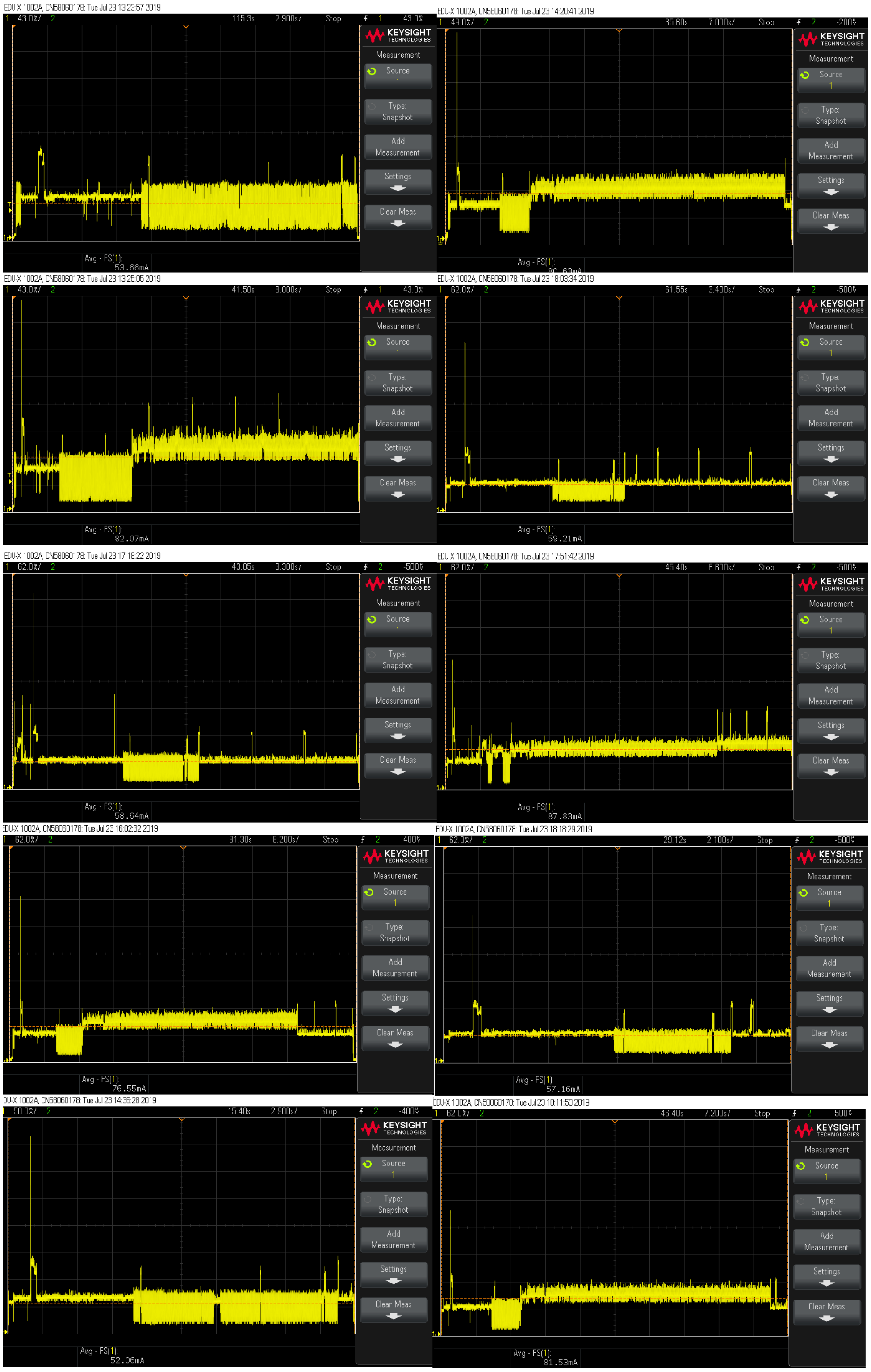

EDU-X 1002A, CN58060178: Tue Jul 23 13:23:57 2019
1 43.0 / 2 115.3s 2.900s/ Stop 1 43.0
Avg - FS(1): 53.66mA
EDU-X 1002A, CN58060178: Tue Jul 23 14:20:41 2019
1 49.0 / 2 35.60s 7.000s/ Stop 2 -200
Avg - FS(1): 80.63mA
EDU-X 1002A, CN58060178: Tue Jul 23 13:25:05 2019
1 43.0 / 2 41.50s 8.000s/ Stop 1 43.0
Avg - FS(1): 82.07mA
EDU-X 1002A, CN58060178: Tue Jul 23 18:03:34 2019
1 62.0 / 2 61.55s 3.400s/ Stop 2 -500
Avg - FS(1): 59.21mA
EDU-X 1002A, CN58060178: Tue Jul 23 17:18:22 2019
1 62.0 / 2 43.05s 3.300s/ Stop 2 -500
Avg - FS(1): 58.64mA
EDU-X 1002A, CN58060178: Tue Jul 23 17:51:42 2019
1 62.0 / 2 45.40s 8.600s/ Stop 2 -500
Avg - FS(1): 87.83mA
EDU-X 1002A, CN58060178: Tue Jul 23 16:02:32 2019
1 62.0 / 2 81.30s 8.200s/ Stop 2 -400
Avg - FS(1): 76.55mA
EDU-X 1002A, CN58060178: Tue Jul 23 18:18:29 2019
1 62.0 / 2 29.12s 2.100s/ Stop 2 -500
Avg - FS(1): 57.16mA
EDU-X 1002A, CN58060178: Tue Jul 23 14:36:28 2019
1 50.0 / 2 15.40s 2.900s/ Stop 2 -400
Avg - FS(1): 52.06mA
EDU-X 1002A, CN58060178: Tue Jul 23 18:11:53 2019
1 62.0 / 2 46.40s 7.200s/ Stop 2 -500
Avg - FS(1): 81.53mA
KEYSIGHT TECHNOLOGIES
Measurement
Source 1
Type: Snapshot
Add Measurement
Settings
Clear Meas

Figure 4.1: State power measurements

As a complement to the measurements in the figure above, and by working with the equation presented the following results were derived and presented in table 4.1.

State	Active time (%)	Inactive time (%)	Active current (mA)	Cycle time (s)	Sleep time (h)
0	1.1	98.9	53.66	28.6	0.714
1	0.72	99.28	82.07	80	3.06
2	1.01	98.99	58.64	33.05	0.9
3	0.77	99.23	76.55	80	2.8638
4	1.14	98.86	52.06	29.05	0.7
5	0.73	99.27	80.63	69.7	2.63305
6	1	99	59.21	34.05	0.93638
7	0.67	99.33	87.83	84	3.45
8	1.04	98.96	57.16	21.25	0.5619
9	0.72	99.27	81.53	72	2.72

Table 4.1: Current consumption values per state

By obtaining the current consumption per measure and in function of the time of the whole cycle per state, it was possible to derive several measurements that the system will be able to send both in the best and worst-case scenarios. This was done by dividing it for the whole capacity of the battery. Table 4.2 depicts the best and worst-case scenarios as well as the number of transmissions achievable with any one of the two cases.

State	Duty-Cycle (h)	Nb of TX p/ day
5 (Best Case Telemetry)	0.71	18
6 (Best Case with GPS)	2.66	4
7 (Worst Case Telemetry)	0.95	12
8 (Worst Case with GPS)	3.48	3

Table 4.2: Number of transmissions p/day based on best and worst scenarios

These tests, allowed the system to be tuned-up in terms of deep-sleep time and also in terms of conditioning the GPS reading to happen. As an experimental test the system was left to run, constantly with these values of sleeping time with a fully charged 1800 mAh battery for a whole month, in addition to vary the scenarios the gateway was often turned off, to simulate an out of range situation and to allow us to test various scenarios and general sensor behavior. At the end of the month of the experimental test, the transmissions took place and the sensor behaved as expected in every situation ending the battery with a 3.7 voltage indication. The system was designed for 3 months, it might endure more time depending on which behavior the sensor presents, but further conclusions in this matter will be pure assumptions with no basis of proof.

4.2 RSSI Distance Estimation

The measurements for distance estimation were done in-situ and located on the south coast of Madeira island. Although four gateways were placed inland, two of them failed due to heat exposure and were unable to capture the greatest part of transmissions, which forced us to work with the data received by gateway 2 (green) and gateway 3 (yellow). However, the full boat trip (total of 1482 points), as well as the location of the gateways, is depicted in the figure below (see fig. 4.2).

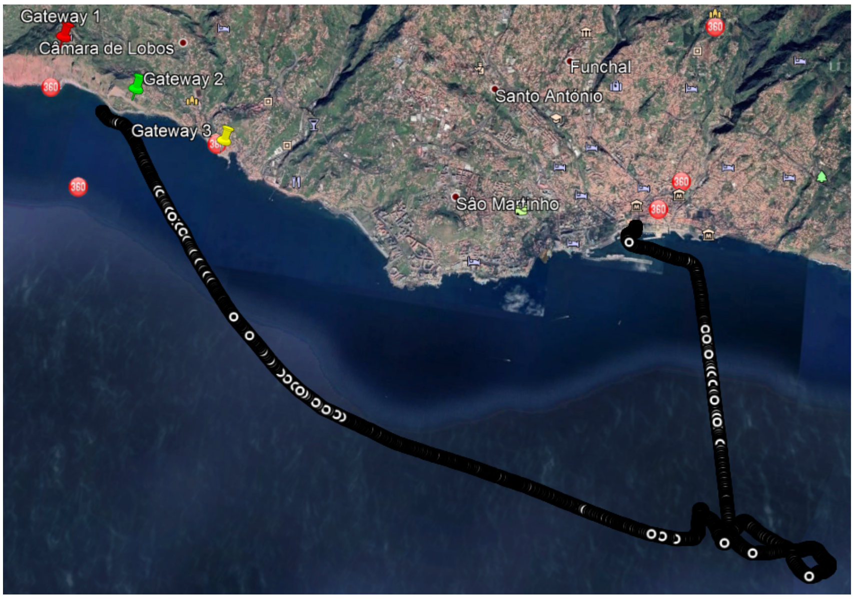


Figure 4.2: Full trip and gateway location

Once the trip was concluded, the data was processed offline. Although before applying the models, the data gathered by each gateway was analyzed individually. Since RSSI measurements can include extreme levels of noise, due to some signal disturbances explained in chapter 2, some pruning of the data was required before analysis. Extremely high values of distance estimation ($\geq$ 10 km) were removed. Also, residual faulty values that did not correspond to the true GPS reading were also excluded. Apart from these pruning, the data was treated to depict a real-world application of a RSSI based positioning system.

The mean error per measurement, as well as the parameters used in each model, can be seen in table 4.3. In addition, by applying the path-loss models to the full data set, it is possible to observe, that the log-path model outperformed the free-space model approximately by a factor of 1.15 in gateway 2 and of 1.5 in gateway 3.

Gateway	Model		Error (m)	
	Log-Path Exp	Free-Space	Log-Path	Free-Space
2	(exp = 2.5)	λ = 0.3453830161	1160	1325
3	(exp = 2.4)	f = 868000000	960	1360

Table 4.3: Model parameters and mean error for distance estimations.

Furthermore, figures 4.3 and 4.4 depict the average error between the applied models as a function of the distance to the gateways. In these figures, the free space model error can be seen as the blue line and the log-path error as the red dotted line.

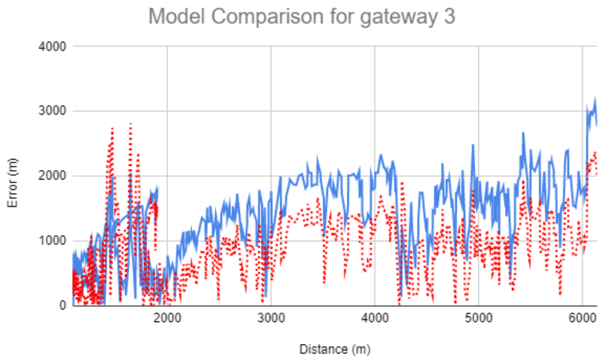


Figure 4.3: Gateway 3 average error per model

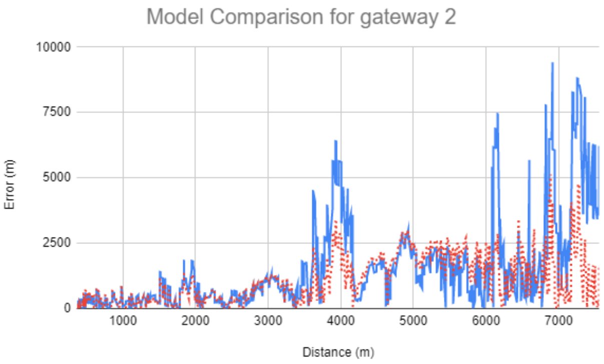


Figure 4.4: Gateway 2 average error per model

One notable difference between the two shown graphs is that gateway 2 shows a much more stable error throughout the time. This is because the sensor was most of the time obstructed by the mast which provided noisy readings throughout the time, and therefore with lesser oscillations. Gateway 3 was intermittently in LoS and non-LoS conditions, which in average provided better values, although with more oscillations.

Leaving aside the propagation loss models and in an attempt to provide a better distance estimation, the RSSI and distance values were modeled by the application of linear regression. The linear regression was modeled using the following as function of distance: (1) the RSSI in logarithmic scale; (2) the linearized RSSI; and (3) the averaged RSSI used in (1) to smooth oscillations of RSSI values that correspond to the same ground truth position. This way, it was possible to obtain a solution that was better on average for the whole set. Table 4.4 depicts the modeled data and the parameters encountered.

Gateway	**Linear Model**	**Parameters**			**Error (m)**
		a	b	R^2	
2	RSSI	-148	-10620	0.748	995
	Linear RSSI	-77777	6709	0.607	1420
	Avg RSSI	-153	-11143	0.775	945
3	RSSI	-220	-16009	0.721	1415
	Linear RSSI	-147244	10901	0.635	1749
	Avg RSSI	-223	-16314	0.731	1180

Table 4.4: Linear model ax+b for the different grouped data.

The best model obtained when applying the linear regression to the grouped data as depicted above, corresponds to the linear regression that weighs the averaged RSSI with the distance to each gateway. However, at this point, the mean error per measurement has not suffered any substantial decrease. By looking at figure 4.5 and apart from the fact that gateway 2 best fits its equation, one can still observe that each model still presents many outliers, that negatively influence each set.

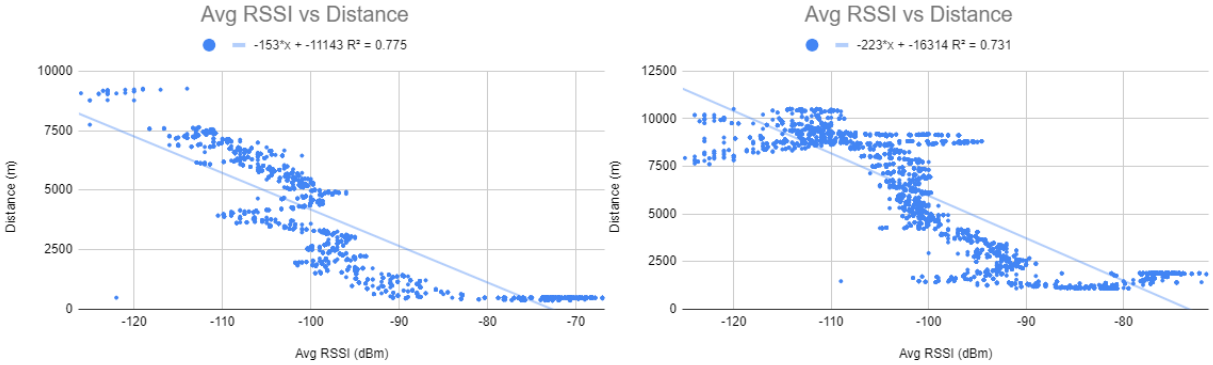


Figure 4.5: Left: Avg RSSI in function of distance of gw 2; Right: Avg RSSI in function of distance of gw 3;

Due to the presence of outliers, the RANSAC (RANdom SAmple Consensus) was applied. This method is a general parameter estimation approach designed to cope with the presence of outliers in the input data. Furthermore, this method generates candidate solutions by using a minimum number of observations [48]. The maximum residual/threshold for a data sample to be classified as an inlier was the MAD (Median Absolute Deviation). From figure 4.5 above, it is possible to see a steady progression of the RSSI values for gateway 2 than gateway 3, although with much more oscillations present in the latter case. These oscillations are most likely resultant from the placement of the node in the boat which was placed behind the main mast of the vessel, and also due to the placement of land nodes, which can be seen in figure 4.2, which contained mountainous, rural and semi-urban terrain obstructing the transmissions LoS. The figure 4.6, shows the respective improvement of the linearized model presented above, as well as the respective inlier and outlier points.

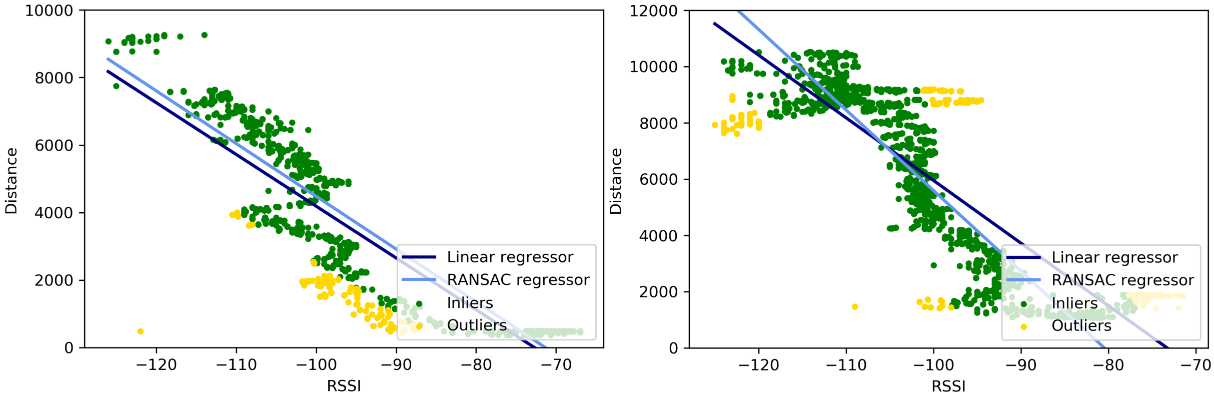


Figure 4.6: Left: Ransac regressor for gw 2; Right: Ransac regressor for gw 3;

Moreover, table 4.5 depicts the improvements weighing the averaged measures of RSSI, which was the best linear model obtained with the least mean error per measurement, and using the RANSAC.

Gateway	**Linear Model**	**Parameters**			**Error (m)**	**Min (m)**	**Max (m)**
		a	b	R^2			
2	Avg RSSI	-153	-11143	0.775	945	0.8	8308
	RANSAC	-156.22	-11140	0.857	732	2.4	2891
3	Avg RSSI	-223	-16314	0.731	1180	0.4	6520
	RANSAC	-286	-23030	0.837	885	5.1	4823

Table 4.5: Error comparison between model and RANSAC analysis.

In the table above, both minimum and maximum errors when ranging are presented. When referring to the LoRa context and in this specific ocean setting, these mean errors have a relatively low impact when comparing with the ground truth distance measured.

Gateway 2 minimum value represents a 99% certainty, as its maximum value refers to an approximately a 50% certainty. Gateway 3 minimum value represents a 99% certainty, as its maximum value refers to an approximately 70% certainty. Moreover, the RANSAC analysis improved the data set by a factor of 1.33 in both gateways. Naturally, the most improved data set was the one corresponding to gateway 3 given that it had more outliers and more dispersed data.

4.2.1 Position Estimation

By knowing the gateway positions in-land and once the estimated distances were calculated, it became a problem to translate the cartesian coordinates into Universal Transverse Mercator (UTM) coordinates. As it was expected, position estimation became, progressively better as the distance to the gateways decreased. In figure 4.7, it is possible to separate the locations estimated into three classes, the first class, represented in red color, corresponds to the points were the LoS was obstructed by the mast of the vessel to both the gateways. This drastically influenced the RSSI values obtained. The red class groups are a set of points that correspond to a precision of approximately 4 km. The yellow class corresponds to the moment where LoS exists for the gateway 3, although remaining obstructed for gateway 2. The yellow class represents a precision of 3 km. Lastly, the green class corresponds to the situation were a clear LoS was achieved for both gateways, and 650 m precision was achieved in the latter. While many estimated points in 4.7 are close to the GPS ground truth, it is possible to see that the data are very much dispersed.

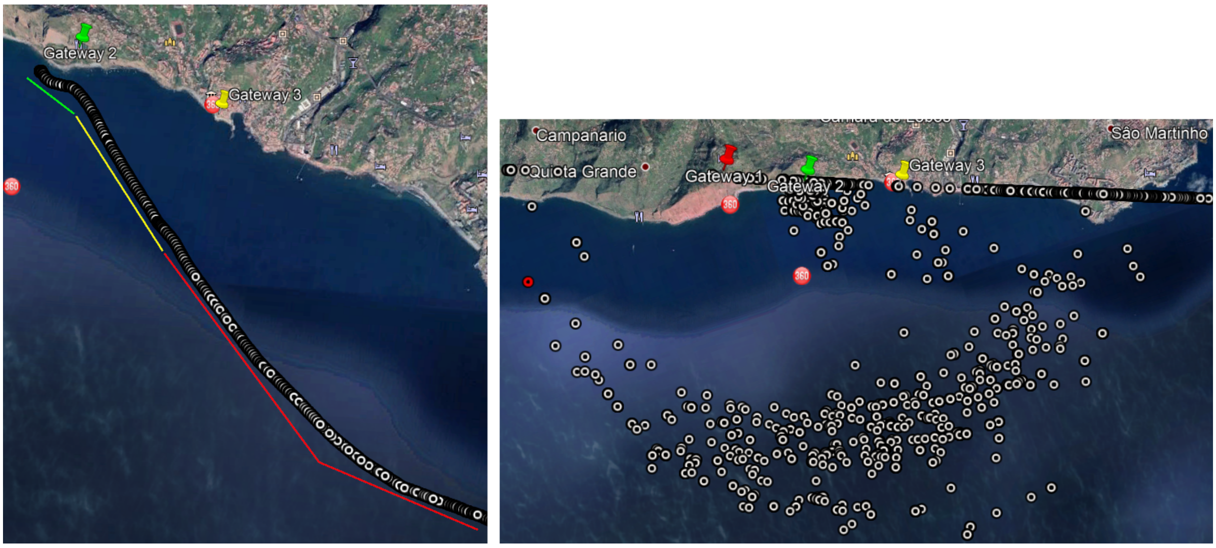


Figure 4.7: Left: Degree of precision depending on the subset; Right: Estimated points of the whole set;

Having larger distances separating the node from the gateways tend to disperse the data due to the nature of the RSSI value that being logarithmic, a small oscillation, corresponds to a large induced distance error. This distance could be further diminished by sampling more RSSI values per location passed. Instead, and because the sea vessel was in constant movement, these values were averaged. Moreover, the situation where LoS is not constantly maintained also contributes to the dispersion of the data.

4.3 Discussion

In this section, we evaluate, analyze and discuss the findings of this study, their contributions to the state-of-the-art and its possible applications.

4.3.1 Research Findings

The research findings presented in this discussion relate to the research questions posed in chapter 1. Answering the questions relating to distance estimation behavior in ocean settings and the overall level of accuracy in position estimation. As to answer the problem relating to the viability of producing a low-cost bio-tag and if the current technologies are sufficient to face the constraints posed in this problem, will take an overall analysis of the topics below discussed.

RQ1 - Is it possible to advance the state of the art in software for tagging seabirds using existing technologies?

One of the features achieved that stands out and elects these types of systems as a viable choice for a bio-telemetry system is the longevity that one can achieve with a small battery. For the tests made with the smallest battery of 1800 mAh, this system could achieve in the best-case scenario a total of 22 transmissions of telemetry per day, with 4 of those yielding a GPS ground-truth measure. In the worst-case scenario, 15 transmissions are sent, being 3 of the total a GPS ground-truth. In terms of frequency of messages and exchange of telemetry, there is a competitive advantage in this system when facing ARGOS which only transmits a total of 14 messages per day. GPS tags provide by far the best accuracy, however, the cost of the tag and the price to pay in power to acquire and transmit is very high, as it was depicted above when studying the power consumption. For what is left, this system presents, the same type of drawbacks that any ground-based system presents. By controlling the duty-cycle, a very satisfying rate of transmission was obtained, while still maintaining the time of experiment to 3 months. To achieve a better resolution (number of samples per time) for a study, and also as a term of scaling the battery capacity up or down, it becomes just a matter of tuning the duty-cycle. As the relation achieved, of number of transmissions per day over the 3 months period of time, was very satisfactory, if this relation is maintained one can improve the number of samples per day by scaling down the overall time of experiment, or either by scaling up the overall time of experiment by decreasing the number of messages per day.

RQ2 - How does distance estimation behave in ocean settings?

One of the focus of this thesis was to use a low-cost and low-power technology solution, to enable the estimation of the position of birds in oceanic environments. This ruled out or at least confined the usage of power-hungry satellite-based technologies but also ruled out more-expensive ToF based technologies. The approaches left to explore were multilateration and multiangulation. Because there was no access to equipment, that would estimate the AoA of the signal accurately and inexpensively, and therefore, the usage of multiangulation was ruled out. As a way of using multilateration, the RSSI values were used by the LoPy to estimate an approximation of the real distance separating the node at sea from the gateways positioned on the coast. Since the RSSI values are inherently sensitive to the environment, as much as the signal is the post-processing of the data still left a lot of noise, as can be seen in figure 4.6. Although, despite the noise, the results presented an average error of 732m and 885m in the two gateways used. Which in the distances measured up to 10 km represents a satisfactory error, although as expected the reduction of the distance also reduces the error per measurement. Also, results show several points where the error per measurement is $\leq$ 1m. Although these are good points to obtain, these points need to remain common to both gateways. As a point with minimum error in one gateway and with a greater error in other gateway, it will also induce error when multilateration takes place. By running the RANSAC analysis, the error of the linear model was reduced. Also, the worst endpoint estimations that were classified as inliers were slightly improved at the cost of other endpoints. This confirms the sensitivity inherent to the model. However, it also suggests that solutions with more number of samples per location or an average of the values result in reducing the error.

RQ3 - What's the overall level of accuracy in position estimation achieved?

As stated before, higher degrees of multilateration will, usually, result in better location estimation. However, in this case, bilateration would be the only case considered. In the test presented 4 gateways were placed inland and 2 gateways failed at providing results. However, there are other two reasons to support only the usage of bilateration which were verified in previous tests, and these are: 1) the coastal gateways are aligned in an almost straight line with a small curvature, which caused the multilateration equations to near a singularity where it is highly unstable and tends to give false results inland, and 2) the very unstable RSSI values do not provide coherent values throughout the time. This means that it is extremely difficult to fingerprint a determined distance with a RSSI value. By scaling the problem to more gateways, with this geometry, translates in more radius from each node resulting in more oscillating results. Problem 1) can be solved by better redistributing the gateways to form a better geometry, one that would contain the node. As for problem 2), better hardware and better testing conditions will result in better-obtained values. Using the post-processed RSSI values, a range was determined and bilateration was executed using the well-known location of the gateways. Afterward, the results were compared to the GPS derived values, to understand the accuracy and quality of the results obtained. By fixing problems 1) and 2) in the future, it would enable a higher degree multilateration with better fingerprinting of RSSI per location.

Another obstacle faced in this study was the limited access to boat trips for testing, usually with no prior knowledge of the trip planned, which depicts a more realistic situation. However, the planning of the gateways on land was made by guessing which way was the trip going to be. This resulted in several points being estimated with RSSI values that were distorted because the Fresnel zone of the transmission was blocked by the mast of the boat, which in the end, induced error. However, by using bilateration the estimated points were separated into classes. The best class obtained yielded results with a mean error of 650m up to the worst class which yielded results in the order of the 4 km. Being the results estimated using RSSI values, all the improvements and drawbacks are shared. Although in absolute values, these errors may seem large and prohibitive in an urban or semi-urban environment, an average error up to 5 km is still adequate for study migration patterns, and the general location of species. Thus, this location estimation proves to be useful in an oceanic scenario. Improvements could be done by further manipulating the data, removing more outliers from the dataset, resulting in further reduction of the error. Although this would not be realistic in a real scenario, because no prior knowledge concerning the point being an in- or outlier would be available. Improvements can be done to the location estimation consequently improving position estimation by the usage of the IMU included in the Pysense board. By raising the frequency of data and reducing deep sleep time, it is possible to make use of algorithms such as the Extended Kalman Filter (EKF) and implement a dead-reckoning[1] algorithm which might reduce error given a starting point. However, the latter was not tried as reducing sleep time and raising the frequency of transmission would result in rapid power drainage, resulting in results favoring one of the requirements of the project, while leaving the rest of the requirements aside.

Sensed Data

For last, figure 4.8, presents an extract of the data sensed by the Pysense onboard of the ship. These correspond to the sent payloads to the gateways. This figure also shows the distinction in temperature, pressure, and humidity found along the way. These values could be used in future applications by employing sensor fusion algorithms as a way to predict meteorological phenomena. Furthermore, the light levels, measured in lux, can be further processed at the application level to make way for implementing methodologies based on GLS.

[1] dead reckoning is the process of calculating one's current position by using a previously determined position

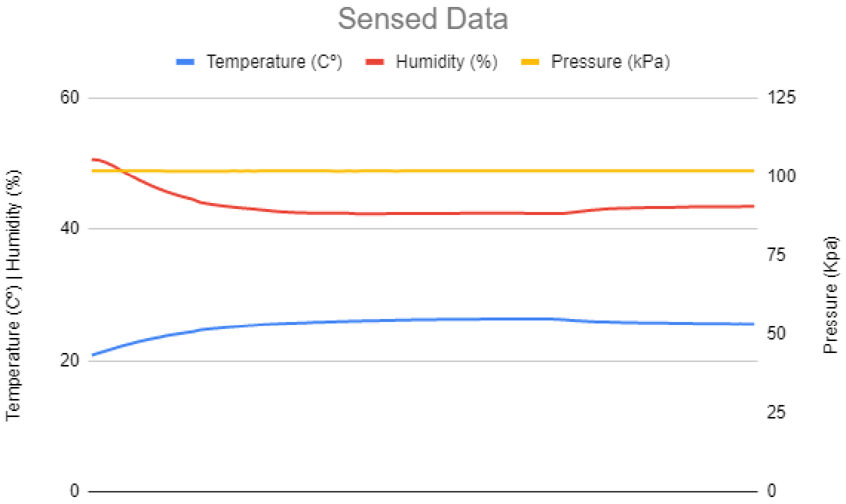


Figure 4.8: Sensed data extracted from the experimental values obtained.

4.3.2 Research Contributions

The main focus of this study was to explore the software to be used by a bio-telemetry tag that would be low-cost, and power-efficient and would provide an efficient geolocation solution. Moreover, it is intended to be used by marine seabirds in oceanic environments, namely to aid in the research and conservation of marine life. The LoRa technology and LoRaWAN protocol were used and its behavior in oceanic settings was explored by resorting to RSSI values that come at no cost in any kind of electromagnetic-based communication technology. In this way, a sensor that provides telemetry, as well as ground-truth measures, has been delivered with satisfactory longevity to be used in experiments in the field. On top of this fact, a modeled bilateration and RSSI based geolocation was developed. Even though it compromises the accuracy of the modern satellite-based technologies, like ARGOS and GPS, it comes at virtually no cost when comparing to the prices, of implementing a tag that makes use of satellite technology. The error related to the geolocation models elaborated is largely due to the fact that no control of the experiment was achieved, oscillating between LoS and non-LoS conditions which largely increased the error. Although one can not characterize a bird's flight as something controlled, this fact contributed to depict the experiment as realistic as possible, as a bird while flying would also be prone to encounter obstacles. Apart from these facts, the results obtained range in average from 660 m to 4 km. This error is adequate for study migration, patterns, general location of animals among other situations where a pin-point location is not needed. These studies not only benefit from the lifespan of the tags gained partially by employing this kind of passive geolocation, but also by removing the need for continued human interaction with the taxa.

4.3.3 Possible Applications

In the context of this study, for the position estimation system to function effectively in any given scenario, better geometry should be achieved as referred above. This would enable the sensor to reach several gateways and to apply higher degrees of multilateration, therefore reducing error. Otherwise, the system fails at providing solid and consistent results. With this fact in mind, these types of systems could be applied, in the wildlife, but to species with a lower degree of movements such as turtles, or reptiles. This would allow, to sample more RSSI per location, which can improve the error obtained, and therefore, obtaining better results. Aside from the ecology field, the most promising applications would be in areas such as cities, where one can make use of the increasing population and their increasing usage of ubiquitous systems to take a role as gateways providing multiple endpoints for a system like these. In rural environments, this system would be feasible in managing crops, where spread sensors would make use of the LoRa technology to send status messages of the monitored crops to a central server. Suitable applications most likely include cases where limitations of power and cost overrule the need for a pin-point location accuracy. Many forms of asset tracking should also be explored and present great opportunities for these systems, like locating cargo containers or managing warehouses and inventory may not require GPS accuracy. In the case, that GPS is used and high accuracy is needed, some form of radio technology is needed for communication. LoRa presents itself as a very feasible way of transmitting data, without the need of recurring to expensive hardware or the need to deal with complex software problems.

5

Conclusion

Contents

In this chapter, an overview of the performed work is provided during research, by describing the lessons learned to result from the study. Afterward, the conclusion follows this study with the future work section.

5.1 Conclusions

The challenge of this research was to use an off-shelf low-cost MCU and lightweight protocol as a way to study marine seabirds in oceanic environments. This is to obtain a way to produce telemetry of the surrounding environment. As marine seabirds access environments that are difficult to access, the possibility of mounting a system of this type would come as a great advantage. The goal was to achieve and analyze the better lifespan of such a system, as replacing batteries after deployment would be intrusive. Together with evaluating the lifespan of the system, passive geolocation has been developed to obtain the general system position. Moreover, a GPS module was also used as a baseline and a way to correct the perception of the system position, receiving a ground truth correction between telemetry readings. Besides, when in out of range for transmission conditions, the GPS module allowed that the system does not perform blindly in terms of geographical position. To accomplish this, a sensor was developed resourcing to the LoRaWAN protocol, and a network server (TTN). These provided a practical and easy implementation of the system, as well as to rapidly test the system both in-vitro and in-situ for battery lifespan enhancement. Battery autonomy was made by creating multiple states as a way of adapting the sensor to sleep and maintain an average low current supplying the sensor. Tests for the passive geolocation system were developed in-situ, on board of a touristic sea vessel at the south coast of Funchal, Madeira island. From the data collected, data were modeled and removed the noise of the model without decreasing the realistic side of the scenario that it represents. Position estimation was performed using multilateration, as there are multiple methods to estimate distance, separating emitters and receptors, and multiple methods to determine the final calculated position. From the first contact with research in the area, it becomes clear that, despite the several choice ranges in the market for hardware, methods, and methodologies, it is required to select the best technologies that fit the problem. It also became clear that the area of ecology would largely benefit from a system of this type, as technologies used, remain at a high cost and do not provide flexibility to the experts working in the area. Also, the systems used are costly partially because high accuracy is employed when not needed, although when there is no viable alternative these end up conditioning experiments. Systems like the one developed in this study will surely contribute to the increase of experiments in the ecology field, open horizons in terms of financial investment, and serve as a way of producing knowledge about the surrounding environment, paving the way for more sustainable measures, and understanding the human footprint and climatic phenomena worldwide.

5.2 System Limitations

In terms of limitations, and being the RSSI an indication of the power of the received radio-frequency signal is susceptible, as mentioned, to obstacles, reflections, multipath and other types of disturbances. These all create a problem when relying on these values to derive distances, as depicted in the plotted charts. Furthermore, in the open sea, it was extremely difficult to condition the antenna position to be in LoS conditions, which serves the purpose of attempting to depict a realistic scenario. Although these conditions result in a great distortion and oscillation of the values, therefore inducing an error in the final results. Another thing to note is that RSSI values are integer values with an inherently very limited range of values (usually 0 to -128) on a logarithmic scale. This greatly limits the resolution of the data as small differences at low distances represent several oscillations between RSSI values and small differences at higher distances represent only small oscillations, thus becoming difficult to fingerprint RSSI values to higher distances. The latter problem was mitigated by averaging the RSSI per location. During this practical experiment, 2 of the land gateways failed to deliver results. However with the very limited access to boats, and constantly being hindered of testing due to conditions at sea, the test proceeded with only 2 gateways. The 2 extra gateways might have increased the overall quality of the study and reduced error in the final position estimation, by duplicating the amount of data to tune the developed model. The limited access to boats largely affects the attempt to track species over the ocean, by forcing the study to be conducted with only land gateways and just one boat. By improving the overall geometry, the system will, for sure, increase the overall quality of the signal and decrease the final estimated point error.

5.3 Future Work

A study of this type, namely in ocean settings with all the constraints, that this environment offers to the application of IoT, opens possibilities for numerous future use cases. Moreover, the reported techniques, using calculated models remain to be further verified using real trajectories of fishing boats, where the geometry of the network could be further improved by implementing gateways on buoys and ships along the shore. In addition, the implementation of an IoT dashboard providing real-time information will also make possible to provide additional back-end processing of the data obtained. In terms of battery autonomy, and accounting with the satisfactory result obtained, can easily be transformed into solar-powered, since Pycom expansion boards all include a battery charger circuits, thus further extending the longevity of the deployment. Furthermore, flashing Pycom can be made in C language, which can contribute even more to the lifespan of the device. Another component for future deployment is to invest in proper casing and miniaturization of the sensor, although these topics were not addressed throughout this study, it remains an important aspect as, ocean environments are harsh to deal with due to salt corrosion and, of course, the presence of water thus water and pressure proofing remains a concern. Also, to cope with the immensity of the ocean a mesh network could be built based on the idea of data mules, that transmits the messages to other nodes until they reach a gateway with an internet connection. These middlemen could have fixed positions, be carried by sea vessels or, in context, be a bird flying receiving messages and working as a monitor of tagged vessels, cetaceans or other marine species. Finally, the latter would create an opportunity to further scaling up the proposed systems at an affordable price enabling other sensors to operate far away from shore collecting levels of parameters such as: salinity, levels of dissolved oxygen, ocean depth among others, which are crucial to the conservation of the ecosystem, where a real-time monitoring system brings huge benefits to the marine research and conservation and the general community.

Bibliography

[1] J. Cota-Ruiz, J. G. Rosiles, E. Sifuentes, and P. Rivas-Perea, “A low-complexity geometric bilateration method for localization in wireless sensor networks and its comparison with least-squares methods,” *Sensors*, vol. 12, no. 1, pp. 839–862, 2012.

[2] A. Fakhreddine, D. Giustiniano, and V. Lenders, “Evaluation of self-positioning algorithms for time-of-flight based localization,” *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 2016*, 2016.

[3] F. Wu, J.-M. Redoute, and M. R. Yuce, “WE-Safe: A Self-Powered Wearable IoT Sensor Network for Safety Applications Based on LoRa,” *IEEE Access*, vol. 6, pp. 40 846–40 853, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8419707/

[4] N. Sakthipriya, “An effective method for crop monitoring using wireless sensor network,” *Middle - East Journal of Scientific Research*, vol. 20, no. 9, pp. 1127–1132, 2014.

[5] R. O. Charlie Karlsson, Gunther Maier, Michaela Trippl, Iulia Siedschlag and G. Murphy, *ICT and Regional Economic Dynamics: A Literature Review*, 2010.

[6] L. Rothacker, A. Dosseto, A. Francke, A. R. Chivas, N. Vigier, A. M. Kotarba-Morley, and D. Menozzi, “Impact of climate change and human activity on soil landscapes over the past 12,300 years,” *Scientific Reports*, vol. 8, no. 1, pp. 1–7, 2018. [Online]. Available: http://dx.doi.org/10.1038/s41598-017-18603-4

[7] M. L. Tasker and R. W. Furness, “Seabirds as Monitors of the Marine Environment. ICES Cooperative Research Report No. 258,” no. 258, p. 733, 2003.

[8] L. Roman, B. D. Hardesty, M. A. Hindell, and C. Wilcox, “A quantitative analysis linking seabird mortality and marine debris ingestion,” *Scientific Reports*, vol. 9, no. 1, pp. 1–7, 2019. [Online]. Available: http://dx.doi.org/10.1038/s41598-018-36585-9

[9] F. Delaine, B. Lebental, and H. Rivano, “In Situ Calibration Algorithms for Environmental Sensor Networks: A Review,” *IEEE Sensors Journal*, vol. 19, no. 15, pp. 5968–5978, 2019.

[10] G. Fehlmann and A. J. King, "Bio-logging," *Current Biology*, vol. 26, no. 18, pp. R830–R831, 2016.

[11] S. J. Cooke, S. Hich, M. Lucas, and M. Lutcavage, "Chapter 18 - Biotelemetry and Biologging," *Fisheries Techniques*, no. January, pp. 819–860, 2012.

[12] J. M. Pearce, "Philopatry: a return to origins," *The Auk*, vol. 124, no. 3, pp. 1085–1087, 2007.

[13] B. Y. Wayne and D. Jeff, "VoL Bird Marki • g Techniques," no. 357.

[14] F. Roquet, C. Wunsch, G. Forget, P. Heimbach, C. Guinet, G. Reverdin, J. B. Charrassin, F. Bailleul, D. P. Costa, L. A. Huckstadt, K. T. Goetz, K. M. Kovacs, C. Lydersen, M. Biuw, O. A. Nøst, H. Bornemann, J. Ploetz, M. N. Bester, T. McIntyre, M. C. Muelbert, M. A. Hindell, C. R. McMahon, G. Williams, R. Harcourt, I. C. Field, L. Chafik, K. W. Nicholls, L. Boehme, and M. A. Fedak, "Estimates of the Southern Ocean general circulation improved by animal-borne instruments," *Geophysical Research Letters*, vol. 40, no. 23, pp. 6176–6180, 2013.

[15] G. N. Tuck, T. Polacheck, J. P. Croxall, H. Weimerskirch, P. A. Prince, and S. Wotherspoon, "The potential of archival tags to provide long-term movement and behaviour data for seabirds: First results from Wandering Albatross Diomedea exulans of South Georgia and the Crozet Islands," *Emu*, vol. 99, no. 1, pp. 60–68, 1999.

[16] R. A. Phillips, J. R. D. Silk, J. P. Croxall, V. Afanasyev, and D. R. Briggs, "Accuracy of geolocation estimates for flying seabirds," *Marine Ecology Progress Series*, vol. 266, pp. 265–272, 2004.

[17] J. A. D. Fisher, D. Robert, A. Le Bris, and T. Loher, "Pop-up satellite archival tag (PSAT) temporal data resolution affects interpretations of spawning behaviour of a commercially important teleost," *Animal Biotelemetry*, vol. 5, no. 1, pp. 1–10, 2017.

[18] S. G. Wilson, M. E. Lutcavage, R. W. Brill, M. P. Genovese, A. B. Cooper, and A. W. Everly, "Movements of bluefin tuna (Thunnus thynnus) in the northwestern Atlantic Ocean recorded by pop-up satellite archival tags," *Marine Biology*, vol. 146, no. 2, pp. 409–423, 2005.

[19] D. W. Kerstetter, B. E. Luckhurst, E. D. Prince, and J. E. Graves, "Use of pop-up satellite archival tags to demonstrate survival of blue marlin (Makaira," *Fishery Bulletin*, vol. 948, no. July, pp. 939–948, 2003.

[20] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. C. Markham, C. Mascolo, B. Pásztor, N. Trigoni, and R. Wohlers, "Poster Abstract : Wildlife and Environmental Monitoring using RFID and WSN Technology," *Distributed Computing*, pp. 7–8, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1644038.1644107

[21] B. Thomas, J. D. Holland, and E. O. Minot, “Wildlife tracking technology options and cost considerations,” *Wildlife Research*, vol. 38, no. 8, pp. 653–663, 2011.

[22] L. D. Mech and S. M. Barber, “a Critique of Wildlife Radio-Tracking and Its Use in National Parks a Report To the U.S. National Park Service,” no. January 2002, pp. 1–83, 2002.

[23] L. D. Mech, *Handbook of Animal Radio-Tracking*, ned - new edition ed. University of Minnesota Press, 1983. [Online]. Available: http://www.jstor.org/stable/10.5749/j.cttts51f

[24] K. L. Heezen and J. R. Tester, “Evaluation of radio-tracking by triangulation with special reference to deer movements,” *The Journal of Wildlife Management*, vol. 31, no. 1, pp. 124–141, 1967. [Online]. Available: http://www.jstor.org/stable/3798367

[25] G. M. Skupien, K. M. Andrews, and T. M. Norton, “Benefits and biases of VHF and GPS telemetry: A case study of American alligator spatial ecology,” *Wildlife Society Bulletin*, vol. 40, no. 4, pp. 772–780, 2016.

[26] D. G. Barron, J. D. Brawn, and P. J. Weatherhead, “Meta-analysis of transmitter effects on avian behaviour and ecology,” *Methods in Ecology and Evolution*, vol. 1, no. 2, pp. 180–187, 2010. [Online]. Available: http://doi.wiley.com/10.1111/j.2041-210X.2010.00013.x

[27] B. Naef-Daenzer, “Miniaturization (0.2 g) and evaluation of attachment techniques of telemetry transmitters,” *Journal of Experimental Biology*, vol. 208, no. 21, pp. 4063–4068, 2005. [Online]. Available: http://jeb.biologists.org/cgi/doi/10.1242/jeb.01870

[28] B. K. Walkerl, G. Elliottl, D. Nichollsz, D. Murray, and P. Dilksl, “ALBATROSS (Diomedea exulans) FROM THE AUCKLAND ISLANDS : PRELIMINARY RESULTS,” no. Gales, 1993.

[29] I. Argos and T. Argos, “Give Vision,” 1970. [Online]. Available: http://www.give-vision.com/

[30] D. G. Nicholls, “Satellite tracking of large seabirds - a practical guide,” *Conver*, no. August, p. 22, 1994.

[31] F. Cagnacci, L. Boitani, R. A. Powell, and M. S. Boyce, “Animal ecology meets GPS-based radiotelemetry: A perfect storm of opportunities and challenges,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1550, pp. 2157–2162, 2010.

[32] M. Hebblewhite and D. T. Haydon, “Distinguishing technology from biology: A critical review of the use of GPS telemetry data in ecology,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1550, pp. 2303–2312, 2010.

[33] J. Danebjer, “A Hybrid Approach to GPS-Free Geolocation over LoRa,” pp. 1–62, 2018.

[34] D. Bissett, “Analysing TDoA Localisation in LoRa Networks.” [Online]. Available: uuid:bea423b1-6f04-4708-8ed4-e8663dd51cde

[35] H. Kwasme and S. Ekin, “RSSI-Based Localization using LoRaWAN Technology,” *IEEE Access*, vol. PP, pp. 1–1, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8764340/

[36] “Outdoor Localization System Using RSSI Measurement of Wireless Sensor Network,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 2, no. 2, pp. 1–6, 2013.

[37] Ł. Chruszczyk and A. Zając, “Comparison of Indoor/Outdoor, RSSI-Based Positioning Using 433, 868 or 2400 MHz ISM Bands,” *International Journal of Electronics and Telecommunications*, vol. 62, no. 4, pp. 395–399, 2016.

[38] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenberger, “A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles,” *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2019.

[39] Cisco, “Antenna Pattern and Meaning,” pp. 1–17.

[40] C. Sommer, S. Joerer, and F. Dressler, “On the applicability of Two-Ray path loss models for vehicular network simulation,” *IEEE Vehicular Networking Conference, VNC*, pp. 64–69, 2012.

[41] T. Rappaport, *Wireless communications: Principles and practice*. Prentice Hall, 1996.

[42] I. Mohamed, “Path-Loss Estimation for Wireless Cellular Networks Using Okumura/Hata Model,” *Science Journal of Circuits, Systems and Signal Processing*, vol. 7, no. 1, p. 20, 2018.

[43] LoRa Alliance, “A technical overview of LoRa ® and LoRaWAN ™ LoRaWAN ™ What is it?” no. November, 2015. [Online]. Available: http://www.semtech.com/wireless-rf/iot/LoRaWAN101{_}final.pdf

[44] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment,” *ICT Express*, 2018. [Online]. Available: https://doi.org/10.1016/j.icte.2017.12.005

[45] N. Tsavalos and A. Abu Hashem, “Low Power Wide Area Network (LPWAN) Technologies for Industrial IoT Applications,” 2018. [Online]. Available: https://lup.lub.lu.se/student-papers/search/publication/8950859

[46] R. S. Sinha, Y. Wei, and S. H. Hwang, “A survey on LPWA technology: LoRa and NB-IoT,” *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.icte.2017.03.004

[47] P. D. I. Milano, “Polo Regionale di Como Final Thesis for MSc . Computer Engineering RSSI Based Localization for Indoor Application by using 868 MHz Radio Signal,” no. 737700, 2012.

[48] K. G. Derpanis, “Overview of the RANSAC Algorithm,” vol. 4, pp. 1–2, 2005. [Online]. Available: http://www.cse.yorku.ca/{~}kosta/CompVis{_}Notes/ransac.pdf

Appendix A

Listing A.1: Example of a message in JSON format

```
{
  ”app_id”: ”my-app-id”,
  ”dev_id”: ”my-dev-id”,
  ”hardware_serial”: ”0102030405060708”,
  ”port”: 1,
  ”counter”: 2,
  ”is_retry”: false,
  ”confirmed”: false,
  ”payload_raw”: ”AQIDBA==”,
  ”payload_fields”: {
    ”accelerometer_X”:”0.34564”,
    ”accelerometer_Y”:”-0.6950505”,
    ”accelerometer_Z”:”9.8998786”,
    ”pitch”:”-160.779”,
    ”roll”:”123”,
    ”humidity”:”56”,
    ”temperature”:”31”,
    ”light”:”1234”,
    ”light1”:”123123”,
    ”pressure”:”101”,
    ”battery”:”4.6”
    },
  ”metadata”: {
    ”time”: ”1970-01-01T00:00:00Z”,
    ”frequency”: 868.1,
    ”modulation”: ”LORA”,
    ”data_rate”: ”SF7BW125”,
    ”bit_rate”: 50000,
    ”coding_rate”: ”4/5”,
    ”gateways”: [
      {
        ”gtw_id”: ”ttn-herengracht-ams”,
        ”timestamp”: 12345,
        ”time”: ”1970-01-01T00:00:00Z”,
        ”channel”: 0,
        ”rssi”: -25,
        ”snr”: 5,
        ”rf_chain”: 0,
        ”latitude”: 52.1234,
        ”longitude”: 6.1234,
        ”altitude”: 6
      },
{
        ”gtw_id”: ”exemplo2”,
        ”timestamp”: 12345,
        ”time”: ”1970-01-01T00:00:00Z”,
        ”channel”: 0,
        ”rssi”: -25,
        ”snr”: 5,
        ”rf_chain”: 0,
        ”latitude”: 52.1234,
        ”longitude”: 6.1234,
        ”altitude”: 6
      }
    ],
    ”latitude”: 52.2345,
    ”longitude”: 6.2345,
    ”altitude”: 2
  },
  ”downlink_url”: ”https://integrations.thethingsnetwork.org/ttn-eu/api/v2/down/my-app-id/my-process-id?key=ttn-account-v2.secret”
}
```

Listing A.2: `Payload decoder in Javascript`

```
function Decoder(bytes, port) {
  var values=[2,2,2,4,4,2,1];
  var values1=[2,4,4,4,1,1,1,2,2,4,4,2,1];
  var decoded=[];
  var a=0;
  var b=0;
  var c=bytes.length;

  if (c==17){
    for (var i = 0; i < values.length; ++i) {
    var aux=[];
    for (var x = 0; x < values[i]; ++x) {
      aux.push(bytes[a]);
      a++;
    }
    if (values[i]==4){
        decoded.push(bytesToFloat(aux));
    } else {
        aux.reverse();
        decoded.push(intFromBytes(aux));
    }
  }
  } else {
    for (var q = 0; q < values1.length; ++q) {
    var aux1=[];
    for (var z = 0; z < values[q]; ++z) {
      aux1.push(bytes[b]);
      b++;
    }
    if (values[q]==4){
        decoded.push(bytesToFloat(aux1));
    } else {
        aux1.reverse();
        decoded.push(intFromBytes(aux1));
    }
  }
  }

  var id =decoded[0];
  var humidity = decoded[1];
  var temperature = decoded[2];
  var light = decoded[3];
  var light1 = decoded[4];
  var pressure = decoded[5];
  var battery = decoded[6];
  var latitude = decoded[7];
  var longitude = decoded[8];
  var altitude = decoded[9];
  var hour = decoded[10];
  var minute = decoded[11];
  var second = decoded[12];

  return{
  id:id,
  humidity:(humidity/10),
  temperature:((temperature/10)-273.15),
  light:light,
  light1:light1,
  pressure:(pressure/10),
  battery:(battery/10),
  latitude:latitude,
  longitude:longitude,
  altitude:altitude,
  hour:hour,
  minute:minute,
  second:second
  };
}

function bytesToFloat(bytes) {
    // JavaScript bitwise operators yield a 32 bits integer, not a float.
    // Assume LSB (least significant byte first).
    var bits = bytes[3]<<24 | bytes[2]<<16 | bytes[1]<<8 | bytes[0];
    var sign = (bits>>>31 === 0) ? 1.0 : -1.0;
    var e = bits>>>23 & 0xff;
    var m = (e === 0) ? (bits & 0x7fffff)<<1 : (bits & 0x7fffff) | 0x800000;
    var f = sign * m * Math.pow(2, e - 150);
    return f;
}

function intFromBytes( x ){
    var val = 0;
    for (var i = 0; i < x.length; ++i) {
        val += x[i];
        if (i < x.length-1) {
            val = val << 8;
        }
    }
    return val;
}
```

B

Project Code

Listing B.1: Node, boot.py

```python
import pycom
import machine
from machine import Pin

pycom.heartbeat(False)

from network import WLAN
from network import Bluetooth
import os

pycom.wifi_on_boot(False)
wlan = WLAN()                           # Instantiates WLAN
wlan.deinit()                           # Turns off wifi
bluetooth = Bluetooth()                 # Instantiates Bluetooth
bluetooth.deinit()                      # Turns off Bluetooth

counter=pycom.nvs_get('count')          # Get ID value from memory

```

```
if type(counter)!= int:                    # If the value is not in memory
    pycom.nvs_set('count', 0)              # ID = 0
```

Listing B.2: Node, main.py

```
import machine
import socket
import ubinascii
import struct
import os
import time
import config
import pycom
import gc
from machine import Pin, SD, WDT, UART, RTC
from network import LoRa
from micropyGPS import MicropyGPS
from util import *
from pysense import Pysense
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2,PRESSURE,ALTITUDE
from struct import *

global sd
global SD_ON_BOOT
global SD_READ
global SEMAFORO
global STRING_READ
global GPS_READ

LOG_FILENAME = 'node.log'
SEMAFORO = True
SD_READ = False
SD_ON_BOOT = False
sd = None
GPS_READ=False
WIFI_SSID = 'M-ITI'
WIFI_PASS = 'M1T1-W1F1'
py = Pysense()
mpp = MPL3115A2(py,mode=PRESSURE)    # Returns PRESSURE in Pascals
si = SI7006A20(py)                   # temp, humidity
lt = LTR329ALS01(py)                 # light
#li = LIS2HH12(py)                    # accelerometer Para apagar
gps_parser = MicropyGPS()            # GPS lib
GPS_TRIES = 400                       # GPS tries to connect
wdt = WDT(timeout=90000)
GPS_pin = Pin('P2', mode=Pin.OUT, pull=Pin.PULL_DOWN)
GPS_pin.value(0)
gc.enable()                          # enable garbage collection
```

```python

# Input: void
# Output: ---------
# Mounts SD card
def mount_SD():
    global sd
    global SD_ON_BOOT
    print('Mount SD')
    sd = None
    try:
        sd = SD()
        if SD_ON_BOOT == False:
            os.mount(sd, '/sd')
            SD_ON_BOOT = True
    except Exception as e:
        print('No SDcard to mount: ' + str(e))

# Input: void
# Output: -------------
# Unmounts SD card
def unmount_SD():
    global sd
    sd = None
    try:
        os.unmount('/sd')
    except Exception as e:
        print('No sd card to unmount: ' + str(e))

# Input: void
# Output: List that contains read values in fixed positions
# Sensor reading from in a determined instant t
def readSensor():
    global GPS_READ
    list=[]

    try:
        # acc_x = li.acceleration()[0]
        # acc_y = li.acceleration()[1]
        # acc_z = li.acceleration()[2]
        # pitch = li.pitch()
        # roll = li.roll()

        id=pycom.nvs_get('count')
        humid = int(round(si.humidity()*10))
        temp = int(round((si.temperature()+273.15)*10))
        light = lt.light()[0]
        light1 = lt.light()[1]
        press=int(mpp.pressure()/100)
        batt = int(py.read_battery_voltage()*10)

        if GPS_READ == True:

            gps_counter = 0
            gps_uart = UART(1, baudrate=9600, pins=('G7', 'G6'))

            for gps_counter in range(GPS_TRIES):
                    print('updating gps' + '.'*(gps_counter + 1))
                    time.sleep(0.1)
                    update_gps(gps_uart)
                    if gps_parser.latitude[0] != 0:
                        break

            latitude=coord_deg_to_dec(gps_parser.latitude)
            longitude=coord_deg_to_dec(gps_parser.longitude)
            altitude=gps_parser.altitude
            hour=gps_parser.timestamp[0]
            minute=gps_parser.timestamp[1]
            second=int(gps_parser.timestamp[2])

            list.append(id)             # i = 4 bytes
            list.append(humid)          # H = 2 bytes
            list.append(temp)           # H = 2 bytes
            list.append(light)          # f = 4 bytes
            list.append(light1)         # f = 4 bytes
            list.append(press)          # H = 2 bytes
            list.append(batt)           # B = 1 byte
            list.append(latitude)       # f = 4 bytes
            list.append(longitude)      # f = 4 bytes
            list.append(altitude)       # f = 4 bytes
            list.append(hour)           # B = 1 byte
            list.append(minute)         # B = 1 byte
            list.append(second)         # B = 1 byte
            return list                 # Total = 34 bytes

        else:
            list.append(id)             # i = 4 bytes
            list.append(humid)          # H = 2 bytes
            list.append(temp)           # H = 2 bytes
            list.append(light)          # f = 4 bytes
            list.append(light1)         # f = 4 bytes
            list.append(press)          # H = 2 bytes
            list.append(batt)           # B = 1 byte
            return list                 # Total = 19 bytes
    except Exception as e:
        print(str(e))

# Input: values from sensors, formats
# Output: returns bytes
# Encodes bytes with struct pack
def encoder(v, f):
    bytes=bytearray()
    values=v
    formats=f

    for i in range(len(values)):
       bytes.extend(struct.pack(formats[i], values[i]))

    return bytes

# Input: bytearray
# Output: List of read values
# Decodes the bytearray into the original read values
def decoder(bytearray):
    global GPS_READ
    formatSize = {
    'b': 1,
    'B': 1,
    'H': 2,
    'h': 2,
    'f': 4,
    'I': 4,
    'i': 4,
    }

    if GPS_READ == True:
        headers = ['id','lat','long','alt','hour','min','sec','humid','temp','light','light1','press','batt']
        NR_VALS = 13
    else:
        headers = ['id','humid','temp','light','light1','press','batt']
        NR_VALS = 7

    fin_i=0
    curr_i = 0
    decoded_val=[]

    for i in range(NR_VALS):
      fin_i = curr_i + formatSize[formats[i]]
      subPkt = bytes2send[curr_i:fin_i]
      decoded_val.append(struct.unpack(formats[i], subPkt)[0])
      curr_i = fin_i

    return  decoded_val

# Input: gps_uart
# Output: Updated gps string
# Function that updates gps
def update_gps(gps_uart):
    try:
        if gps_uart.any() > 0:
            gps_line = gps_uart.readline()
            final_str = ''.join([chr(decimal) for decimal in gps_line]).replace('\r','').replace('\n','')
            #printd(final_str)
            for char in final_str:
                gps_parser.update(char)
    except:
        print('Error reading GPS')

```

```python
# Input: filename, string
# Output: -------
# Writes string into designated string
def write_toSD(filename, string):
    try:
        if sd != None:
            str_toWrite=ubinascii.hexlify(string)
            with open('/sd/' + filename, 'a') as f:
                f.write(str_toWrite +'\n')
                f.close()
            return
    except Exception as e:
        print(str(e))
        return

# Input: filename
# Output: string read from filename
# Read first line from file in SD card
def read_fromSD(filename):
    try:
        global SD_READ
        if sd != None:
            print("Read SD")
            with open('/sd/' + filename, 'r') as f:
                str_read=f.readline()
                if str_read !='':
                    str_read=str_read.replace("\n", "")
                    str_read=str_read.encode()
                    str_read=ubinascii.unhexlify(str_read)
                    str_read=bytearray(str_read)
                    SD_READ=True
                    return str_read
                else:
                    SD_READ=False
                    str_read='No_msg_TX'
                    return str_read
    except Exception as e:
        print(str(e))
        return

# Input: filename
# Output: --------
# Deletes first line from SD card
def deline_fromSD(filename):
    try:
        if sd != None:
            f1=open('/sd/aux.log', 'w')
            with open('/sd/' + filename, 'r') as f:
                a=f.readline()
                while a != "":
                    a=f.readline()
                    f1.write(a)
                f1.close()
                f.close()
            os.remove("/sd/node.log")
            os.rename("/sd/aux.log", '/sd/' + LOG_FILENAME)
            return
    except Exception as e:
        print(str(e))
        return

# Input: LoRa instance
# Output: -------------------- (End of Cycle)
# LoRa callback responsible for managing events of transmission
def lora_cb(lora):
    global SD_READ
    global SEMAFORO
    global STRING_READ
    global GPS_READ
    events = lora.events()
    print(" -------- !!!! CALLBACK !!!! -------- ")

    if events & LoRa.TX_PACKET_EVENT:

        print("SD_READ: ",SD_READ)

        if SD_READ==False:
            str_read=read_fromSD(LOG_FILENAME)
            STRING_READ=str_read
            if STRING_READ=='No_msg_TX':
                lora.nvram_save()
                time.sleep(0.1)

                if GPS_READ==True:
                    py.setup_sleep(9479)
                    py.go_to_sleep(False)
                else:
                    py.setup_sleep(2520)
                    py.go_to_sleep(False)

            else:
                deline_fromSD(LOG_FILENAME)
                SD_READ=True
                s.send(STRING_READ)
                return

        else:
            lora.nvram_save()
            time.sleep(0.1)

            if GPS_READ==True:
                py.setup_sleep(9792)
                py.go_to_sleep(False)
            else:
                py.setup_sleep(2023)
                py.go_to_sleep(False)


    elif events & LoRa.TX_FAILED_EVENT:
        if SD_READ == False:
            write_toSD(LOG_FILENAME, bytes2send)
            lora.nvram_erase()
            time.sleep(0.1)

            if GPS_READ==True:
                py.setup_sleep(10310)
                py.go_to_sleep(False)
            else:
                py.setup_sleep(3240)
                py.go_to_sleep(False)


        else:
            write_toSD(LOG_FILENAME, STRING_READ)

            lora.nvram_erase()
            time.sleep(0.1)

            if GPS_READ==True:
                #py.setup_sleep(10)
                py.setup_sleep(12454)
                py.go_to_sleep(False)
            else:
                #py.setup_sleep(10)
                py.setup_sleep(3371)
                py.go_to_sleep(False)

lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)

lora.nvram_restore()
time.sleep(0.1)

if not lora.has_joined():
    dev_eui = ubinascii.unhexlify('70B3D54995F6E9F1')
    app_eui = ubinascii.unhexlify('70B3D57ED0019C87')
    app_key = ubinascii.unhexlify('169C2EF739EAAF4051CD16B8CDA3D56A')

    # set the 3 default channels to the same frequency (must be before sending the OTAA join request)
    lora.add_channel(0, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)
    lora.add_channel(1, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)
    lora.add_channel(2, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)

    try:
        lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key), timeout=16000, dr=config.LORA_NODE_DR)

        while not lora.has_joined():
            time.sleep(2.5)
            print('Not joined yet...')
        print('LoRa joined!')

    # remove all the non-default channels
        for i in range(3, 16):
            lora.remove_channel(i)

    # create a LoRa socket
        s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

    # set the LoRaWAN data rate
        s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, True) # send confirmed messages
    # make the socket non-blocking
        s.setblocking(False)
        lora.nvram_save()
        time.sleep(0.1)

    except Exception as e:
        print(str(e))
else:
    print("Join already")
    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, True)
    s.setblocking(False)

mount_SD()
lora.callback(trigger=(LoRa.TX_PACKET_EVENT | LoRa.TX_FAILED_EVENT), handler=lora_cb)

while (True):
    print("CICLO")
    global sd
    global SEMAFORO
    global STRING_READ
    global GPS_READ
    bytes2send = bytearray()
    values=[]
    a=pycom.nvs_get('count')
    decoded_val=[]

    if a%4==0:
        GPS_READ = True
        GPS_pin.value(1)
        formats=['H','H','H','f','f','H','B','f','f','f','B','B','B']
    else:
        GPS_READ = False
        formats = ['H','H','H','f','f','H','B']

    print("GPS_READ :", GPS_READ)

    values=readSensor()
    bytes2send=encoder(values, formats)
    GPS_pin.value(0)
    # print('bytes2send', bytes2send)
    decoded_val=decoder(bytes2send)
    print("Decoded values: ", decoded_val)

    a+=1
    pycom.nvs_set('count', a)

    if lora.has_joined():
        try:
            print("Lora joined -> 1st send")
            s.send(bytes2send)

            while (SEMAFORO == True):
                pass

        except Exception as e:
            print(str(e))

    if not lora.has_joined():
        print("Not joined -> Save message")
        write_toSD(LOG_FILENAME, bytes2send)
        time.sleep(0.1)

        if GPS_READ==True:
            #py.setup_sleep(10)
            py.setup_sleep(11032)
            py.go_to_sleep(False)
        else:
            #py.setup_sleep(10)
            py.setup_sleep(2572)
            py.go_to_sleep(False)
```