Investigating LoRa for use in a Cattle Tracking and Monitoring System

Presented by:
Mark Njoroge

Prepared for:
Dr. Simon Winberg
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfillment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering.

11 November 2020

Key words:
IoT, LoRa, Localisation, TOA, Synchronisation, Farming

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another person's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:
Mark Njoroge

Date: 11 November 2020

Acknowledgements

First and Foremost, I would like to thank **God** for the endless grace and mercy provided onto me. Without Him, I am nothing.

I would like to thank my Supervisor, **Dr Simon Winberg**, for the invaluable advice and guidance bestowed upon me during the course of this project.

Next to receive thanks is my wonderful family. My Mother **Mary** for being there for me for as long as I can remember. My father, **John** for being the epitome of strength and determination, and an all round role model in my life. Also to my brother, **Sam**, for paving the way in life and always ready with a helping hand when I need it.

To my fellow engineering peers, who know themselves, for the wonderful four years we have shared together. From the long nights and the stressful submissions to the fun times shared, I would not have wished for a better experience.

Finally, I would like to thank everyone I've met along my journey. Everyone who has had an impact in my life, big or small. Every encounter that shaped my mind, that led me to this final stage. There are far too many to count, and I am grateful to all.

Abstract

Keeping track of cattle is an important aspect of farming. There are dangers of cattle being stolen or leaving farmlands while grazing. Existing measures to reduce such happening may be expensive or prove to be ineffective. Therefore, farmers have, for many years, been interested in electronic means, to track & monitor their livestock rather than relying on sleepy guards patrolling large vulnerable areas.

In this project, the objective is to prepare a comprehensive analysis of how a LoRa based system, using single central gateway, can be designed around providing a low-cost solution to the need for tracking cattle (in particular cows), in addition to other useful monitoring data that can be gleaned from such a sensor network. This is not planned to be limited to tracking cattle, and reducing stock theft; but to add additional benefits as well –such as the acquisition of sensory data, which can be fed into data science applications, and assist in the planning and management of cattle farming.

The specific challenges involved with animal tracking with networked nodes is in the localisation of the sensor nodes. The project uses a Time of Arrival based scheme to calculate distance between two sensor nodes. This requires accurate timings between the sensor nodes such that the time of flight can be adequately calculated. In order to achieve this, a synchronization protocol is implemented on both nodes.

Contents

List of Figures

List of Tables

Nomenclature

ISR Interrupt Service Routine

LoRa Long Range

LPWAN Low Power Wide Area Network

RBS Reference Broadcast Synchornization

RSS Received Signal Strength

TDOA Time Difference of arrival

TOA Time of Arrival

TPSN Timing Synch Protocol for Sensor Networks

WSN Wireless Sensor Network

1 Introduction

In livestock farming, a major task farmers have to perform is to protect their livestock. As with anything, they are at risk of theft or even straying too far off grazing lands. For years, there have been different techniques that have been implemented to prevent the loss of cattle as this translates to a huge loss for the farmer. These techniques could involve hiring of guards and fencing off farmlands - but this could prove to be expensive and may not have the desired effect.

In the ever progressing technological world, and in the midst of this fourth industrial revolution, an electronic solution to the tracking and stock-keeping of livestock would prove to be an effective solution to the problem. This project investigates the possibility of developing an electronic tracking system that could be deployed in farms and provide a farmer with the knowledge of the location of their cattle.

1.1 Background

The task is to investigate the possibilities of a low-cost tracking system based on LoRa technology. This poses general constraints to the development that would take place. Such constraints include low cost, low power budget, low bandwidth , etc. These constraints come about as a result of the tradeoff that exists in this realm of wireless networks. Being able to utilize this technology to its best potential will be the ultimate goal.

The use of long range network links to achieve tracking has been investigated before, although minimally. There already exists ready made tracking applications both for consumers as well as targeted applications. These employ technologies such as GPS however the largest drawback observed is in their expensive nature. Other works that have explored tracking without the use of pricey technology have been constrained to indoor or low area outdoor locations. The principles employed in these are similar to that which would be explored, however the constraints experienced differ. The problems faced with sensor networks deployed over large geographical areas include increased signal interference, a limited power supply and low bandwidth.

1.2 Objectives

The ultimate objective of this research is to develop a point to point connection between two sensor nodes that can be used to transfer data over a large area, in order to track and monitor cattle. The functionality of this connection can be broken down into specific cases. These can be listed as:

1. Have a communication Scheme so as to understand each other
2. Achieve distance tracking. That is to say, the main node should be able to calculate how far away the second node is.
3. The prototype design should allow for further development which would include scalability (increasing number of nodes in the network) and adding peripherals such as sensors to each node.

1.3 Terms Of Reference

As per the objectives, this section develops on the system's user and functional requirements. These will be based on the operation of the prototype to be developed.

The Following is a list of the user requirements for the system. This is limited to what a user expects of the system, and how they might interact with it.

UR1: Identify a Specific Sensor Node
This requirement is based off the fact that cattle farming usually entails having multiple cows. Being able to identify the cow from a central location allows the user to be able to gain more information about the cow. In the event that more functionality is added to this system, such as monitoring vital signs of the cow, it would be necessary to have an ID to each cow.

UR2: Know where the sensor Node is at a particular time
This requirement stems from the tracking nature of the investigation. The user would need to be able to tell where the sensor node - and hence cow - is at any time. For this prototype, this would mean knowing the distance away from the gateway.

UR3: The System should be long lasting and cover a large area
This requirement is based on the fact that farmlands usually cover a very wide area, which means that connectivity to a main power grid would be unachievable. Additionally, the gateways are potentially kilometres away from the nodes, and this distance would need to be covered by the messages.

Each user requirement contains various aspects in order to achieve it. This leads to functional requirements that describe the way in which the system is expected to function to achieve what is necessary. The functional requirements are therefore listed as:

- **FR1: Have a communication scheme that allows the Devices to only interact with each other.**
- **FR2: Each Node Should have a unique Identifier.**
- **FR3: Each Message sent should Include the Node ID.**
- **FR4: The clocks on the Nodes should be syncrhonised.**
- **FR5: The Main node should be able to calculate how far away the second node is.**
- **FR6: Sensor Nodes should have Low Power Consumption**
- **FR7: The System should be able to work over large areas, Inclusive of Obstructions**

Each of the requirements outlined will need to be tested. A detailed explanation of the tests performed will be explained in chapter 5, however the summarised traceability matrix is shown in table I

Test	**Description**	**Functional Requirements Checked**	**User Requirements Being Tested**
T1	Testing Clock Synchronization	FR1, FR2, FR3, FR4	UR1
T2	Testing the range of the device with Different Tx Powers	FR6	UR3
T3	Testing ranging with different LoRa Configurations	FR7	UR3
T4	Testing with obstructions Involved	FR7	UR3
T5	Testing distance calculations	FR5,	UR2

Table I: System Traceability Matrix

1.4 Scope and Limitations

This project, like many BSc level research projects, contains constraints. The largest limitation is in the time allocated. The project spanned a period of 12 weeks, from mid August 2020, to mid November 2020. The entire work was to be done in this time frame, from conception to implementation and finally to completion. There also exists a monetary constraint; the system aims to be low-cost, and in addition to this, there is a maximum budget. Another constraint that is unique to this project in 2020, is the covid-19 pandemic. The pandemic has affected procurring components (limiting to south african retailers), and due to lockdown regulations, has forced the project to be done remotely with limited access to equipment and resources such as oscilloscopes and a spectrum analyser.

The detailed scope and limitations of this project are detailed as:

1. Only two radio modules were procured, thus limiting the tracking to simply distance tracking. This is because at least three gateway nodes would be required to accurately find the location.
2. Because of the previous point, there will be no need to centrally store garnered details to a central server.
3. No additional sensory data will be used. It is assumed that this data can be acquired if need be easily and appended to message data.
4. The prototype will employ a timing synchronising scheme, which will aim to synchronise the clocks of the two nodes, and hence provide accurate distance measurements.
5. In order to mitigate the constraint in accessing labs and equipment, "ready made" components and modules will be favoured for use.

In summary, the main scope of this project would be to achieve a reliable communication link between two sensor nodes, with the ability to measure distances between them.

1.5 Document Outline

This paper is divided into 7 chapters. Each chapter explores a specific section of the research undertaken. A short description of each chapter follows.

Chapter 2: Literature Review
This chapter analyses the existing works that have been done in relation to the topic at hand. It reviews the different sections that relate to this project, and how they can be applied and improved for the successful implementation of the project.

Chapter 3: Research Methodology
This chapter details the research methodology employed. It describes the different phases of the project and how they interrelate. It also shows a flow diagram of the various phases. It explains the different design and testing methodologies used.

Chapter 4: Design and Implementation
This chapter describes the design of the prototype. It contains the physical design, which entails the procurement of hardware and components. It also contains the software design, inclusive of all algorithms, compensations and considerations dealt with.

Chapter 5: Testing Procedure
This chapter describes the different tests that were performed. The various tests aim to gain

information on the performance of the developed prototype. As such, the chapter explains what each test entails and therefore the aspect that is being investigated. It provides the testing procedures used and other facts in regards to the tests.

Chapter 6: Results and Discussion
This chapter details the results of the tests undertaken. It analyses what information has been obtained from the tests, and confirms whether the prototype met the requirements or not. In addition, it provides a commentary on ways in which the prototype can be improved given the results obtained, and possible refinement of the requirements that would arrise.

Chapter 7: Conclusion
This chapter concludes the paper. It contains a summary of the initial aims of the project and the way in which they were attempted to be achieved. It gives a brief overview of the designs and results obtained. It also makes recommendations for future work given what was learnt from the project.

Chapter 8: Recommendations
This chapter gives a brief list of proposed recommendations to take the project further. It provides the details and aspects that would be required to take the work presented into a full and complete system that can be utilised fully.

2 Literature Review

This chapter analyses the different aspects regarding this project, and related works pertaining to those sections. The sections include: wireless communications, IoT used in Agriculture, localisation, Synchronisations and Networking.

2.1 Wireless Communications

There are various means of communicating wirelessly over the air between devices and each have their own merit and fallbacks. The choice of connection depends on the application it is usually intended for. This is because there exists a trade-off between the range of communications and bandwidth offered [1]. This trade-off stems from the power consumption associated with the communication technology being used. In order to transmit large data streams over a long distance, a lot of power is required. Furthermore, if you have a specific power budget, in order to increase range, you'd need to decrease the amount of data being transmitted, and conversely, to increase data, the distance range is reduced.

The different existing wireless technologies all differ in what they prioritise. For example WiFi has low range, but is able to transmit vasts amount of data. In contrast, wireless technologies that focus on range such as SigFox and LoRa are capable of transmitting far less amounts of data [1].

With the design constraints and desires of machine to machine communication, the desired characteristic of the wireless technology to use would be to consume low amounts of power and covering a wide area, which is referred to as a Low Power Wide Area Network (LPWAN) [2]. The different technologies that exist that fall under this category include NB-IoT [3] , SigFox [4] and LoRa [5]. Analysing each one, we see different advantages that would suit the design. The low power characteristics of these technologies translate to battery lives that last years, which in a sensory, Internet of Things network, is an incredibly valuable characteristic.

2.1.1 LoRa and LoRaWAN

LoRa and LoRaWAN are two distinctive terms. LoRa is a modulation scheme that is patented by Semtech [6]. It describes the physical layer of communication protocol between end devices in an RF network. The frequency it operates at depends on the region and it is either 433MHz, 868MHz or 915MHz. These frequency bands are unlicensed [7] which makes using the technology cheap to develop.

LoRa

LoRa is based on a Chirp spread spectrum modulation scheme. This means that the data is encoded in a series of chirp signals within a narrow bandwidth over the carrier signal. The bandwidth is usually either 125kHz, 200kHz or 500kHz. This bandwidth affects the modulation and demodulation rate. Another parameter that affects the data transfer in LoRa is the Spreading Factor (SF), which denotes the number of chips per symbol. [8]. The Spreading factor used is between 7-12. In simple terms, it is a measure of the amount of bits that can be encoded onto a single symbol in a piece of data. A large SF increases energy consumption and range, whereas a smaller SF increases speed of transfer.

A final parameter involved in describing the LoRa mode is the Coding Rate (CR) which is a measure of the ratio of useful data in a message frame. It can fall between 4/5 and 4/8, and also affects the energy consumption and speed of transfer [7].

In a LoRa wireless link, the sender and receiver needs to have these three parameters matched in order to achieve successful communication. One can tailor the choice of these parameters based on the particular constraints and use case of the system. The user has flexibility in how to set up the link in their network.

LoRaWAN

LoRaWAN describes the networking protocol that makes use of LoRa technology. It is a star of star topology that connects end nodes to gateways, and gateways to the internet [9]. Further analysis of networking topologies is discussed in section 2.2. Figure 2.1 shows the layer of communication when using LoRa and LoRaWAN.

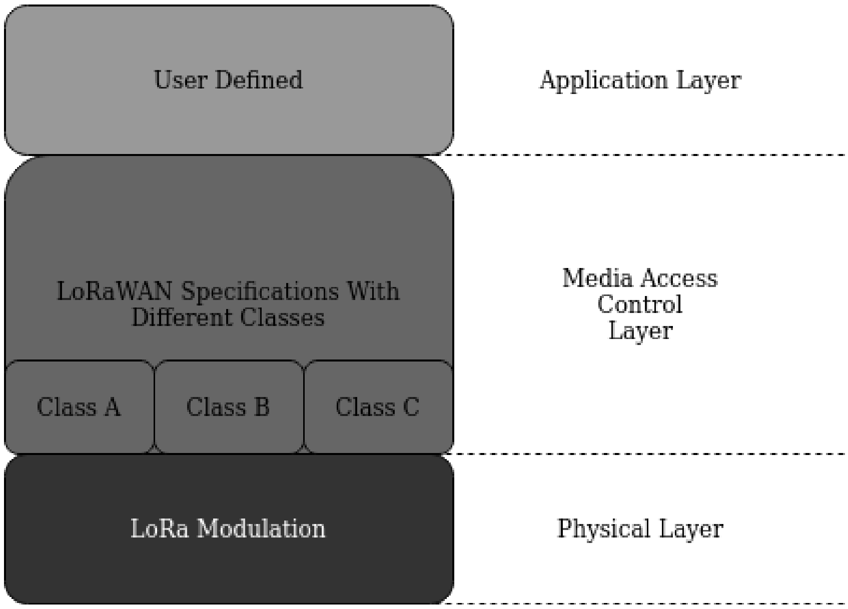

Figure 2.1: Layers of LoRaWAN

As described, the LoRaWAN gateway acts as the connector to the internet. There are various online servers to connect a gateway to, but the biggest one to date is The Things Network [10]. Users are able to set up their own gateways and connect them to the platform. An example of this is the work done by Petratiru et al [11] who designed a LoRaWAN gateway based on the Raspberry Pi 3B+.

Additionally, LoRaWAN can be broken down into 3 classes that cater to different strengths and use cases. These are Class A, B and [5]. The big difference in the classes lies in the operational schemes. Class A and B are suited for battery powered end nodes as they incorporate periods of time whereby the end nodes are awake and listening to messages and times when they are asleep. The biggest difference is that class B incorporates a synchronised receiving scheme for end nodes, based on a beacon sent by the gateway. Class C differs from the first two mainly in that the end nodes would need to be constantly kept on as receivers. This makes the class half duplex, and also very power consuming. Nodes operated in this class will require connection to a continuous power source, as a battery will not last.

2.1.2 Comparison of LoRaWAN to other LPWAN Technologies

As mentioned earlier, there are other technologies that can be used and achieve the same goal. The two most noteworthy are Sigfox and NB-IoT. This sub section analyses the two and how they compare to the chosen LoRa.

SigFox
SigFox is the oldest and biggest LPWAN IoT technology that utilises low bandwidth communication techniques [12]. It describes both the physical modulation layer as well as the networking architecture. It operates in the 868/915MHz band as with LoRa, however has a much less frequency spread of 100KHz. The SigFox structure also only focuses on communication from end-node to central gateway, and so greatly reduces the capabilities of two-way communication [13].

SigFox radios come cheaper than LoRa radios, however the ability to use SigFox is governed by network operators, and so it would cost money to make use of SigFox. [7]. This makes it less open to users, and concurrently removes the possibility of having a private network.

NB-IoT
NB-IoT stands for Narrowband IoT. This is a cellular technology that, like SigFox and LoRa, allows for wide area connections. It uses a different modulation scheme that takes up less time and therefore allows for faster and larger amounts of data. However, this comes at a cost of power consumption, which will largely decrease the battery life of end-nodes. A main disadvantage that this scheme poses over LoRa is that, unlike LoRa, it operates on licensed frequency bands [14] and is therefore much more expensive.

It is also not suited for rural land masses that have limited cell tower connectivity [14]. As the main purpose for this project is to focus on these two aspects, using NB-IoT would be counter intuitive, regardless of the added benefits.

2.2 Topologies in Sensor Networks

The arrangement of different nodes in a network is referred to as a network topology. The architectural design of a network can vary in multiple nuanced ways, however all are based on the same basic topologies that exist.

The most common are:

1. Bus Topology
2. Ring Topology
3. Star Topology
4. Mesh Topology

These four topologies form the building blocks of greater and more complex networking designs. In wireless sensor networks, the last two are most prevalent. This is because they are the most flexible to implement, and offer highly scalable characteristics. This section gives a brief overview of them and looks at the main advantages and disadvantages posed.

2.2.1 Bus Topology

This is the simplest topology, it exists mainly in wired networks and is rarely seen in wireless technologies. Each node in the network is connected to a main line that acts as a bus. In order to send data, the device sends into the bus and every other device in the network has access to it. Each device has a specific address and that is how messages reach their intended targets [15]. Figure 2.2 shows a simplified diagram of a bus network:

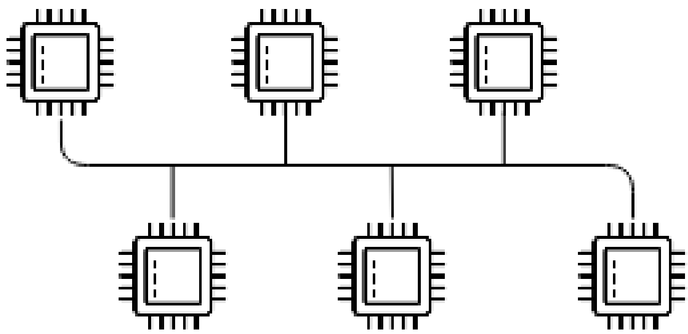

Figure 2.2: Bus Topology

In wired networks, this topology is the simplest to implement. In wireless networks it would not be as simple because of the nature of propagation of signals. This topology is not ideal for constant communication because the data bus has a chance of reaching its limit, at which point end nodes would need to wait in order to send signals.

2.2.2 Ring Topology

In this topology, devices are only connected to two other devices, which are known as point to point links [16] This connection runs from the first device and goes on until the final device connects back to the first device. Messages are forwarded between devices until they reach their target destination. In this way, a circular pattern is formed with no clear defined start or end. In wirelessl networking, this topology is again rarely used due to its impracticality and fragility. Once one link breaks, the appeal of the topology ceases to exist.

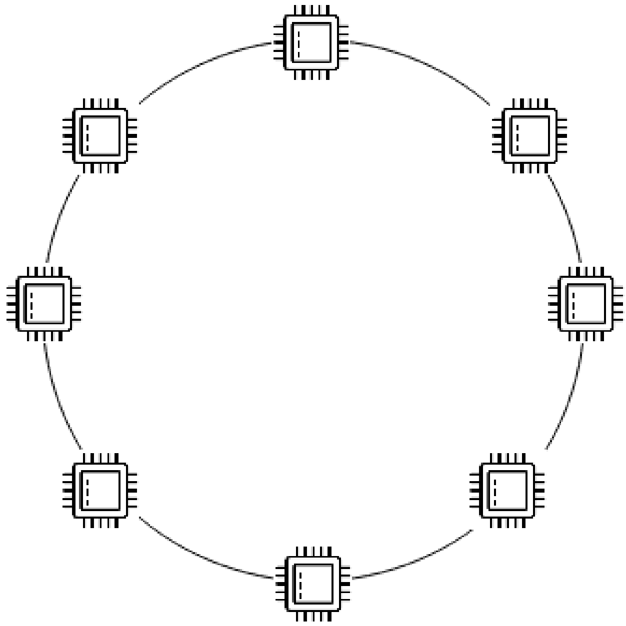

Figure 2.3: Ring Topology

As each node is connected to only two other nodes, data transfer rates may be slow as data would need to be sent through multiple devices in order to get to its destination. A large number of messages could also lead to complexities in the directions of data being transferred. This topology would not be ideal for wireless censored networks.

2.2.3 Star Topology

This Topology involves multiple nodes in the network connecting to a central hub. The nodes are only capable of communicating with the central hub and not with each other. Figure 4 shows the basic arrangement.

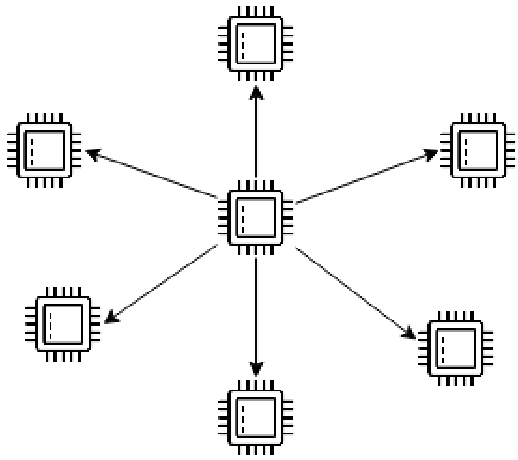

Figure 2.4: Star Topology

This topology offers flexibility in that nodes can easily be introduced or removed from the network without altering the structure of the network. This also makes it ideal for wireless networks due to the nature in which they operate. A possible drawback in this network scheme is that for one node to communicate with another node, the message would need to be sent through the central device which will route the message to the intended target. This could add unwanted latencies, especially for a time dependent application.

2.2.4 Mesh Topology

This topology involves nodes in the network being interconnected with each other. Communication can originate from any node and in the same way, any node can receive communication. In a WSN, this flexibility is highly desired because a signal that originates from a node propagates out and any receiver that is within range can be reached.

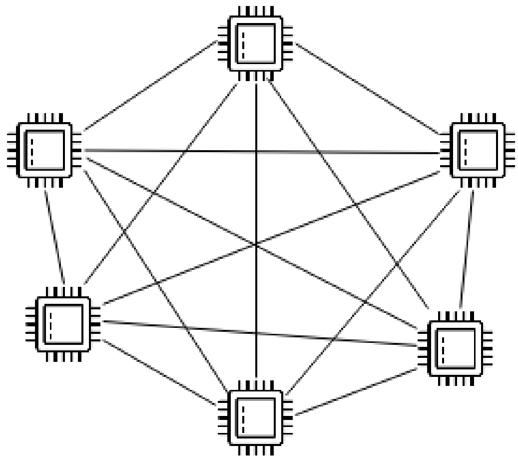

Figure 2.5: Mesh Topology

This topology offers even more flexibility than that of star topology which makes it even better for use with wireless networks. As the transmitting signals propagate, any node within range would be able to receive the message. If the message was not intended for the node, then it would simply ignore it. As with the star topology, new devices can easily be added or removed into this topology and latencies are mitigated because messages no longer need to travel through the central node.

2.2.5 Network Topologies with LoRa

As mentioned earlier, the LoRaWAN protocol employs a star of stars topology. This entails having multiple gateways that would connect to the server, and each gateway having its own specific set of sensor nodes. There are other variants of network topologies that have been developed making use of LoRa modulation. Ultimately, each variation is used to optimise the efficiency in the specific use case.

The mesh topology has been used for multiple low area WSNs. This approach has also been employed and analysed with LoRa [17] to extend the range of the sensor networks. The LoRaWAN architecture has also been modified to involve different variations of meshed topology as seen in [18]. The paper presents three different hybrids of mesh and star topologies as variants to the traditional LoRaWAN technologies. The goal was to decrease the number of gateways needed to achieve the same functionality, as well increase the range covered by traditional WSNs.

2.3 Internet of Things used in Farming

With the wireless connections and networking topologies looked at, the next aspect to consider is the involvement of internet of things technology in the agriculture industry. Farming as a practice represents a constant state of monitoring different conditions for both crops and animals. This poses a great potential for employing low powered, wide area sensor networks to aid farmers in this regard. There exists a vast amount of conditions that could be measured and acted upon by farmers. These include things such as soil moisture content, soil and air temperature, humidity, animal tracking and more. In addition to sensing and monitoring, automating systems such as watering the land would ease the workload of farmers. Data garnered could also be used to produce models that predict various conditions, which would provide further insight for the farmers.

LoRaWAN based systems that target this potential have been designed and proposed. There are nuances in the methods and equipment used, but the underlying concepts are the same. It involves some sort of data collections, be it crop production [19] [20] or animal tracking [21]. Data is usually collected by the sensor nodes, sent to the gateways and then uploaded to servers for interpretation. At this stage, the farmer has free range on how to analyse the data. As mentioned, the data could be used to produce models for prediction, but also keeping it more simply, the farmer could view the trends of the conditions of the farm, and make decisions based on this. An example of trends being observed is in the use case presented by [20].

Challenges experienced with the implementation of sensor networks in farming include extending battery lives and reliable real-time monitoring. A big need for real-time monitoring lies in the sensitivity of information being monitored - a farmer would need to act quickly to save their farm. As such, having equipment and methodologies that are as accurate as possible would be of utmost value. A system with large sources of error, or a system prone to failure would prove more harm than good to the farmer.

Taking specific attention to animal tracking, there exists potential for differing strategies. The biggest problem that has been attempted to be solved is in the actual localisation of the sensor nodes. Depending on the technology and budget available, systems have used GPS to locate the position of the animals [21], others have also attempted to use localisation techniques by use of trilateration. There also exists hybrid works that involve using both GPS for broad localisation, and then using an IoT based system to eradicate inaccuracies [22] The former poses problems because of the high power requirements from a GPS module. Furthermore, ensuring that a private network - extending to the localisation - is kept would keep costs down, which is a very large aspect of the system that is to be designed. Developing one's own localisation system poses an ever larger range of decisions. This will be discussed further in section 2.4.

2.4 Localisation

As the main function of the system would be to produce animal tracking, particular attention needs to be made to the options available to achieve this. The ability to compute the location of a sensor in a sea of nodes is the main problem to be solved. There are two main approaches to sensor localisation that have been investigated in detail in the past. Each has proved to be a reliable and successful option, pose a particular set of strengths over the other, and ultimately the choice lies in the particular use-case, as well as the designer. These two approaches are namely range-based [23] localisation and range-free localisation [24]. This section looks at them both, presenting an argument for each.

2.4.1 Range-Free Localisation

The main fact attributed to this kind of localisation is that the physical distance is not explicitly measured. Rather, the sensor nodes are localized relative to their location in the network, and their subsequent distance to a central node. This approach is useful for large scale sensor networks that contain a multitude of sensor nodes. This type of localisation is not highly accurate when compared to range based localisation [25].

The are different algorithms that have been proposed and implemented that employ this technique. The main ones are Approximate Point in Triangle (APIT), DV-Hop, Multi-hop ad Centroid algorithm. All have their own merits and advantages based on their use case, as found by [25] and [26]. Although this method is generally easier, cheaper and less resource intensive to implement than range-based localisation, the reduced accuracy is a problem for this particular use case. The use of distance measurements and subsequent accuracy is required to obtain the tracking functionality sought after.

2.4.2 Range-Based Localisation

For this method, the important parameter for localisation is physical distance and thus makes it relevant to this project. This can be found in various ways and the most common are Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA) and Received Signal Strength (RSS). The underlying principle put in place is to use one of these methods to calculate the distance between a sensor node and a gateway. From there, the localisation can be done mathematically either by using trilateration or angulation [23]. It is important to note that although this localisation scheme is more accurate, it is very heavily linked to line of sight (LOS) measurements, and the distances measured could lose accuracy if the signal bounced off obstructions.

To successfully localise a target, one would need to make use of at least three different gateways. This is explained in further detail as follows.

Trilateration

In trilateration, each of the gateways measures the distance between a given node and itself. The direction of this distance is unknown, and therefore it represents the radius of a circle (in a two dimensional space) or a sphere (in a three dimensional space). For simplicity, the 2-D analysis is examined.

With two gateways, the circles would intercept at two points, giving two possible locations for the node. A third gateway would eliminate one point, and the intersection of all three circles would be the location of the node in question. This is shown in figure 2.6.

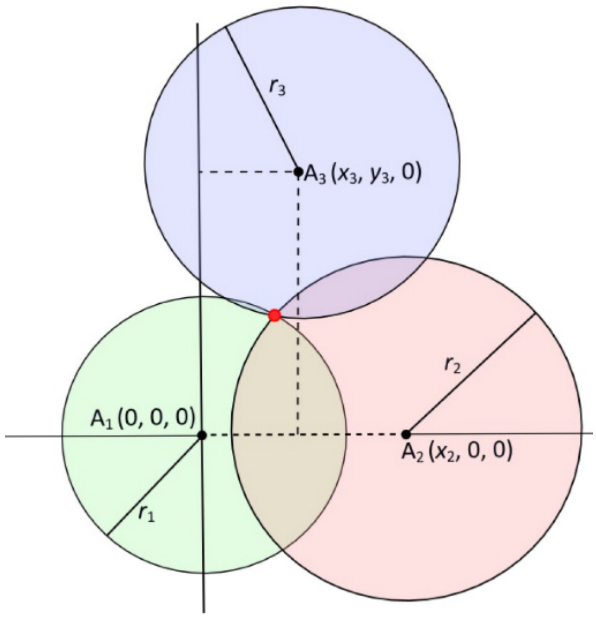

Figure 2.6: 2-D Simplified Trilateration [27]

Analytically, this can be found via a series of equations. If the node is located at (x, y) and the gateway positions are at $(0, 0)$, $(x_2, 0)$ and (x_3, y_3), then the equations for the respective circles can be described as:

$$r_1{}^2 = x^2 + y^2 \tag{1}$$

$$r_2{}^2 = (x - x_2)^2 + y^2 \tag{2}$$

$$r_3{}^2 = (x - x_3)^2 + (y - y_3)^2 \tag{3}$$

These equations can be simplified, and therefore the co-ordinates of the node would be given by:

$$x = \frac{r_1{}^2 - r_2{}^2 + x_2{}^2}{2x_2} \tag{4}$$

$$y = \frac{r_1{}^2 - r_3{}^2 - 2xx_3 + x_3{}^2 + y_3{}^2}{2y_3} \tag{5}$$

Angulation
Also commonly known as triangulation, this method involves using bearings of the point in question relative to two known receivers points, and a reference direction [23]. The minimum number of gateways required to implement this would therefore be less, however additional hardware would be required in each gateway to detect angles, as explained later in this section. Figure 2.7 below shows the basic set up to achieve angulation.

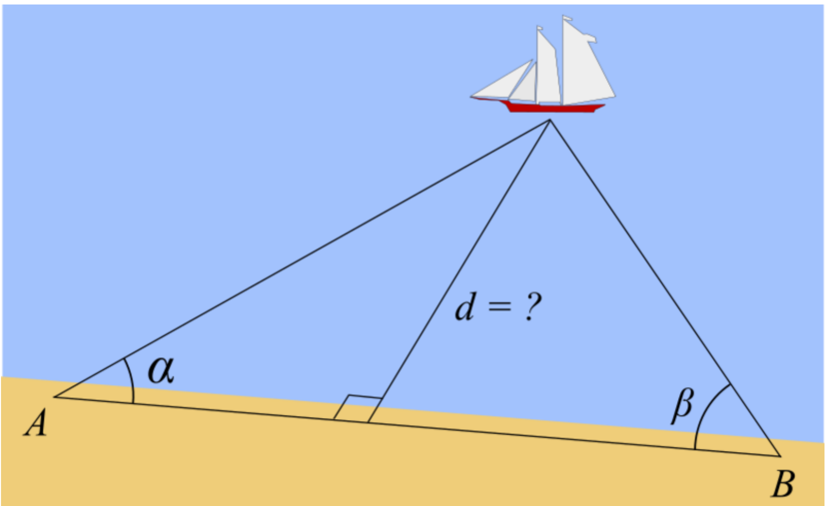

Figure 2.7: General Use case for Triangulation [28]

As the figure shows, the distance between the two gateways would also be required to be known. The variable to be found from this would be the distance of the node in question from the line connecting the two gateways, which will be denoted as d. The distance between two gateways is denoted as l and the angles the node makes with each gateway are α and β. Using the trigonometric properties of a right angled triangle, the distance l is represented as:

$$l = \frac{d}{\tan \alpha} + \frac{d}{\tan \beta} \tag{6}$$

Rearranging and simplifying the equation, the required distance d from the line connecting both gateways can be found with the following equation:

$$d = \frac{l \sin \alpha \sin \beta}{\sin \alpha + \beta} \tag{7}$$

Time Of Arrival

As the name suggests, this method involves using a timestamp at the time at which a message from the node was received by the gateway. The message itself contains a timestamp indicating when it was sent and the time difference between these two is used to calculate the distance between the receiver and the sender using the simple kinematic equation 8.

$$Distance = speed \times time \tag{8}$$

As would be expected, this method requires very specific timing requirements in order to be implemented. Additionally, since the signal speeds would be the speed of light, these timings would need to be incredibly precise because a small drift in timing would lead to a large discrepancy in the distance calculated.

Time Difference Of Arrival

The TDOA technique involves similar mechanisms to the TOA technique. The sensor node broadcasts a message containing a time stamp to the different gateways. It however differs in the way the distance is calculated. The different gateways log the time that the message was received, and the difference in these timings is used to calculate the location of the node.

Angle Of Arrival

The AOA method uses the angulation method shown in figure 2.7 rather than trilateration to compute the location of the sensor node. Once again, the sending node broadcasts a message, and the receivers log the angle at which the message is received. This is done by having multiple receiving antennae on the receiver to accurately compute the angle. The additional components for this method differ from the previous two. Instead of accurate timings required, the angle of the signal received needs to be computed. This would entail an antenna array being used.

Received Signal Strength

The final method to calculate distance is based on the signal strength of the received signal. The message sent by the node can contain any form of data, such as sensor data. The receiver needs to be capable of reading the strength of the signal at the antenna. In addition to this, the gains of both the sender and receiver antennas would need to be known in order to accurately calculate the distances. The received signal strength is a function of distance from the emitting node and other variables such as signal interference. The relationship between distance and signal strength is:

$$RSS \propto \frac{1}{distance^2} \tag{9}$$

The received signal strength can be easily read by the microcontroller. This method of localisation has been implemented before on LoRa based systems [24] and it was found that the accuracy can vary very easily. It has been widely observed that RSSI based localisation is not as accurate as timing based localisations, due to hindrances such as interference and frequency hopping. For accurate results then, it is better to use a time based localisation technique.

2.5 Timing Considerations

The method chosen for localisation is Time of Arrival. This means that the timings in the system are critical for tracking performance and particular attention needs to be made in making sure that there are no uncertainties in the timing data. Without the nodes and gateways being synchronized to the same clock, the data used to calculate the distances would lead to incorrect results. For example, if the clock on the sensor node is a millisecond ahead of the clock on the gateway, the computed distance can be as much as 300km off. This difference in timings between nodes can be referred to as clock offset, which would need to be accounted for. Another factor to consider when synchronizing is the existence of clock drift due to the unavoidable discrepancies in hardware. Clock drift occurs when the two processors operate at differing frequencies. Normally this would be acceptable, but for the sensitive requirements of achieving accurate distances, this could lead to great error. As a result, the synchronization steps would need to be taken periodically, to make sure the clocks are constantly synchronized.

The two considerations to account for are therefore the clock offset and clock drift. There have been various papers that have been involved in developing timing synchronization algorithms for wireless sensor networks. A simple lightweight algorithm was proposed in [29] however it accounts for a clock drift of about 4.5ms, which as explained earlier leads to a very large error in distance calculated. However, this method could be refined further to reduce the allowable clock drift to meet the required execution. A survey of the common synchronization algorithms is performed in [30]. Each vary in implementation, however they achieved the same goal.

2.5.1 Sources of Differences in Timings

Transmitting data packets wirelessly in a network contains inherent overheads. These overheads include sending and receiving times, as well as propagation delays. The propagation delay is the specific overhead that is used to calculate distance between sensor nodes in a TOA localisation scheme. In a wireless network that is not highly time sensitive, these overheads bear little to no effect on the functionality of the network. However in a very time sensitive application such as this, the goal required is to minimise and mitigate these difference as much as possible.

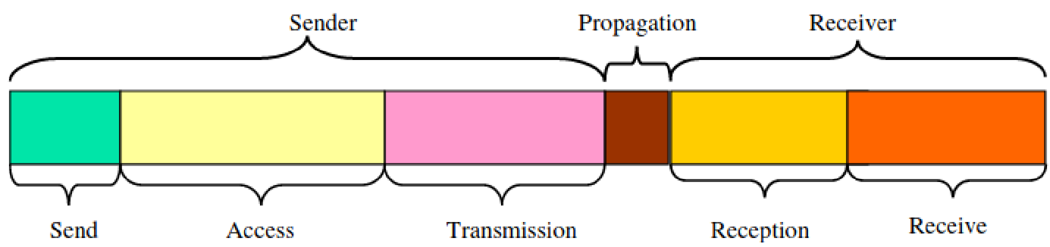

Figure 2.8: The Overheads in sending a Signal Between two Wireless Nodes [31]

Figure 2.8 outlines the various sources of overhead errors. The sizes of each overhead are not to scale, however they are used to show an intuition of the magnitude of each overhead. It can be seen that the propagation delay is much smaller when compared to the processing and transmission/reception times of the nodes. This is expected as the former is usually in the order of nanoseconds, whereas the latter is in the order of microseconds and milliseconds.

On the sender side, the processing overhead stems mainly from the delays from the time taken once it has decided to transmit to once the transmission is complete. This includes generating the message for transmission, sending it to the transmission medium, and finally the physical transmission through the antenna. On the receiver side, the delays are slightly less as only physical reception time and decoding time are a factor. [31] Minimising these multiple sources of overheads are a vital task, however there is a physical limitation placed on them, thus rendering their existence unavoidable.

2.5.2 Synchronisation Protocols

The two main synchronization protocols that have been developed are:

- Reference Broadcast Synchronization [32]
- Timing Synch protocol for Sensor Networks [31]

Each aim to eradicate the error overheads mentioned previously, but achieve this in slightly differing methods. It is important to note that both protocols do not account for clock drift, and so would be inefficient in the presence of clock drift.

Reference Broadcast Synchronization (RBS)
Traditional syncrhonization algorithms are from sender to receiver. RBS acknowledges the fact that a large proportion of error stems from the sending device, and so avoids the error in timings from the sender side by synchronizing two receivers together. This is shown in figure 2.9.

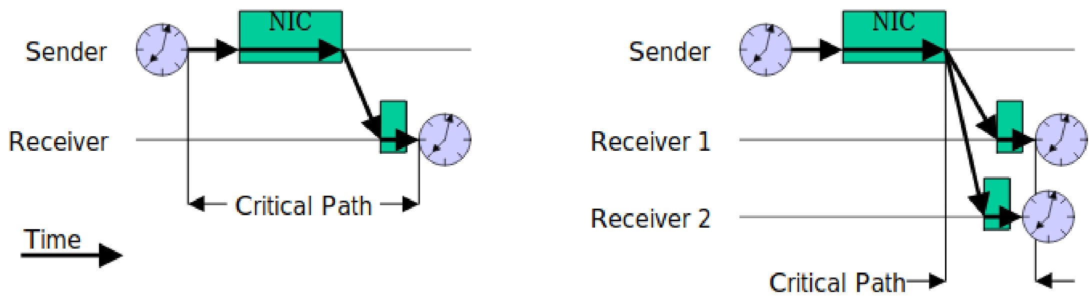

Figure 2.9: Comparison of RBS with other Synchronization Protocols [32]

The operation in RBS is simple. The broadcast message sent at the start need not contain any timing information as it is simply a reference. The receivers log the time at which the message has been received and then exchange times with each other. They then estimate their clock offsets and at that point either a local reference clock is used or a global clock is used [27].

Timing Synch Protocol for Sensor Networks (TPSN)
This is a more traditional synchronization protocol that involves both sender and receiver requiring to be syncrhonized. It makes use of a two message communication between two nodes, and attempts to calculate the overheads involved. The application of the protocol is to structure the nodes in a network to differing levels in a hierarchy, and then having each level synchronize with the level above it. As such the protocol contains two phases, the first being the arrangement of the levels and the second being the actual syncrhonization.

Focusing on the synchronization, the process is as follows:

1. A node of a certain level sends a time stamp to a node in a higher level.
2. The receiving node logs the time that the first message was received and immediately sends a message back to the original sender. This message contains the original time stamp, the time stamp of the time the message was received, and the time stamp at which the second message is sent.
3. The original node receives the receipt message and logs the time of receipt. Using these four time stamps, it calculates the offset and then adjusts its clock accordingly.

This process is shown figure 2.10.

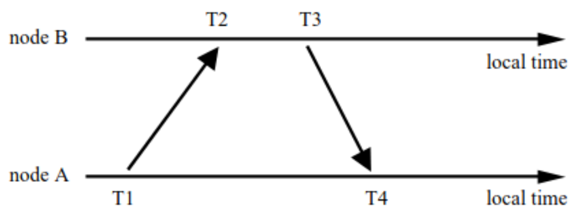

Figure 2.10: A two way message exchange [31]

The times T1, T2, T3 and T4 refer to the send and receive times of both sensors. The calculations performed by the receiving node are to find the clock offset (denoted by Δ) and propagation delay (denoted by d). The clock offset is found by:

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2} \tag{10}$$

Similarly, the propagation delay is found by:

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \tag{11}$$

With these found, the node can adjust its clock in order to be synchronized with the node in the higher level.

3 Research Methodology

The main objectives of this research is found in Chapter 1. In summary, it is to establish a long range wireless communication link between two nodes that can be used to implement livestock tracking and monitoring. In order to achieve this, a structured methodology with multiple phases was employed. This chapter therefore details the method used, and explains each phase in detail.

There are various process models that serve as templates for project methodologies. For this project, a hybrid of two models, namely the Spiral Model [33] and the Waterfall model [34], were used. This is because both models contain aspects that are advantageous to the project and application. In particular, the design and implementation section requires an iterative process that evaluates the results and improves on them. Conversely, the steps taken from identifying requirements to conclusions happen in order, with each phase utilising the work done in the previous phase. The methodology used is outlined in figure 3.1 below:

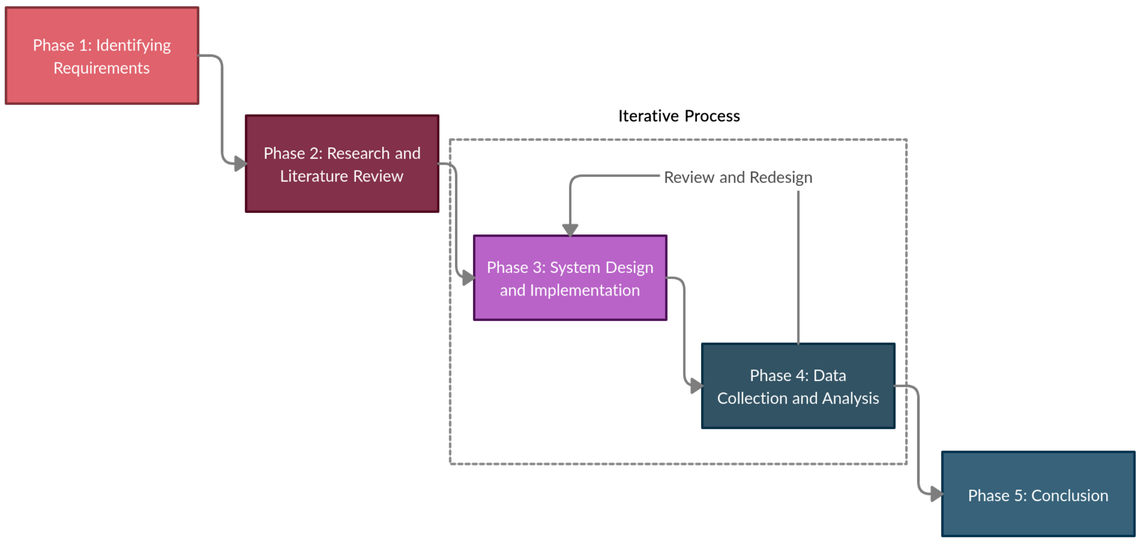

Figure 3.1: The Research Methodology Undertaken

3.1 Phase 1: Identifying Requirements

The first phase of this project was to identify the requirements. These are based on the intended use case and would be used to fuel the decisions made in later phases. The completion of this phase hinges on reviewing different possible use cases, and understanding the advantages and disadvantages that come with them. It would also be beneficial to understand the theories and common practices in livestock farming, and agriculture as a whole. This would help pin point the needed functionality of the system, in order to produce a useful prototype.

3.2 Phase 2: Research and Reviewing Literature

The next phase, once the system requirements have been set up, would be to review existing literature related to the fundamental topics involved. The purpose of this phase would be to analyse existing techniques and theories that would be implemented in the system. Going further, the goal of this phase would be to bridge the different aspects explored to combine a functioning system. This would fill in the missing gap of existing technology.

3.3 Phase 3: System Design and Implementation

The first of the phases to be iterated, this phase involves designing the and implementing the different sub systems that make up the overall prototype. Once a design is made, it would be tested. This would lead onto the following phase. Once the data is analysed, it is compared with the expected outcome and the design would be tweaked accordingly. This will be done during the development of the relevant communication development and synchronization algorithms.

The choices to be made in this phase will revolve around the requirements laid out in chapter 1. Considerations involved would be the choice of hardware, protocols and message structures.

3.4 Phase 4: Data Collection and Analysis

This phase contains two stages. The first stages is in the iterative process with Phase 3, whereby a new design is implemented and analysed. The results of the analysis may be deemed to be acceptable, at which case the specific module is complete. In contrast, the results may be unacceptable, and the design would therefore be refined. Once all the designing is complete, the second part of the phase may begin.

The second part of this phase involves going out and performing tests on the design. The tests to be performed are summarised in chapter 1, and are provided in detail in chapter 5. The specific aspects of the design tested in this stage are listed as:

1. Clock Synchronization
2. Power Consumption
3. Ranging

4. Signal Interference
5. Distance Tracking

After the tests are completed, the results are collected and analysed in chapter 6. Which concludes this phase.

3.5 Phase 5: Conclusion

The final phase rounds up the study. It presents a summary of the objectives and the design used to obtain them. It then goes on to give a brief review of the results obtained, and what they entail. The phase is ended off by a recommendations chapter that provides insight into how the developed prototype can be improved in the future.

4 Design and Implementation

This chapter details the design of the prototype system. As this project requires over the air communication between devices, the minimum requirement is to have two devices through which one would act as the gateway, and the other as a sensor node. This would be to facilitate testing of communication scheme as well as the ranging processing that would be deployed in a full network of sensors. The design approach taken was to minimise costs as well as maximise scalability. That is to say, 'design in such a way that additional nodes or gateways can be introduced to the network with minimal problems in the future.'

With a budget of around R1500 provided by the Department of Electrical Engineering, the maximum number of nodes that could be procured was 2. As a minimum number of 3 Receiving gateways are required to successfully localise a sensor node, the limitations of this prototype are therefore to create a communication scheme between the two nodes, with one acting as a gateway and the other as a field node. The desired outcome therefore would be to accurately calculate distance between nodes.

4.1 Hardware and Components

As mentioned in chapter 2, the LoRa modulation scheme is patented and as such only two companies are allowed to make LoRa transceiver modules. These are Semtech, the owner of the patents, and HopeRF who have been allowed to produce modules. Designers are able to purchase these components and incorporate them into their LoRa based projects. There are also third party companies that have produced development boards that include these modules for prototyping systems. These boards differ in different ways such as in their pricing, their functionalities and in the way they're used. There are two main types at which they differ. They are either available as a peripheral to an existing microprocessor (as a shield, or by connecting jumper wires) or as an encompassing system that contains the processor and other peripherals on the same PCB.

Of equal importance, the antenna that would be used in conjunction with the radios and microcontrollers would greatly affect the accuracy and range of the system. It is imperative to use matched antennae to maximise successful communications between modules. Considerations for the antenna to be used were the length, material, number of poles, and so on.

4.1.1 LoRa Modules as Peripherals

LoRa modules designed as peripherals come as shields or as PCB breakout boards with holes for soldering connections and antennae or on a dedicated board or connecting to a larger system on a breadboard. The most commonly used are:

- Adafruit RFM95W LoRa Radio Transceiver Breakout Board [35]
- Elecrow RFM95 Shield for Arduino Uno [36]
- Dragino LoRa Shield with Semtech SX127x LoRa Radio [37]

These modules are primarily transciever extensions to existing microcontrollers, and are therefore found for lower prices as compared to development boards. It is important to note that when compared to other transcievers that employ a different modulation scheme to that of LoRa, the LoRa transcievers are at a higher price point due to the patented technology.

Figure 4.1: Adafruit Breakout Board

Figure 4.2: LoRa Shield for Arduino UNO

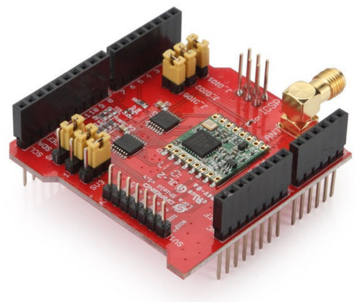

Figure 4.3: Dragino LoRa Shield

For the purpose of this project, LoRa boards as peripherals were not considered due to the additional components that would need to be sourced which would increase the expenses. Additionally, with limited access to equipment, it was decided to use a development board, which is explored in the following section.

4.1.2 Development Boards

The more applicable type of module for this project is a fully fledged development board that contains both the microcontroller and LoRa transciever. This class of development board aids greatly in the design and incorporation of LoRa into any project due to minimal setting up. The factors considered when choosing a board of this nature can be summarised as:

1. Cost
2. Microcontroller
3. Ease of Use
4. Availability in the Market
5. Power Consumption

Considering the microcontroller specifically, there are specific characteristics that are of utmost importance. These are the clock speed and memory available. The clock speed would ultimately affect the accuracy in the distance measurements and calculations done by the microcontroller, and the memory available affects how much code can be loaded onto the microcontroller. With larger memory, one can include things such as a real time operating system on the device which would handle the required processing system, as opposed to limited code that would not utilise any scheduling or other OS features and merely perform the task required.

There are three development boards that were considered for this project; The STMB-L072Z-LRWAN1 Discovery Kit [38], the Adafruit Feather 32u4 [39] and the Arduino MKRWAN 1310 [40]. A detailed comparison between these boards is shown in table II

	STM B-L072Z-LRWAN1	**Adafruit Feather 32u4**	**Arduino MKR WAN 1310**
Rated Voltage	3.3V	3.3V	3.3V
Microcontroller	STM32L072CZ	ATMega32u4	SAMD Cortex M0
Bus Width	32 bits	8 bits	32 Bits
Clock Speed	32MHz	8Mhz	32MHz
Memory	192KB Flash, 20KB SRAM	32KB Flash, 2.5KB SRAM	256KB Fash, 32KB SRAM
LoRa Transciever	SX1276	RFM95W	CMWX1ZZABZ (Contains SX1276)
Documentation and Ease of Use	Moderately Documented Relatively harder to use	Well Documented Relatively Easy	Well Documented Relatively Easy
Cost	R1613	R762	R917
Availability	Sold Out in all South African Retailer	Available in South Africa	Available in Some South African Retailers

Table II: Comparison Between the Development Boards

The three boards offer similar functionality but at different levels. The microcontrollers on the ST and Arduino have greater amounts of memory that allow space for a real time operating system, and are clocked at higher speeds which would aid in accuracy. However they are at higher price points and are not as readily found in South Africa.

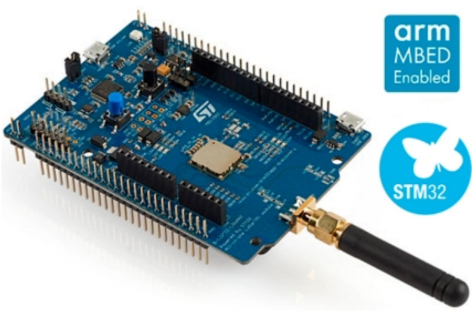

Figure 4.4: STM B-L072Z-LRWAN1 Development Board

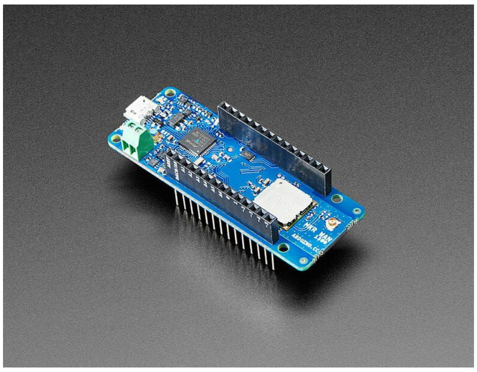

Figure 4.5: Arduino MKR WAN 1310

The Feather offers less processing power at a cheaper price point and was available in South Africa at the time of purchase. The Lower numbers do not indicate inferior performance because operations needed to achieve the required functionality can still be performed. Theoretically, the clock speed of 8MHz translates to a clock cycle of 125ns, which when converted to a distance using equation 8, leads to a sensitivity of around 38m. For experimentation, this is a sufficient sensitivity. Furthermore, the memory available on the ATMega32u4 microcontroller is limited, which would mean that it would not be suitable to run an RTOS. This is once again not a problem, as the amount of tasks to be performed are not that many. The radio module on the feather is the HopeRF RFM95W [41]. This module offers a sensitivity of -148dBm, which makes it makes it very suitable for long range communication, as expected.

Ultimately, the feather was the choice of board used. The frequency variant chosen for use was 915MHz. The feather provides sufficient specifications to meet the requirements of the project and also falls within the required budget.

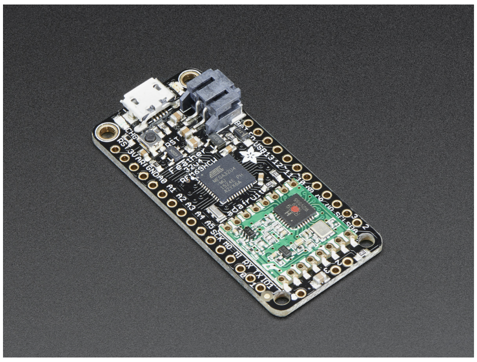

Figure 4.6: Adafruit Feather

4.1.3 Antenna

The Feather has a solder point to attach an antenna of the designer's choice. This is to be in line with the operating frequency chosen for the use case. This frequency is either 433MHz or 868-915MHz. The simplest design of an antenna is a wire of the length of a quarter of the wavelength, known as a quarter wave antenna. For a frequency of 915MHz, which is what is being used, the length of the wire is found by:

$$\lambda = \frac{v}{f} = \frac{3 \times 10^8}{915 \times 10^6} \approx 32cm \tag{12}$$

$$\therefore length = 8cm \tag{13}$$

The two options for connecting the antenna are either a simple through hole to solder the 8cm wire or solder pads for a UFL adapter in order to connect a more robust antenna. In line with the financial constraints, the former was the elected choice of antenna. Adafruit have a 915MHz simple spring antenna to accompany the feather that is priced at R12, which has a gain of 2.15dBi and an impedance of 50Ω. This was the choice of antenna used.

This choice would also eradicate human error in cutting wires of accurate length. Additionally, having matched antennas would increase the likehlihood of signals being received by the nodes specifically over large distances. Unfortunately however, It was noted that although the same antenna was ordered for both devices, the antennas provided by the retailer were not the same model and differed slightly in length. This uncertainty would no doubt affect the performance of the devices.

4.1.4 Power Management and Battery

An additional consideration in the hardware design is in the power management of the devices. Because the intended use case is in long range communication, with nodes potentially being hundreds of meters away from any power source, having a battery as a portable power source would be vital. The Feather can either by powered by a connection to the Micro USB port, or by a lithium polymer/ion battery rated at 3.7V or 4.2V. If using a battery, the feather can also charge it by connecting a PC to the Micro USB Port. It is known that the microcontroller has a maximum 3.3V Rated voltage and the batteries and USB ports exceed this, but this is mitigated by the on board voltage regulator.

For the purpose of this experiment, one 3.7V, 1200mAh Lithium Polymer Battery priced at R148.35 was chosen to power the end node. The Gateway would be powered by the connection to the computer, which would also be necessary to view the results of tests while in operation.

4.1.5 Assembled Design

The prototype design of both the node and gateway can be found in figure 4.7. Due to the comprehensive design of the development chosen development board, the assembling required was minimal. It consisted of soldering on headers to connect to a breadboard (this is not necessary, however it is recommended as it makes it easier to connect other peripherals such as sensors to the board) and the antennae. The Sensor Node also contained the battery connection.

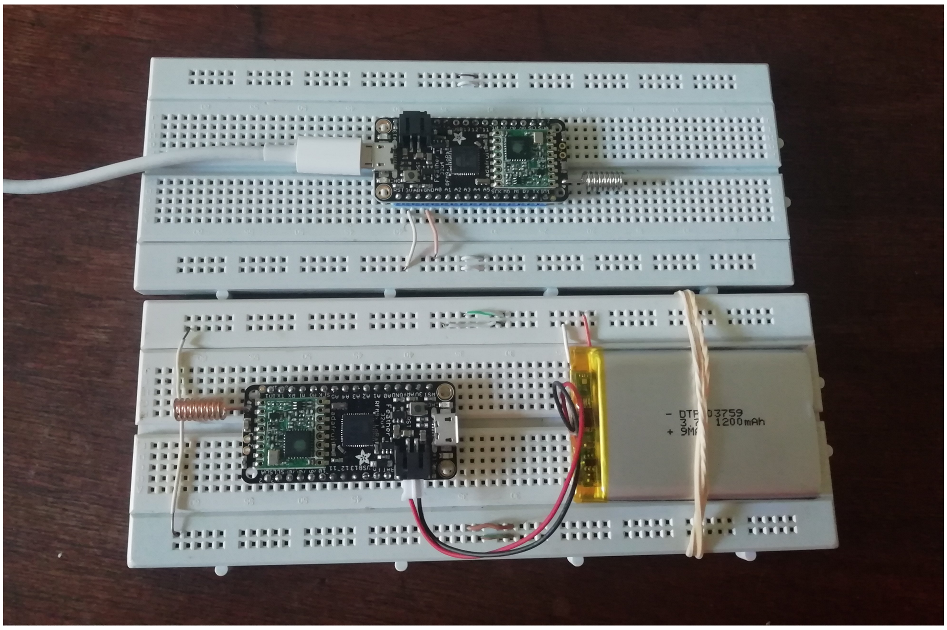

Figure 4.7: Senor Node and Gateway

Table III Shows the Bill of Materials for this prototype.

Component	**Unit Price**	**Quantity**	**Total**
Adafruit Feather 32u4	R762	2	R 1 524
Antenna	R12	2	R24
Battery	R129	1	R129
Total:			R1 677

Table III: Final Hardware Materials

4.2 Software Design

With the required hardware chosen, the next stage in the design process was to develop the embedded code that would achieve the desired functionality. In order to do this, the development environment would need to be setup. Adafruit, the producer's of the Feather board that is used, offer a large amount of support. They realise that most of the uses of the board would be for rapid prototyping - or with hobbyists - and have made it relatively easier to set up by providing a tutorial [42].

The board is included the Adafruit Library of boards that can be downloaded as a 'Boards Library' in the Arduino IDE. Subsequently, programming the ATMega32u4 Microcontroller on the development board can be done via the Arduino IDE, which increases the ease of programming massively. The programming language used is C/C++ which is most common in embedded applications. In addition to this, there also exists libraries that aid in the operation of the LoRa Radio module. The library being used in this project is the RadioHead packet radio library for Embedded Microcontrollers [43]. This library contains multiple header files for different radio modules that can be used in conjunction with an Arduino Board, or a board enabled in the Arduino IDE, such as the feather.

4.2.1 Overview

The system contained two devices, one acting as a gateway and the other as a node. They each performed differing actions to achieve the tracking and monitoring functions. As such, the firmware developed for each slightly differed. There was some similar application in both. Each were either idle, sending or receiving data, and performing operations based on the received data. This sub section looks at the overview of the operation of each device. The processes involved shall be described in the following subsections. These include setting up and synchronizing the clocks, accounting for clock drift, the structure of each message sent, and finally the distance calculations.

Gateway

The gateway operated on a tight loop, checking to see if various flags were raised, in order to perform certain operations. The flowchart of the main loop is displayed in figure 4.8.

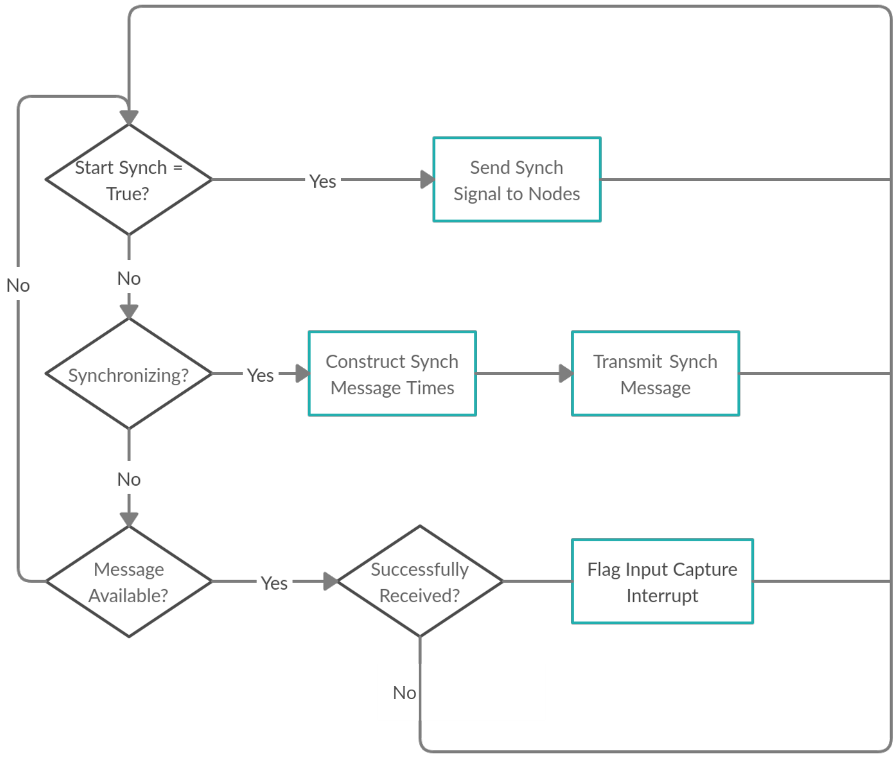

Figure 4.8: Overview of Gateway

Figure 4.8 shows that there were three main flags used, namely if it was time to start a synchronization, if it was currently synchronizing, and if a message was available. The operations for each case are explained in the following sections.

Node

As with the gateway, the node is set up in the same way. It iterates checking for various flags, operating on a tight loop.

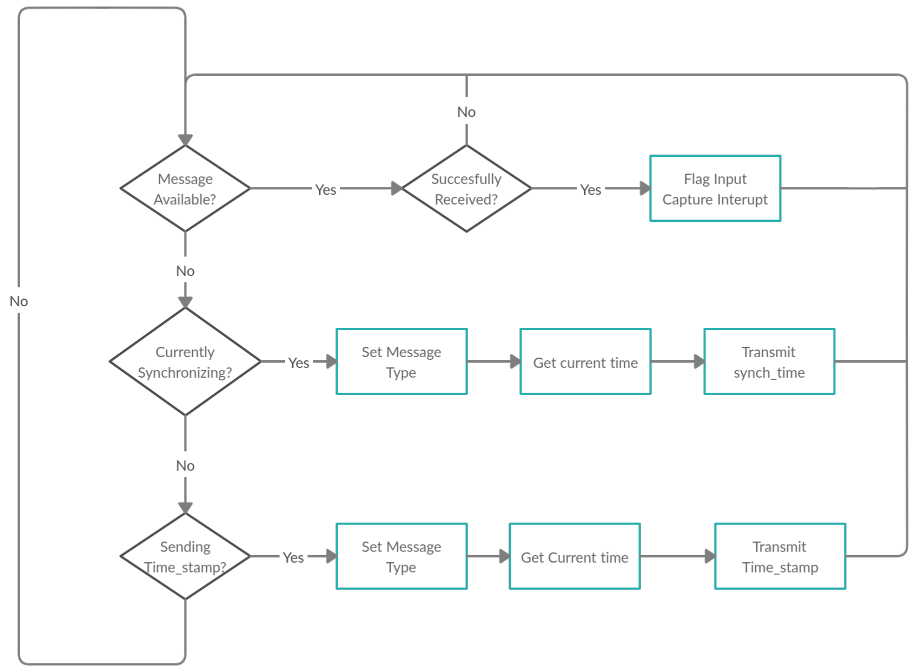

Figure 4.9: Overview of Node

The flags associated with the node are similar to that of the gateway, with the presence of a synchronizing and message available flag. The differing flag is the flag that signifies a time stamp being sent to the gateway.

4.2.2 Clock Set Up

The most effective method of clocking the system and ensuring it runs at the 8MHz of the processor is to use the built in timers and interrupts on the microcontrollers. The ATmega32u4 offers four timer/counter functions. Of these four, two of them offer input capture interrupts, which are Timer1 and Timer3. However, the interrupt pin on Timer1 has been allocated to a radio pin, which leaves Timer3 to be the choice of clock. The usual practice of using input capture interrupts is to record the rising/falling edge of an external pin, however for this project there are reasons why that is unachievable. Most notably, the LoRa module contains its own interrupt sources, which is handled by the radio driver for sending and receiving messages.

Both timer1 and timer3 are 16 bit timers, which means the maximum value that can be held by the registers is 65535 before resetting back to 0. With a frequency of 8MHz, the overflow happens at a rate that makes it impossible to synchronise different clocks. As such, an overflow interrupt was also enabled that would increment a variable that stores the number of 'cycles' the timer has elapsed. A time stamp to be transmitted can then be calculated via:

$$TimeStamp = (cycles \times 65535) + CurrentTimerValue \tag{14}$$

For identical clocks, both devices timers are set up in the same exact way. The registers that are written to are summarised in table IV. It is important to note that no prescaler is chosen in order to ensure the timer runs at the same rate as the system clock.

Register	**Bit**	**Function**
TCCR3B	ICES3	Set input Capture on Rising Edge
TCCR3B	ICNC3	Enable noise cancellation
TCCR3B	CS30	No Prescaler Value
TIMSK3	ICIE3	Enable Input Interrupt Flag
TIMSK3	TOIE3	Enable Overflow flag

Table IV: Timer3 Set Up Registers for System Clocks

As mentioned, the input capture interrupt is not triggered by a rising edge on an external pin, but rather by writing the required port bit high in software. To obtain as accurate as possible times, it was decided that this would be done as soon as a message has been successfully received, as seen in the code snippet below.

```
if (rf95.recv(buf, &len)) {
 //flag ISR
 PORTC |= (1 << PC7);
 }
```

The ISR would then perform the logic required to decipher the type of message, and therefore act accordingly. It is important to note that although this special care was taken, there is still some inaccuracies in the times, as the recorded time of receipt (and indeed sending) includes the processing time that the microcontroller interfaces with the LoRa module. This inevitably leads to clock drift, and is dealt with in sub section 4.2.3.

The Input Capture ISRs for both devices are slightly different. It is understood that an ISR should be as short as possible, and as such their main goal is to simply set flags based on the message type received, which would then be handled by the main loop. The following flowcharts outline the function of the Input Captures of both the gateway and node.

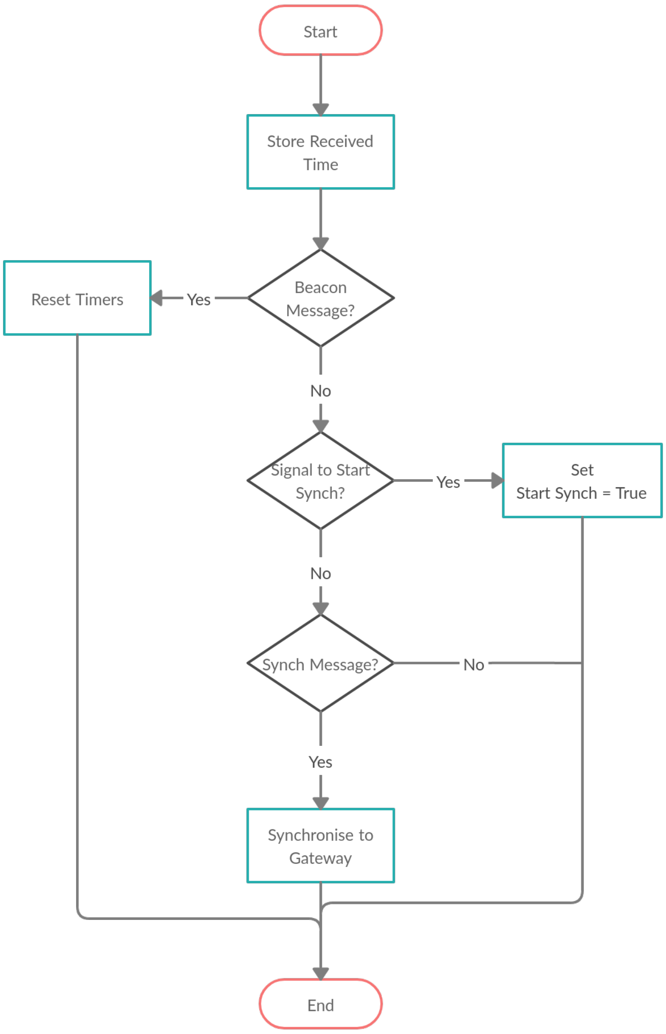

Figure 4.10: Node Input Capture Interrupt Service Routine

The Node ISR is first shown in figure 4.10. The first process is to store the time the ISR was triggered, this is done as early as possible to avoid the time being overwritten by a succeeding input capture. There are three types of message the node can receive, which are explained later in this chapter. Depending on the type of message, the node will reset its timers, set the synchronization flag or adjust its clock.

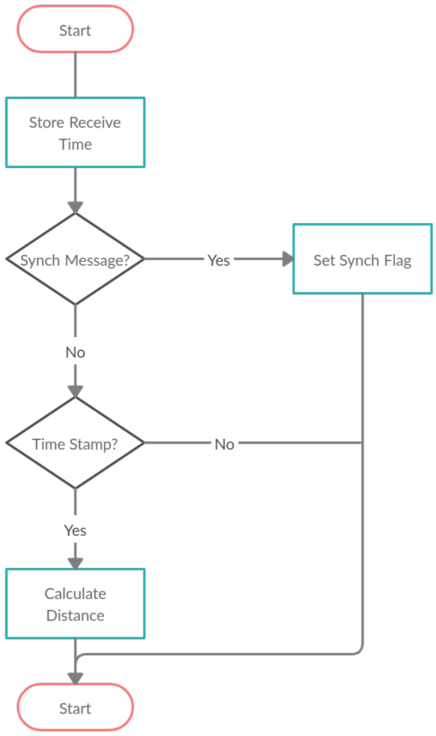

Figure 4.11: Gateway Input Capture Interrupt Service Routine

Figure 4.11 then shows the ISR used in the gateway. As with the node, the first process is to store the current clock time. The two messages the gateway receives are a synchronization message and a time stamp message. Both of which are explained later in this chapter.

Additional Clocks

In both devices, timer1 was used to create output compare interrupts. These were needed for two reasons:

1. The system needed to synchronize periodically, to ensure the clocks were in constant synch,
2. The time stamps sent by the node for disance measurements needed to be sent periodically.

In output compare mode, the timer counts to a maximum value that is described by the Output Compare Register (OCR). An interrupt is flagged at this point, and the timer is reset. The value of the OCR can be calculated based on the clock speed, prescalar setting, and required frequency. They are related by the equation:

$$OCR = \frac{ClockSpeed}{PrescalarValue \times InterruptFrequency} \tag{15}$$

It was chosen that syncrhonization would occur every 32 seconds, where as time stamps would be sent every 5 seconds. This would translate to a frequency of 0.03125Hz and 0.2Hz respectively. Using the maximum prescalar setting of 1024 and equation 15, the maximum values to be stored in the OCR were found to be 250 000 and 39 062. The OCR can hold a maximum number of 65535, which is considerably less than 250 000. This issue was resolved by multiplying the required frequency by 4, leading to a new MAX value of 62 500. Having a simple counter to count the number of times the interrupt is flagged will regain back the required interrupt frequency.

The bits that needed to be written are summarised in table V

Register	**Bit**	**Function**
TCCR1B	WGM12	Sets Clear Timer on Compare (CTC) mode which sets OCR as max value
TCCR1B	CS12 and CS10	Sets Prescalar value to 1024
TIMSK1	OCIE1A	Enables Output compare Interrupt

Table V: Timer1 Set up Registers

Summary of Clocks

Timer1 and Timer3 were used to provide the timings of the system. Timer3's function was the system clocked. The values obtained from this timer were used to calculate the distances that will ultimately be used for tracking, which is to be explained in section 4.2.6. For accuracy, the clocks would need to be synchronized as well, and this is explained in section 4.2.4. Timer1's function was for triggering the periodic events that occur in the system. This being synchronization and sending time stamps.

4.2.3 Accounting for Clock Drift

In an ideal world, setting up the clocks to run at the same frequency would lead to accurate time stamps. However, it is known that measurements are usually prone to uncertainty. The desired drift between the operation of the clocks would ideally be zero, which is practically unachievable. The goal would therefore be to have clock drift that is as minimised it can.

Initial testing of the clocks revealed a clock drift between the two devices. This was observed by utilising the 5 second interrupt to transmit a time stamp. Running at 8MHz, the count value expected to be reached at to produce 5 second intervals is found as:

$$Ticks = \frac{5}{125 \times 10{-}9} = 40 \times 10^6 \tag{16}$$

The average interval number of ticks produced by the interrupt was observed to be 39 999 887. The receiver time stamps would therefore need to be as close to this number as possible to eliminate clock drift. It was observed that the average number of ticks was 39 866 325. This constitutes a 0.33% difference, which could cause unwanted error.

The devices were then swapped, with the same code intended for the node loaded onto the feather acting as a gateway and vice versa. The goal of this was to see whether the same difference would be found. In this test, it was observed that the average interrupt interval ticks was 39 999 905, which is less than 20 ticks away from the first test and is therefore within an acceptable range. The average receive interval displayed a new average of 40 1330 264 ticks. These findings can give rise to the following conlusions:

1. The average interrupt interval time is within acceptable range for both boards.
2. In test one, the average interval time for the receiving node was shorter than the average interrupt time, whereas in test two, this value was greater.
3. The microcontollers clocks were running within acceptable range, and with the same code loaded on both, the only obvious difference between the two would be the peripherals attached.

Expanding on conclusion 3, the specific peripheral suspected to cause the discrepency would be the antenna. It was noted in section 4.1.3 of the slightly differing antennas provided by the suppliers, and this is believed to be the reason cause of drift observed.

Reverting back to the original set up, the clock drift was accounted for by multiplying the time stamp sourced at the beginning of the input capture ISR of the node by the constant that was found in equation 17. It was decided to alter to the node clock due to the fact that various nodes may need to account for the differing clock drifts when compared to the gateway.

$$Scale = \frac{39866325}{39999887} = 0.99666 \tag{17}$$

The modified code for the node received time on the node, is shown below:

```
receive_time = ((cycles * 0xFFFF) + ICR3) * 0.996665; //clock count, accounting for clock drift
```

This method of accounting for clock drift proved to only be successful in the first few minutes of start up, to which point the clock differences began behaving unpredictably, and this minor adjustment was unable to account for the larger clock drift. This observation is noted in chapter 6.

4.2.4 Syncrhonizing The Clocks

The synchronization protocol chosen to be used is a modified version the TPSN [31], as explored in chapter 2.5.2. The reason for this choice was because it allowed for increased accuracy by accounting for the errors. The protocol was slightly modified to fit the use case. The propagation delay calculated in equation 11 is the required measurement for the use case. During initial testing, the propagation delay was used to produce a constant variable that stores the delay at a distance of approximately 0m. This would then be considered the error timings during the distance calculations.

Implementing the protocol was a relatively simple task. The main decision to be made was which node would count as the low level node in the protocol. As it is expected that in the full scale network all devices are synchronized to the same clock, the original level structure would be discarded. In favour of this, a centralised gateway to which every other node and gateway would syncrhonise to in the system was chosen. Doing this allows for the different nodes in the network to adjust their clocks accordingly and independently of each other. The level discovery phase in the original TPSN protocol would also be discarded because there are only two levels in this implementation.

In the design of this system, the gateway would initiate the synchronization procedure. It sends a signal to the node indicating that it needs to synchronize, and the node begins the protocol. This is simply a synchronization flag that tells the node to begin the synchronization procedure.

As in figure 2.10 in chapter 2.5.2, there are four time stamps required to syncrhonize. T1 and T4 are produced by the node, and T2 and T3 are produced by the gateway. Upon receiving a flag to start synchronizing, the node sends T1 to the gateway, at which point the gateway includes T2 and T3. The following snippet of code shows this being done:

```
/******* Set type of message *********/
strcpy(data,"2");
strncat(data,"-",1);

/*** add in the received time_stamp from gateway to the message, T1  ***/
strncat(data, (char*)buf+1, sizeof(buf));
strncat(data,"-",1);

/*** add in the time the message was received, add in T2   ***/
sprintf(pulse_receive_time, "\%lu", receive_time);
strncat(data, pulse_receive_time, sizeof(pulse_receive_time));
strncat(data,"-",1);

/******* finally add in T3 **********/
unsigned long ack_time = ((cycles * 0xFFFF) + TCNT3) * 1.003340883746; //accounting for clock drift
sprintf(ack_message_time, "%lu", ack_time);
strncat(data, ack_message_time, sizeof(ack_message_time));
```

The node then receives the timestamps and calculates the difference in clocks as well as the propagation delay. Both these calculations are done using equations 10 and 11. The following snippet of code shows this process being done by the node:

```
//calculate delta
 delta = ((T2 - T1) - (receive_time - T3))/2;
 Serial.print("Clock drift:   ");
 Serial.println(delta);

 //calculate error
 propagation_delay = ((T2 - T1) + (receive_time - T3))/2;
 Serial.print("Propagation Delay:   ");
 Serial.println(propagation_delay);
```

With these values found, the node can then adjust the time stamps to send to the gateway. This is done by adding both the clock offset and error to the time stamp. This is done in the following line of code:

```
unsigned long transmit_time = (((cycles * 0xFFFF) + TCNT3) + delta + propagation_delay) * 0.99666;
```

It is important to note that this protocol runs on the assumption that the clock drift in the sensor network is constant. The existence of clock drift was found in this system, and attempted to be mitigated as described in section 4.2.3, however there was still drift observed. As such, this synchronization procedure would need to be undertaken periodically, in order to ensure the clocks stayed in synchrony.

4.2.5 Message Structure

The implementation contained different types of messages to be sent between the devices. The devices needed to be able to understand each message. This consisted of understanding the type of message and performing operations based on the message content. As such, a communication scheme needed to be designed.

As LoRa is focused towards long range communication, the tradeoff observed is less data available for transmission, as mentioned in section 2.1. To be precise, the maximum number of bytes that can be transmitted at a single time is 251. This is known as the payload. As such, all messages to be transmitted in this prototype would need to fall within this maximum. The different types of messages that needed to be sent are:

1. A Beacon message sent by the gateway.
2. A Synchronization flag coming from the gateway.
3. A synchronization message coming from the node.
4. A synchronization message coming from the gateway.
5. A heartbeat message coming from the node.

Each of these messages offered a similar structure, but contained different content. As such, the messages also needed an identifier. In addition to the identifier, there was an device identifier included, in order to cater for the multiple devices that would be expected to join the network. The different message structures are described in detail in this section.

Beacon Message and Synchronization Flag

Figure 4.12 Below shows the overview of the message structure for a beacon message and synchronization flag message. Both messages have the same structure, but different functionalities. The intended functionality of the beacon message is to alert the devices within range that a gateway has come online. The functionality of the synchronization flag message is to trigger the synchronization sequence of timings of node and gateway.

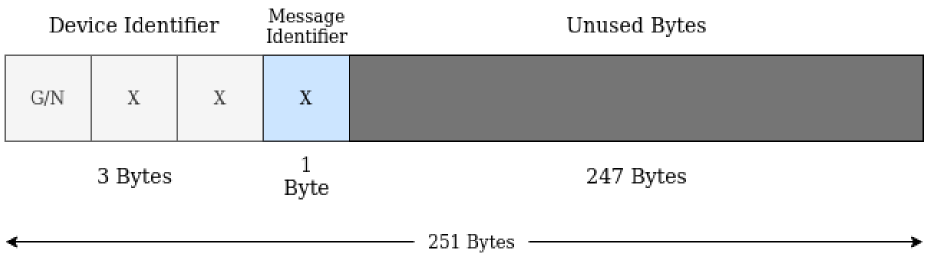

Figure 4.12: Beacon Message Structure

The first three bytes in the message denote the device identifier. The first of these bytes denotes whether the sending device is a gateway or a node, with the final two bytes being a 2 digit ID. This structure allows for multiple gateways and nodes to be included into the system. The final byte in the message is the message identifier. This tells the receiving device what kind of message it is. For a beacon message, the identifier will be ’1’, whereas for a synchronization flag the identifier is a ’2’.

Synchronization Message From Node

The synchronization message structure from the node is displayed in figure 4.13. This message is sent by the node after receiving the synchronization flag by the gateway. It consists of the 3 byte device identifier and 1 byte identifier. The following bytes contain the time of transmit by the node. The exact length of this section is unknown, as the timings to be sent are essentially the number of ticks the clock has made. This means it could range anywhere from 1 byte to 10 bytes, based on the number of digits in the transmitted number.

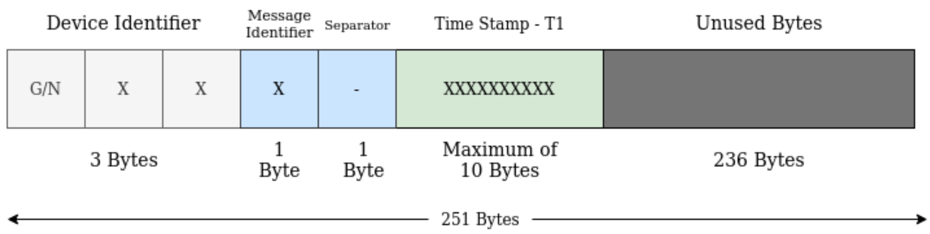

Figure 4.13: Synchronization Message from Node

The message identifier for this type of message sent is '3'.

Synchronization Message From Gateway

Once the synchronization message is received from the node, the gateway continues the synchronization protocol by including two additional times to be sent back. This is the longest message that can be sent in this prototype as it has a possible length of 37 bytes. This message structure also contains separators between the time stamps. This is due to the unpredictable length of time that has been mentioned. The separator serves as an indicator that the time stamp has been completed, and the following bytes are a new time stamp. This also allows the received message to be separated into different sections, and hence time stamps, by the receiver. The message is structure is shown below:

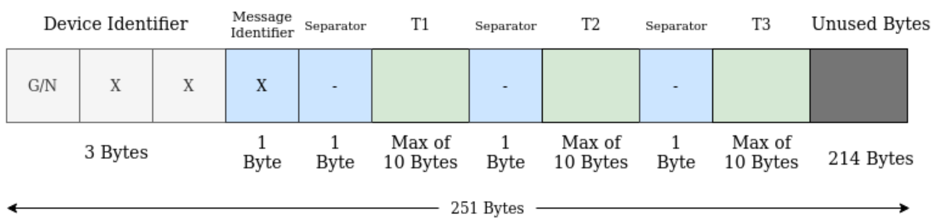

Figure 4.14: Synchronization Message from Gateway

As with the previous message type, this message as the identifier '3'. This is to signify it is the same type of message, and the only difference is has been sent by the gateway.

Heartbeat Message

This final message type is found in figure 4.15 below. It is similar to structure as the first synchronization message send by a node as it contains a singular time stamp. It is important to note that this message type is intended to be expanded on in future iterations of the prototype. The unused data bits are expected to be utilised by additional data that would come from sensors attached to the nodes. The message identifier associated with this kind of message is '4'.

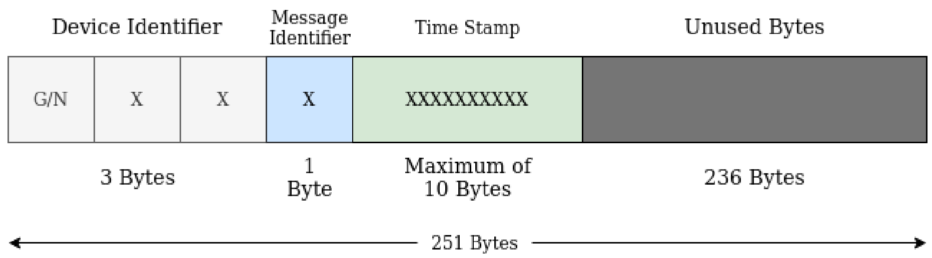

Figure 4.15: Heartbeat Message Structure

4.2.6 Distance Measurement

Using the Time of Arrival Technique, the distance measurement takes in the received time and time stamp embedded in the message in order to calculate the distance measured. The precision needed for this measurement is invaluably high, and so the error overheads involved would need to be accounted for. The process in calculating the distance is shown in figure 4.16.

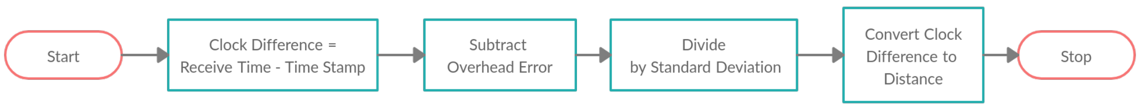

Figure 4.16: Process For Calculating Distance

The method of eliminating this error was statistical. It began with finding the average clock difference in clock counts when the gateway and node were approximately five to ten centimetres away. This almost completely renders the propagation delay overhead negligible, and leaves all the overheads to be from the transmitting and receiving process. The found average would then be subtracted from all synched messaged, in an attempt to eradicate the sending and receiving overheads.

During initial testing it was found that there clock counts suffer from constant jittering, and so this would exponentially increase error in tracking which would render the data unusable.

In order to further mitigate this error, the net different in clocks (that had the average subtracted) was further divided by the standard deviation observed from the jittering clock times. Doing this would create a buffer zone for which the cluster involved in clock timings would not affect the distance calculation.

Finally, with an integer value of difference in clock counts, the distance to be measured can be calculated using equation 8. The clock counts are converted into time, and then multiplied by the speed of electromagnetic radiation. This is shown in the code snippet below:

```
double distance = (difference * 125E-9) * 2.9925E8;     //calculating distance
```

5 Testing Procedure

Testing the prototype focused on key main aspects on the operation of the device. These were:

1. The Synchronization Procedure.
2. Range offered.
3. Power Consumption and Battery Life.
4. Effect on Range by Power consumption.
5. Line of sight (LoS) and Obstructed interference.

The radiohead driver mentioned in section 4 allows for different spreading factors and bandwidths. These parameters affect the speed of a transmission as well as the range. As such, in addition to the test cases mentioned above, an analysis of the performance of the prototype with two different radio settings was done. These radio settings favour different aspects, and can be summarised in table VI below:

Configuration	Spreading Factor	Bandwidth	Condition Optimized For
1	7	125KHz	Default, Medium Range
2	7	500KHz	Fast Data Rate, Short Range

Table VI: LoRa Module Test Configurations

As is expected, a configuration that favours a fast tranmsission time is requires less power, and therefore extends battery life. It would also be possible to change the transmit power of the LoRa module in software, which also affects the range offered along with power consumption.

5.1 Test 1: Syncrhonization Procedure

The first test to be conducted was testing the reliability of the Synchronization Procedure. As mentioned in chapter 4.2.4, the point at which the time stamp is read from the Registers on the microcontroller code is vital to achieve high accuracy. Unfortunately however, there is a limit at how soon before the transmission the time stamp can be read. Once the transmission has started, it takes a certain amount of time before it is completed. This transmission time aids to the error found in synchronization times.

This test aims to learn how well the synchronization occurs with the two different configurations as described in table VI. It also aims to view how close the clocks on both devices can get to be to each other.

5.2 Test 2: Testing Ranging Based on Tx Power

For this test, the aim is to learn how the transmit power of the node affects the range and received signal strength of the gateway. This test is carried out in a field with no obstructions. The purpose of this is to have the transmit power being the main variable that affects the range. A satellite image of the test location is shown in fig 5.1.

Figure 5.1: Open Field Test Location

The testing procedure involves placing the receiver node approximately 100m away from the sending node, and iterating between different transmit powers. These powers are 1dBm, 5dBm, 10dBm, 15dBm and finally 20dBm. The configuration used for this test is kept constant. This is to ensure the results are a reflection of the transmit power only.

5.3 Test 3: Testing Ranging the Different Configurations

This test involves iterating through the configurations described in table VI. While test 1 aims to understand the effect of the configurations on the transmit time and hence synchronization, this test aims to investigate their effect on range offered. The test location is the same as in test 2 due to the same desired conditions.

The important parameter that will indicate the performance is the received signal strength at the receiver. This will give an indication to the maximum range offered.

5.4 Test 4: Testing Received Signal Strength with Obstructions

This test is undertaken to learn the effects of the obstructions in the system. It was taken in a field that contains trees, in order to mimic a rural area or farmlands. A satellite image of the test location is shown in figure 5.2.

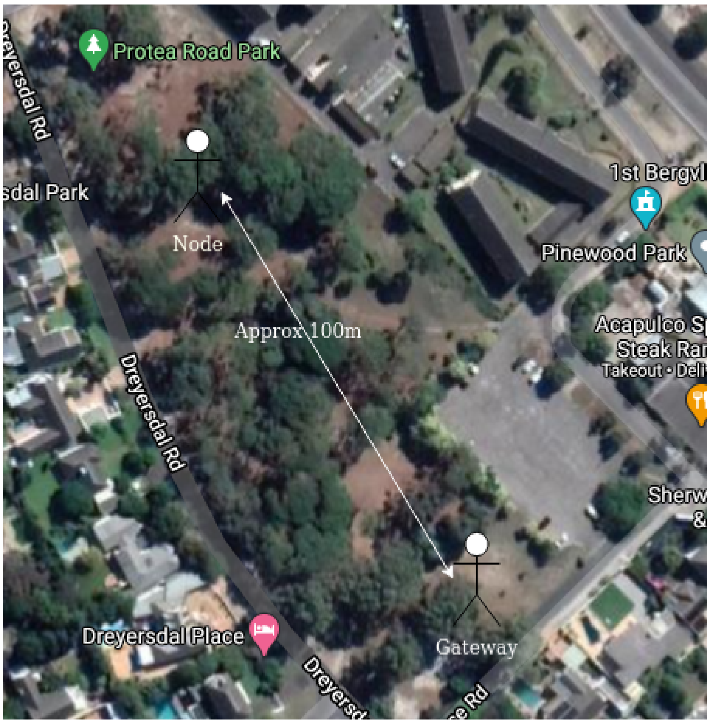

Figure 5.2: Field Test with Obstructions

For this test, the transmit power and configuration choice is held constant. The choice chosen was the one that offers optimal range with minimal power consumption. This is to mimic the use case of the system.

5.5 Test 5: Testing Distance Calculations

The final test would be to investigate the validity of the distance calculation algorithm. The procedure would involve placing the receiver approximately 30cm away from the node and then gradually increasing this distance to 100m. The desired goal would be to observe how the system reacts to the increasing distance, and if distance tracking has been achieved.

6 Results and Discussion

This chapter details the results of the tests that were described in the previous chapter. The results of each test are presented with an analysis of what they represent. The chapter is divided into 4 sections that each relate to a main aspect.

6.1 System Clock Uncertainties

The results from test 5.1 are summarised in figure 6.1. The results are obtained after the clock drift was accounted for. It can be seen that there the difference in clock counts is significant due to the increased bandwidth.

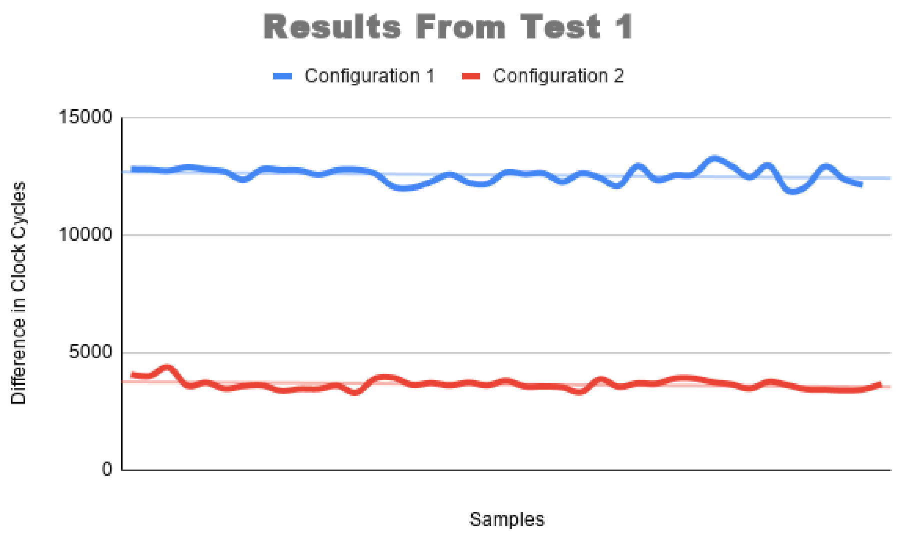

Figure 6.1: Difference In Clock Counts Between Devices

Further analysis of the clock differences for both configurations is summarised in table VII.

Configuration	Average Clock Count Difference	Standard Deviation of Clock Counts
1	12553	310
2	3654	219

Table VII: Test 1 clock Count Averages

Both figure 6.1 and table VII show that synchronization was achieved to within a few thousand clock ticks from a distance of about 5cm. In the case of Configuration 1, if the clock count is converted to time, it represents an accuracy of 1.57ms. In the case of configuration 2, the accuracy is reduced to 0.457ms. These clock differences are not constant either, as there are fluctuations in the clock counts. This could be related to the processes required to be performed by the microcontroller taking varied and unpredictable lengths. the The standard deviation also varied with configuration, which could further indicate that the cause for the errors in clock signals is in the transmission rates of messages.

Clock Drift

In both tests, it can be seen that there is a slight negative correlation in clock drift. This was obtained even after the mitigation that was designed in chapter 4.2.3. Over a short period of time, this level of drift is negligible and doesn't produce a large noticeable error due to the method used for distance calculation. The level of hardware precision for floating point numbers also limits how accurately the clock drifts can be mitigated.

After a period of constant operation, it was observed the clock drift extends to a level at which the multiplication factor (0.99666) that was calculated does not adequately cater to the error. In this state, the gateway was observed to tick at both faster and slower than expected, with no apparent cause. This phenomenom is shown in figures 6.2 and 6.3 below.

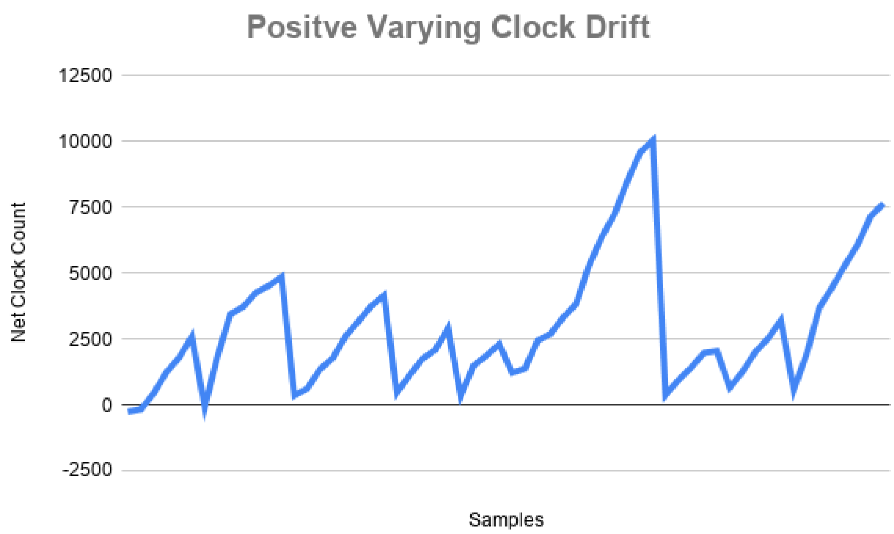

Figure 6.2: Gateway Clock Moving Faster than Node Clock

The first graph shown above portrays the gateway operating at a slightly faster clock speed than the node. This is opposite to what is observed in the second graph below. In both cases, it can be seen that after the 5 second interval, the clocks re-syncrhonize, and the clock differences reduce to a value close to zero.

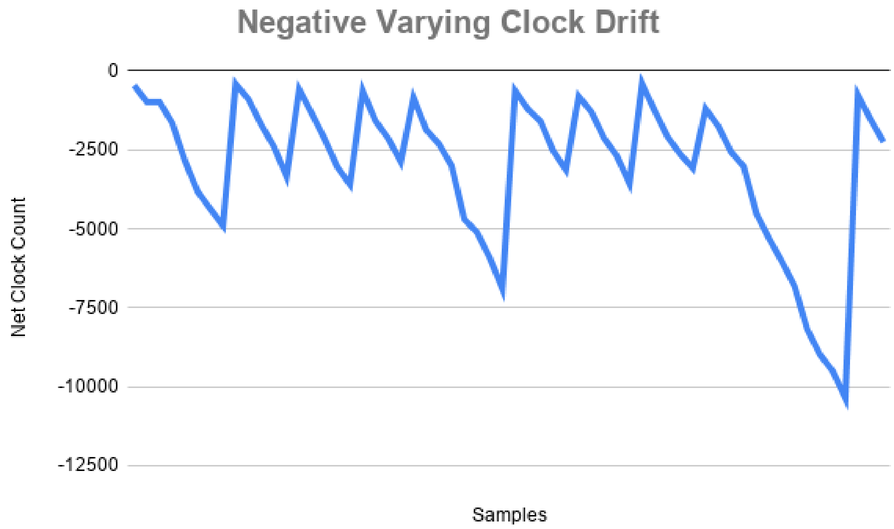

Figure 6.3: Node Clock Moving Faster than Gateway Clock

The important observations from these graphs are:

1. The clock drift is much larger than what was accounted for.
2. The clock drift is non-linear and unpredictable.

Solving this problem is not necessarily straight forward. Adjusting the multiplication factor to account for a larger clock drift would cause opposite effect at different operation times in the system, and once again doesn't factor in the possibility of an even larger clock drift at a later stage in time. A more robust procedure for dealing with clock drift would need to be devised.

A proposed solution that would increase robustness is to have the gateway constantly monitoring the clock drift between the two devices, and then proceed to instruct the node to account for it. This essentially automates the process described in chapter 4.2.3. This process would involve the gateway calculating the difference in clocks for a set number of messages stretching a few seconds into the past. These values would be averaged out, and then the multiplication factor would be calculated. Finally, the factor could be appended to the synchronization flag message (4.2.5) from the gateway to the node, at which point the node alters its clock accordingly, in the same way it does when using the TPSN protocol.

A possible error involved in this method is the lack of a device identifier in the message structures designed, which in a system comprising of multiple nodes each with varying clock drifts, would cause more confusion. The solution would be to include a destination address similar to that of the device identifier, to ensure that although each device receives a signal, they would only act on messages intended for them.

6.2 Distance Measurement

The results in the previous section show the raw times gained from the synchronized clock vary wildly and as such the time of flight would be hidden in the varying values. It was found that even after subtracting the average sending and receive overhead, the clock times were varied within a few hundred ticks. This is still not accurate enough as even a 100 tick difference would lead to a measured distance of 3750m, due to the 8MHz clock speed. This error renders the measurements completely unusable, which is the main reason for the method developed in chapter 4.2.6.

The results of this process are shown in figure 6.4 below. This was obtained after test 5 that was explained in chapter 5.5.

Figure 6.4: Results from Measuring Distance

The figure shows that even after the filtering process, the distance measured is still unpredictable and quite inaccurate. The maximum distance observed was about 112m, but due to the fluctuating nature of the clocks, it is clear that that measurement is more of a result of the errors than an actual measurement. Additionally, it is important to note that the lack of distances greater than 112m could indicate that the filtering process may work to a certain extent, but the other limitations - such as too slow a clock speed and not a large distance between the devices to see any difference, are more likely to be at fault.

Furthermore, it was originally thought that a clock speed of 8MHz yielding a maximum achievable accuracy of around 37.5m would be adequate. From figures 6.1 and 6.4 we can see that this is not achievable in practice. There exists multiple sources of error and uncertainty to produce accurate and reliable measurements in a TOA localisation scheme. It is believed the best way to increase accuracy from a purely computational method would be to have a microcontroller at a higher clock speed. It is unknown what the variation in clock counts will be, however the increased number of ticks per second would aid in accuracy. For example, using the 32MHz clock of the STM32L072CZ or SAMD Cortex M0 microncontrollers mentioned in chapter 4.1 would produce a theoretical maximum accuracy of: $c \times \frac{1}{32MHz} = 9.375m$.

If we assume the same distance measurement and filtering design is imposed on the faster clocks, the results in testing would produce distances with higher accuracy. From the experiment we can see that for a distance of 100m, the largest error is not from measuring a distance far greater than the real distance, but rather from falling short. The error could be as large as 100m, which translates to 3 standard deviations. On a 32MHz clock, that number of standard deviations would correspond to 30m.

6.3 Ranging and Power Consumption

The two variables that were changed to experiment on the ranging of the devices were the transmit power and configuration. The received signal strength indicator (RSSI) with each message was measured during testing and the average of these values are shown in tables VIII and IX. It's evident that these indicators are quite low and are in fact lower than the recommended signal strength for reliable communication, however it is important to note that from testing about 95% of messages were successfully received and processed. This can be attributed to the high sensitivity and link budget the LoRa technology contains.

Furthermore, the testing procedure involved starting the distances from 0m and gradually increasing to 100m. It was found that the RSSI in all cases reached -80dBm approximately 10-15m away from the transmitter, at which point the readings started plateauing off.

Test	Average RSSI
Configuration 1	-90.65cB
Configuration 2	-90.01dB

Table VIII: RSSI from 100m for the different Configurations

As can be seen in table VIII, there was minimal difference with the different configurations used. This could be due to the fact that the 100m used for testing was not a large enough distance at which to obtain results on the maximum range offered by the different configurations.

Transmit Power	Average RSSI
20dBm	-90.65dB
15dBm	-87.51dB
10dBm	-92.88dB
5dBm	-92.44dB
1dBm	-93.85dB

Table IX: RSSI from 100m Using Different Transmit Powers

Table IX above shows a slight trend in the received signal strength for different transmit powers. As would be expected, the average RSSI decreases at less power. There are slight irregularities, for example at 15dBm the RSSI is seemingly away from the trend. This could result from having a smaller sample size when experimenting.

The most noteable observation is that at a transmit power of 1dBm, communication still exists between the devices. This means that at about 1.25mW transmit power, a reliable connection can be created between the LoRa radios in an open field. A possible application for this finding in a agricultural context, would be crop monitoring, as this land area is akin to the test area.

Battery Life

Without access to lab equipment to measure overall power consumption, only predictions and estimations can be made in regards to battery life. According to the manufacturer, the Feather 32u4 used in this prototype draws an average current of about 13mA, with 11mA going to the microcontroller and other components on the board. They also state that the current drawn for a +20dBm Transmit power of a 20 byte long message, the measured current draw is about 130mA for 7ms [42]. Finally, the current drawn while receiving a message is rated at 10.3mA [41].

In chapter 4.2.5, the largest payload size that can be sent by the node is 15 bytes long and so if we assume an upper limit of 20, as per the manufacturer's experiments, we can predict a battery life with a considerable margin for error. In chapter 4.1.4 The battery chosen was a 1200mAh battery, and so that will be the battery capacity used in the calculations.

With the end node sending a message every 5s, this translates to 720 messages in an hour. It receives 2 synchronization messages every minute, and therefore 120 in an hour. Denoting the number of hours the battery can last as x, the following equation is used to calculate how long the battery powering the end-node can last before being recharged.

$$1200 = 13x + 130(720)(1.94 \times 10^{-6})x + 10.3(120)(1.94 \times 10^{-6})x$$
$$\therefore x = \frac{1200}{13 + 0.18 + 2.4 \times 10^{-3}} = 91$$

The battery life of the current prototype is found to be about 91 hours using the battery chosen. This translates to slightly less than 4 days of battery life, which is far below what is required. In order to mitigate this, the microcontroller and radio could be put into sleep modes, at which case the quiescent current drawn is about 0.3mA [42]. Using this figure, the new battery life that can be found is 2500 hours, which translates to 3.7 months of battery life.

6.4 Interference and Reflection

The results from test 5.4 are shown in figure 6.5 below. It is seen that the received signal strength varies at around -100dBm in the presence of obstructions such as trees. This received power is about 10dBm less than what was recorded for the same distance, with the same configuration and in an open field.

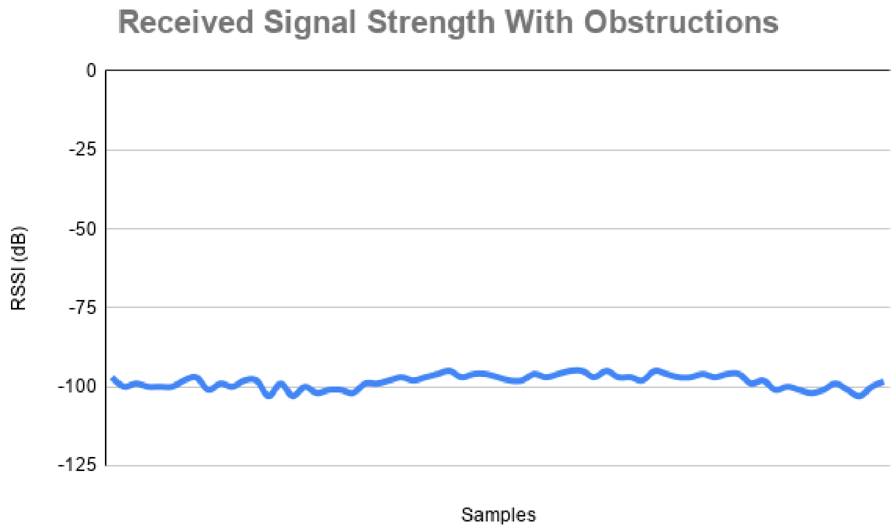

Figure 6.5: Received Signal Strength With Obstructions

The takeway from these findings is as expected. The presence of obstructions affects the signal strength, and subsequently the range offered by the system. While it is unknown the maximum range that can be attained, considerations need to be made in regard to the type of terrain the farmlands occupy. A particularly hilly area, or one with a population of plant life that could obstruct the signal would produce less coverage.

If the land being used does negatively affect the signal strength, a potential solution would be to have a multi-stage gateway system. The intermediary stages would be closer to the grazing area and would perform the distance measurements of the cattle nodes, before forwarding the packets to the highest level gateways that will perform the localisation. The benefit of this architecture lies within the fact that although there is an increased number of gateways, the synchronization with the end nodes would still occur at the lowest level. This then minimizes synchronizations errors that would occur in a massive networked system.

7 Conclusion

The purpose of this study was to investigate the possibility of using a LoRa based system to achieve reliable cattle tracking. While a high level plan was to have a networked system of devices, and hence employ a time of flight based localisation scheme, this study focused on designing and implementing distance tracking between a point to point connection between two nodes.

The most vital aspect of such a system is in ensuring the entire system is synchronized to the same clock, in order to be able to calculate the time of flight. To achieve this, an adaptation of the Timing Synch Protocol for Sensor Networks [31] was employed. It Involved a two way communication that synchronized the clock on the node to the clock on the gateway. Upon syncrhonization, the node would periodically send time stamps to the gateway at which point the gateway would calculate how far away the Node is.

A series of tests were performed with the intention of gaining information about the workings of the LoRa connection and distance tracking of the system. The parameters of importance were of distance tracking, ranging and power consumption. They produced mixed results that have lead to a series of recommended improvements.

Results

During the synchronzisation process, the obstacles encountered involved clock noise and clock drift. Their existence rendered the distance tracking to be grossly inaccurate. Mitigating these factors was imperative. Upon doing this, synchronization was achieved to an accuracy level of 1.57ms. This value accounted for the noise in clock differences and as such acted as a buffer.

The subsequent distance measurements obtained were discrete values of steps of 37.5m. This is due to the physical limitations in the hardware that was chosen. It was observed that this was too inaccurate to achieve reliable distance tracking, and therefore microcontrollers at higher clock speeds would be recommended.

The ranging and power consumption tests provided insight into the limitations of the system. It was found that obstructions such as trees would hinder the signal quality and as such would reduce the range offered by the system. In terms of battery consumption, the size of the battery plays a big part. The system is capable of drawing a total current of $151\mu A$ in an hour, which leads to a battery life in months.

Ultimately, this first attempt at distance tracking and by extension TOA based sensor localisation in a private network has proven to be an incredibly time sensitive task. While possible, it would require specific microcontrollers that are capable of clock speeds that are at least 32MHz, in order to mitigate the inevitable errors that exist in the intricate timings that are required. Additionally, The use of LoRa is adequate in terms of price, range and power consumption.

8 Recommendations

There are multiple ways that this project can be improved in future in order to obtain more reliable results, especially in regards to tracking and distance measurements. The project can also be expanded to include other features that would be bespoke to the application in question, in order to best utilise the resources in use. The key aspects identified as areas for improvement and innovation are highlighted in this chapter.

Dynamic Clock Drift Compensation

As seen in the experimentation, completely suppressing clock drift is a difficult task. Microcontrollers are rated at specific speeds, however in reality these speeds are slightly off the true value. Choosing microcontrollers that are rated at the same clock speed may minimise clock drift, but for a time sensitive application such as this, one would need to go further. As such a proposed improvement strategy would be to employ a dynamic clock drift compensation, as explained in chapter 6. This would increase robustness, reliability and accuracy in the synchronization process and subsequent localisation.

Localisation

Localisation was not addressed in this prototype, however it is still the desired end goal of this study. As such, it is recommended that once distance tracking is achieved via the method described above, the next step would be to add in further devices to form the required network. Beginning with two additional gateways, the first step would be to achieve syncrhonization to one clock, followed by distance tracking and finally localisation. Leading on from that, additional nodes would be introduced. This would achieve localisation of multiple nodes on a network, and the ground system would be completed.

Sensor Additions

Once successful localisation is achieved, the system can be expanded further to include other peripherals. As a cattle tracking and monitoring system, the monitoring aspect would involve reading various important parameters. This could include body temperature, blood pressure, heartbeat pulse, and more. These would be included into the tracking system, and be transmitted back to the gateways for monitoring. This would allow the farmer to have full knowledge of their cattle. Additionally, the system could be expanded to more than just cattle, and could encompass entire farming activities, such as crop production. The parameters to be measured in this application would include air humidity and temperature levels, soil moisture content and soil acidity, and more.

Printed Circuit Board Design

This recommendation is aimed at maximizing efficiency. The current prototype makes use of ready made development boards that contain through hole pins, which would make sensor additions bulky and requiring multiple breadboards and multiple wires. Designing a bespoke PCB that will contain only what is necessary for the application, allowing the designer to make particular selections on the components needed based on the specifications. It would also allow a robust and reliable circuit that optimises the use case as well as power consumption. This would therefore prolong battery life to an even further extent. A possibility is to have three different PCBs that account for the three types of devices. This would be one for the cattle monitoring, one for the crop production and the final one for the gateways. This would further optimise the performance of each device.

Internet Connection

The final recommendation is to connect the gateways - and hence the system - to a central server via the internet, to complete the internet of things paradigm. This would allow the system to upload to a database that would be used to store the information gained. This information could be then be used for predictions on the farm, which would ultimately aid in the productivity of the farm. The history of the farm conditions would be stored on the database, allowing for for in depth knowledge for the workings of farm, again with the aim of including productivity.

References

[1] M. Mahmoud and A. Mohamad, "A study of efficient power consumption wireless communication techniques/ modules for internet of things (iot) applications," *Advances in Internet of Things*, vol. 06, pp. 19–29, 01 2016.

[2] GSMA, "3GPP Low Power wide area TechnoLogies," 2016. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf

[3] Thales, "Narrowband IoT (NB-IoT)." [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/resources/innovation-technology/nb-iot

[4] "Sigfox website." [Online]. Available: https://www.sigfox.com/en

[5] "Lora alliance website." [Online]. Available: https://lora-alliance.org/

[6] "Lora patent." [Online]. Available: https://patents.google.com/patent/US8406275

[7] J. P. Queralta, T. N. Gia, Z. Zou, H. Tenhunen, and T. Westerland, "Comparative study of lpwan technologies on unlicensed bands for m2m communication in the iot: beyond lora and lorawan," *Procedia Computer Science*, vol. 155, pp. 343–350, 2019.

[8] S. Kim, H. Lee, and S. Jeon, "An adaptive spreading factor selection scheme for a single channel lora modem," *Sensors (Basel)*, vol. 20(4), 2020.

[9] A. Lavric and A. I. Petrariu, "Lorawan communication protocol: The new era of iot," in *2018 International Conference on Development and Application Systems (DAS)*, 2018, pp. 74–77.

[10] "The things network." [Online]. Available: https://www.thethingsnetwork.org/

[11] A. I. Petrariu, A. Lavric, and E. Coca, "Lorawan gateway: Design, implementation and testing in real environment," in *2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2019, pp. 49–53.

[12] B. Ray, "Nb-iot vs. lora vs. sigfox," 2018. [Online]. Available: https://www.link-labs.com/blog/nb-iot-vs-lora-vs-sigfox

[13] ——, "Sigfox vs. lora: A comparison between technologies & business models," 2018. [Online]. Available: https://www.link-labs.com/blog/sigfox-vs-lora

[14] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *ICT Express*, vol. 3, no. 1, p. 2017, 14-21.

[15] A. Goel, "What are network topology in internet of things (iot)," 2018. [Online]. Available: https://engineering.eckovation.com/what-are-network-topology-in-iot/

[16] C. Singh, "Computer network topology – mesh, star, bus, ring and hybrid," 2012. [Online]. Available: https://beginnersbook.com/2019/03/computer-network-topology-mesh-star-bus-ring-and-hybrid/

[17] R. Kirichek, V. Vishnevsky, V. D. Pham, and A. Koucheryavy, "Analytic model of a mesh topology based on lora technology," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, 2020, pp. 251–255.

[18] J. G. Panicker, M. Azman, and R. Kashyap, "A lora wireless mesh network for wide-area animal tracking," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2019, pp. 1–5.

[19] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, and N. Koteli, "Iot agriculture system based on lorawan," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018, pp. 1–4.

[20] M. M. Suwaid, M. H. Habaebi, and S. Khan, "Embedded lorawan for agricultural sensing applications," in *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 2019, pp. 1–5.

[21] S. Ramnath, A. Javali, B. Narang, P. Mishra, and S. K. Routray, "Iot based localization and tracking," in *2017 International Conference on IoT and Application (ICIOT)*, 2017, pp. 1–4.

[22] ——, "Iot based localization and tracking," in *2017 International Conference on IoT and Application (ICIOT)*, 2017, pp. 1–4.

[23] H. Wang, Z. Gao, Y. Guo, and Y. Huang, "A survey of range-based localization algorithms for cognitive radio networks," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 2012, pp. 844–847.

[24] Li Nian-qiang and Li Ping, "A range-free localization scheme in wireless sensor networks," in *2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop*, 2008, pp. 525–528.

[25] E. Shakshuki, A. A. Elkhail, I. Nemer, M. Adam, and T. Sheltami, "Comparative study on range free localization algorithms," *Procedia Computer Science*, vol. 151, pp. 501 – 510, 2019, the 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

[26] S. P. Singh and S. Sharma, “Range free localization techniques in wireless sensor networks: A review,” *Procedia Computer Science*, vol. 57, pp. 7 – 16, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[27] P. Cotera, M. Velazquez, D. Cruz, L. Medina, and M. Bandala, “Indoor robot positioning using an enhanced trilateration algorithm,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 110, 2016.

[28] “Triangulation (surveying).” [Online]. Available: https://en.wikipedia.org/wiki/Triangulation_(surveying)

[29] L. Tessaro, C. Raffaldi, M. Rossi, and D. Brunelli, “Lightweight synchronization algorithm with self-calibration for industrial lora sensor networks,” in *2018 Workshop on Metrology for Industry 4.0 and IoT*, 2018, pp. 259–263.

[30] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.

[31] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” *The First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 138–139, 2004.

[32] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *ACM SIGOPS Operating Systems Review*, vol. 36, 06 2002.

[33] “Spiral model.” [Online]. Available: https://en.wikipedia.org/wiki/Spiral_model

[34] “Waterfall model.” [Online]. Available: https://en.wikipedia.org/wiki/Waterfall_model

[35] “Adafruit rfm95 transceiver breakout board.” [Online]. Available: https://www.adafruit.com/product/3072

[36] “Elecrow shield for arduino.” [Online]. Available: https://www.elecrow.com/wiki/index.php?title=Lora_RFM95_Shield-_915MHZ

[37] “Dragino lora shield for arduino.” [Online]. Available: https://www.dragino.com/products/lora/item/102-lora-shield.html

[38] “Stm32l0 discovery kit.” [Online]. Available: https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html

[39] “Adafruit feather 32u4.” [Online]. Available: https://www.adafruit.com/product/3078

[40] “Arduino mkr wan 1310.” [Online]. Available: https://store.arduino.cc/mkr-wan-1310

[41] “Rfm95w lora module.” [Online]. Available: https://www.hoperf.com/modules/lora/RFM95.html

[42] “Adafruit feather 32u4 tutorial.” [Online]. Available: https://learn.adafruit.com/adafruit-feather-32u4-radio-with-lora-radio-module/

[43] “Radiohead packet radio library for embedded microcontrollers.” [Online]. Available: http://www.airspayce.com/mikem/arduino/RadioHead/

Appendix

A Code

The code used can be found at the following link: `https://github.com/marrrk/cattle-tracking`

B Test Site

Figure B.1: Eye Level View for Open field Tests

Figure B.2: Eye Level View for Obstructed field Tests

C Further Results

RSSI Graphs

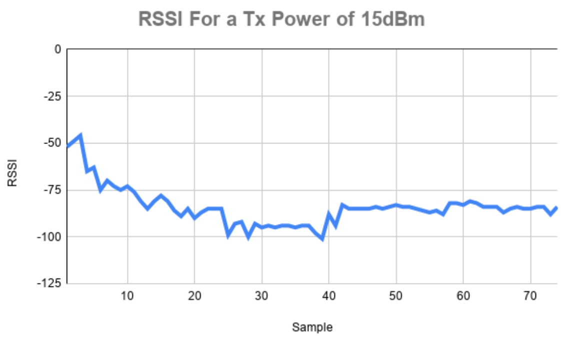

Figure C.1: RSSI for a Tx Power of 15dBm

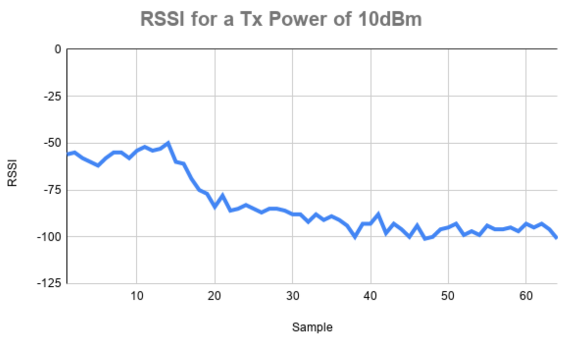

Figure C.2: RSSI for a Tx Power of 10dBm

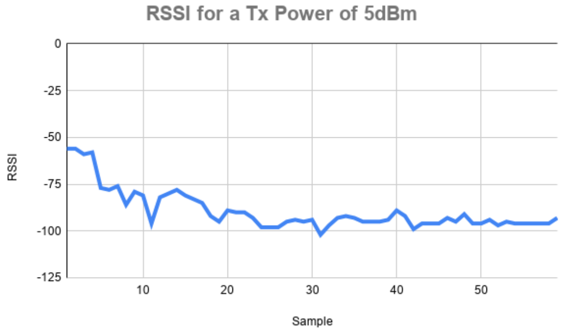

Figure C.3: RSSI for a Tx Power of 5dBm

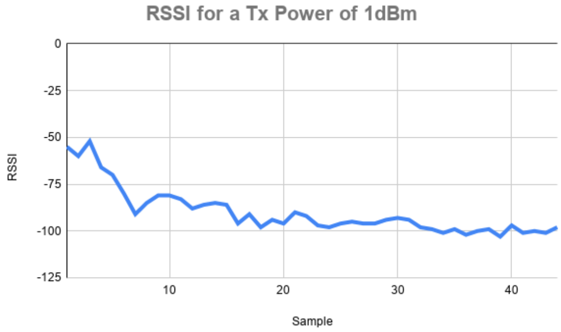

Figure C.4: RSSI for a Tx Power of 1dBm

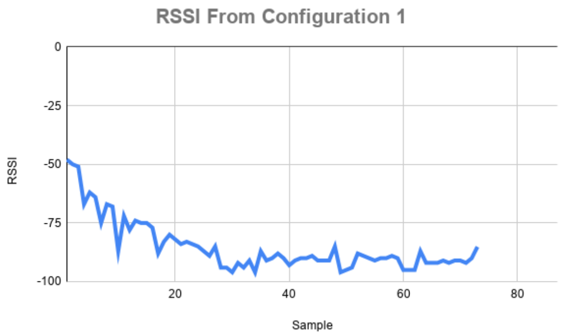

Figure C.5: RSSI From Configuration 1

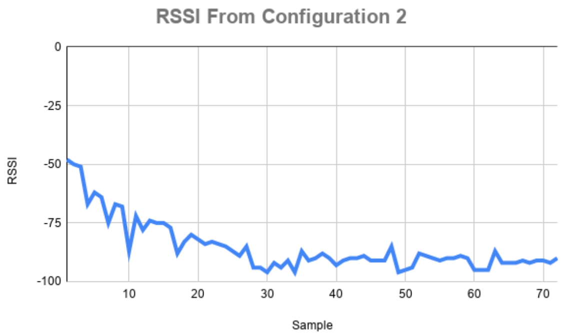

Figure C.6: RSSI From Configuration 2

Battery Calculations

Time to send message: $7ms$.

$\therefore$ Time in hours: $\frac{0.007}{3600} = 1.94 \times 10^{-6}$.

Number of messages in an hour (at message sent every 5 seconds): $\frac{60}{5} \times 60 = 720$.

Total equivalent time during a message is sent: $720 \times 1.94 \times 10^{-6} = 1.4 \times 10^{-3}$.

D GA Tracking Form and Ethics Clearance

EEE4022F/S Final Year Project	Student Name: Mark Njoroge	Student Number: NJRMAR003	DP Awarded!!
Supervision GA Tracking Form 2020	**Supervisor Comment**	**Supervisor Comment**	**Supervisor Comment**
GA 1: Problemsolving Identify, formulate, analyse and solve complex engineering problems creatively and innovatively	The initial project requirements were expolred and the student prepared an effective high-level design for the proposed system. The student is following an effective approach of refining the design and produce a functional prototype to evaluate the processing methods. The student has a good understanding of the design process, has shown competence in creative problem solving and has handled the procedural and non-procedural design aspects of the system effectively. The student is making good progress on the project, and is anticipated to submit in good time.		
ACTION REQUIRED FROM THE STUDENT: **The student needs to develop a tracking system based on a low powered, wide area network. This involves designing a networked system of sensor nodes and a centralised gateway to act as a reference point. As the system involves tracking, the main problem to be solved would be to calculate distance between sensor nodes and the gateway. This would include synchronising the timings between the sensors, or between the sensor and gateway nodes. Then, the student shall employ a communication scheme between the sensor node and gateway that will produce data that can be used to calculate the distance between the sensor node and the gateway**			
GA 4:Investigations, experiments and data analysis Demonstrate competence to plan and conduct investigations and experiments. The balance of investigation and experiment should be appropriate to the discipline. Research methodology to be applied in research or investigation where the student engages with selected knowledge in the research literature of the discipline. Note: An investigation differs from a design in that the objective is to produce knowledge and understanding of a phenomenon and a recommended course of action rather than specifying how an artifact could be produced.	The student has done good planning for this project. He decided on effective ways to do the design and pursue experiments for testing the system. Throughout the project he has followed an effective approach.		
ACTION REQUIRED FROM THE STUDENT: **The student needs to investigate the possibility of tracking based on the requirements of the project. The student would need to plan experiments that would be carried out to fine tune the parameters involved in order to achieve tracking. This would entail testing the synchronizing functionality of the devices being used, the distance measurement calculations and their accuracy, other factors that could affect the functionality of the system such as obstructions and maximum range, and battery life performance. The specific data obtained from these experiments would be analysed to understand the capabilities of the developed prototype.**			
GA 6: Professional and technical communication Demonstrate competence to communicate effectively, both orally and in writing, with engineering audiences and the community at large. This course evaluates the long report component of this outcome at exit level. Material to be communicated is in an academic or simulated professional context. Audiences range from engineering peers, management and lay persons, using appropriate academic or professional discourse. Written reports (10 000 to 15 000 words plus tables, diagrams and appendices) should cover material at exit-level. Methods of providing information include the conventional methods of the discipline, for example engineering drawings, as well as subject-specific methods.	The student has maintained a high level of professionality, for both written and oral communication, throughout the project. . In terms of the report he has shown good understanding of what is expected and has made good progress. The student has used good wording to explanations and is following an effective writing style for reporting. He has shown a good understanding of use of visualization methods. Based on the progress so far, it is anticipated that the student will complete on time.		
ACTION REQUIRED FROM THE STUDENT: **The student needs to investigate the possibility of tracking based on the requirements of the project. The student would need to plan experiments that would be carried out to fine tune the parameters involved in order to achieve tracking. This would entail testing the synchronizing functionality of the devices being used, the distance measurement calculations and their accuracy, other factors that could affect the functionality of the system such as obstructions and maximum range, and battery life performance. The specific data obtained from these experiments would be analysed to understand the capabilities of the developed prototype.**			
GA 8:Individual, team and multidisciplinary working Demonstrate competence to work effectively as an individual, in teams and in multidisciplinary environments. This course evaluates the individual working component of this learning outcome at exit level.	The project incorporates some multidisciplinary aspects, e.g. in terms of user interaction with the system. This is predominantly a scientific computing project. The student has refined the project objectives and methodology, and is maintaining a good focus on these. The student has participated adequately in interacting with others, there has been some scope for him to interact with people where the system will be installed to gain insight into desirable operation and use scenarios of the system..		
ACTION REQUIRED FROM THE STUDENT: **The project is completed individually by the student. With the supervision of an academic, the student should work on the different and challenging aspects by himself in order to meet the requirements of the project. The student should engage in responsible planning and time keeping skills - and should take responsibility for all the deliverables of the project.**			
GA 9: **Independent learning ability** Demonstrate competence to engage in independent learning through well developed learning skills. Operate independently in complex, ill-defined contexts requiring personal responsibility and initiative, accurately self-evaluate and take responsibility for learning requirements; be aware of social and ethical implications of applying knowledge in particular contexts.	The student shows good competence in working independently on complex tasks. The student has demonstrated ability to take responsibility for learning what is needed and to use the tools to develop the required system.		
ACTION REQUIRED FROM THE STUDENT: **The student is expected to engage in various resources to expand his knowledge gained during his studies to successfully complete this project. In order to achieve success, the student would also need to set out a structured timeline that must be adhered to, in order to complete the necessary milestones of the project**			
Student Name	Mark Njoroge	Internal Examiner Name	Simon Winberg
Student Signatures		Internal Examiner Signatures	
Designation	Final Year Student of Engineering	Designation	Internal Examiner
Dates	16/10/2020	Dates	16-Oct-20

Project Title	**Investigating LoRa for use in a Cattle Tracking and Monitoring System**	10/12/2020
	by **Mark Njoroge** in **EBE Electrical Submissions Undergraduate**	id. 17866215
	marksnjoroge@gmail.com	

Original submission

10/12/2020

Project Aims	**This project aims to develop a low-cost tracking system based on LoRa technology. The main goal of the research would be to develop a communication scheme such that the devices can send and receive data. Additionally, as the desired functionality is tracking, a means of being able to tell distance between devices would be investigated.**
Ethical Issues	**There are no ethical issues in this research. There are no planned tests and interactions with any humans or animals.**
Application Checklist	**Read the EBE Ethics in Research Handbook before completing this application**
Researcher(s)	**Mark Njoroge**
Department	**Electrical Engineering**
E-mail	**njrmar003@myuct.ac.za**
Status of Applicant	**Student**
Degree Being Studied (For Students Only)	**BSc Eng Mechatronics**
Name of Supervisor (For Students Only)	**Dr Simon Winberg**
Review Track	**Normal**
Motivation for an Expedited Review	n/a

Completed Ethics Application Form

Ethics_Form_-_Investigating_LoRa_for_use_in_a_Cattle_Tracking_and_Monitoring_System.pdf

SECTION 1: Overview of ethics issues in your research project	n/a
Question 1: Harm to Third Parties	**No**
Question 2: Human Subjects as Sources of Data	**No**
Question 3: Participation or Provision of Services To Communities	**No**
Question 4: Conflicts of Interest	**No**
	If you have answered YES to any of the above questions, please ensure that you append a copy of your Research Proposal (Addendum 1), as well as any interview schedules or questionnaires and consent documentation (Addendum 2) and complete further addenda as appropriate. **I hereby undertake to carry out my research in such a way that:** **1. there is no apparent legal objection to the nature or the method of research; and** **2. the research will not compromise staff or students or the other responsibilities of the University;** **3. the stated objective will be achieved, and the findings will have a high degree of validity;** **4. limitations and alternative interpretations will be considered;** **5. the findings could be subject to peer review and publicly available; and** **6. I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism**
ADDENDUM 1 Supporting documents	n/a

Research Proposal

Ethics_Proposal_-_Mark_Njoroge.pdf

ADDENDUM 2 To be completed if you answered YES to question 2 in section 1	**It is required that you read the UCT Code for Research involving Human Subjects in order to be able to answer the questions in this addendum.** **Ethical research should safeguard the interests of society and the welfare of all who participate in the research, be they individuals or groups. In this section the researcher is asked to consider the implications of their research on participants in the research. The researcher should outline risks that participants will face by being involved in the research.** **When a research involves vulnerable people, a researcher is expected to obtain informed consent from participants. This informed consent should be signed by the participants. Informed consent is intended to protect the interest of both participants and the researcher should something go wrong or should conflict arise between the researcher and the participant.**
Question 2.1: Discrimination	n/a
Question 2.2: Participation of socially or physically vulnerable people	n/a
Question 2.3: Informed consent	n/a
Question 2.4: Confidentiality	n/a
Question 2.5: Anonymity	n/a
Question 2.6: Risks of physical, psychological or social harm	n/a
Question 2.7: Payments and giving of gifts	n/a
Interview Schedule	n/a
Consent Form	n/a
Additional Comments	n/a

ADDENDUM 3 To be completed if you answered YES to question 3 in section 1	**Research may sometimes interfere with the organization, progress or advancement of communities. In this section the researcher is asked to consider the effect of their research on a community or communities involved in the research. Attention should be paid to whether the research will disrupt or interrupt the normal activities of the community and how the research will influence communities in the long term.**
Question 3.1: Community participation	n/a
Question 3.2: Termination of economic or social support	n/a
Question 3.3: Provision of sub standard services	n/a
Additional Comments	n/a
ADDENDUM 4 To be completed if you answered YES to question 4 in section 1	**A conflict of interest may compromise the conduct or outcome of a research project. It may also infringe on the interests of other researchers. In this section the researcher is asked to consider if their research may be compromised by the inclusion of certain individuals or groups in the research. The researcher is also asked to consider whether the inclusion of certain individuals or groups in the research will compromise the research of others at the university. For example, if any participants in the proposed research project are also involved in other projects at the university, have you considered if this participation will negatively affect their work?**
Question 4.1: Conflicts of interest	n/a
Question 4.2: Sharing of information	n/a
Question 4.3: Conflict of interest with other research	n/a
Additional Comments	n/a