ASHESI

# ASHESI UNIVERSITY

## Geolocation with LoRa and Machine Learning using TDOA Algorithm

**CAPSTONE PROJECT**

B.Sc. Computer Engineering

**Lisa Princesse Ikirezi**

**2021**

**ASHESI UNIVERSITY**

# Geolocation with LoRa and Machine Learning using TDOA Algorithm

**CAPSTONE PROJECT**

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

**Lisa Ikirezi**

**2021**

## DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

# Acknowledgements

To my supervisor, Francis Gatsi, and the entire Engineering Department whose encouragement and academic advice helped me undertake this project.

# Abstract

A vehicle tracking system is a crucial system used in fields such as transportation, delivery services, and criminal investigations. GPS and GSM system is the most popular means of determining vehicle location. It is accurate; however, it is costly and consumes a lot of power due to the need for frequent battery replacement. Implementing it in local public transportation, especially in African countries is not feasible. This is because installation and maintenance are expensive.

Various research has been going on to improve geolocation with LoRa. However, one of the issues that remain is the time accuracy in collecting the time of arrival which is crucial in determining the vehicle position in the Time of Difference of Arrival (TDOA) algorithm. This project designed a low-cost and low-power LoRa vehicle tracking system and researched ways to improve LoRa geolocation using the TDOA algorithm using machine learning. A multilateration algorithm was used in positioning station gateways for the TDOA algorithm to provide an accurate measurement, it required a nanoseconds precision, however, only a millisecond precision was achieved from the hardware setup. The machine learning model used RandomForest for training and prediction advantages. The lack of a microcontroller with a nanoseconds precision, limited resources to build a full multilateration system affected the accuracy of the TDOA algorithm.

## Table of Content

# Chapter 1: Introduction

## 1.1. Background

When it comes to geolocation and vehicle location tracking, different technologies such as Radio Frequency Identification (RFID), IP address, and GPS have been used in determining the location. The Global Positioning System (GPS) is a popular option, to track mobile objects, that has been on the market for over four decades. This technology uses Global System for Mobile Communications (GSM) for communication in the GPS receivers. While GPS networks operate fairly well in determining locations, it is not often adapted to local public transportation in most African countries. This is usually because of the installation and maintenance costs that come with this technology. The GPS technology market has grown its availability across various cities and countries; however, it consumes a lot of energy ranging between 30-50 mA and requires frequent battery charging [1]. Besides, the GPS network cannot be received in some parts of the world mainly rural areas. These, therefore, become challenges that limit adaptation of the technology to the local public transportation industry in most African countries.

## 1.2. Problem Definition

While GPS technology is an intelligent solution to track vehicles and help enforce security measures in the way public transportation is conducted, it is not cost-effective to the African local public transportation industry. This is mainly because the installation of the GPS receivers in all the public buses would be costly, and it would require high power maintenance to power the GPS receivers. To fill this gap posed by the GPS technology, alternative means are explored to serve the same geolocation purpose in a more cost-effective solution that would have better power usage and provide accurate data generation which would facilitate in determining vehicle locations.

## 1.3. Research Thesis

To solve a battery life challenge from GPS technology, LoRa technology is used for communication in the vehicle tracking system. LoRa technology is a wireless technology that allows the transmission of data and ensures communications in Internet of Things (IoT) networks. This technology consumes low power and ensures long-range transmissions of data [2]. LoRa covers about 15 km while GPS covers about 20 000km of distance. Even though LoRa covers a smaller distance compared to GPS, it is a better alternative compared to other wireless technology such as Bluetooth, and Wireless Fidelity (WiFI). Besides, LoRa provides a relatively good transfer rate ranging between 3-50 kbps which would be essential in creating a high-capacity network that is responsive [3].

To ensure accurate vehicle location determination using LoRa, location determination algorithms will be explored. This project will focus on the Time Difference of Arrival (TDOA) approach in determining the vehicle's location. TDOA approach is a feature of the multilateration method which determines the location by synchronizing transmitters with each other, but not the receivers [1]. TDOA approach is similar to the Time of Arrival (TOA) approach, a trilateration method feature, which is used in GPS to determine location. TOA is the most accurate approach and requires synchronization of transmitters and receivers and uses the absolute time at which packets are received to determine the location [4]. The similarity between the TDOA and TOA approaches will enable evaluation of the LoRa performance and help decide its accuracy compared to that of GPS.

TDOA approach will be used over an alternative Received Signal Strength Indication (RSSI) approach because RSSI is often affected by the nature of Terrain which affects the accuracy of vehicle locations in hilly places. The nature of this challenge makes it delicate to resolve since we cannot control the nature of the terrain. On the other hand, the TDOA approach shortcoming is mainly on the timers to provide an accurate time difference to be used in determining the distance. This project seeks to improve the accuracy of the available timers for the LoRa receivers to provide an accurate time difference which would ensure accurate vehicle locations.

To enhance the results from the TDOA approach, machine learning algorithms will be used to study the behavior of LoRa in transmitting geolocation data. This will also learn from the generated geolocation data and use it to determine accurate vehicle location and provide much quicker results.

Research in using LoRa in geolocation will give an affordable means to track vehicle locations that can be installed and used in the local public transportation. The adaptation of this technology would facilitate the organization of public transportation in African countries through the easy allocation of vehicles which would help in travel schedules, and also help to ensure security in transportation through accurate vehicle tracking. This research will also contribute to improving the internet of things networks that depend on location services and use LoRa for data transmission.

To optimize the LoRa vehicle tracking system, this project seeks to answer a question: To what extent does time enhancement and machine learning improve accuracy in the LoRa geolocation system using the Time Difference of Arrival (TDOA) algorithm.

# Chapter 2: Literature Review

## 2.1. Introduction

Over the past decade, a lot of research has gone into exploring LoRa Technology to optimize processes such as localization for vehicle tracking. Most of the work focused on implementing a geolocation system that uses LoRa and compared the Time Difference of Arrival (TDOA) algorithm and Received Signal Strength Indicator (RSSI) algorithm to determine which algorithm produces a more accurate vehicle position. According to Kelvin, the RSSI approach provided a vehicle location that was off by 40 m compared to the TDOA approach [5]. Even though Kelvin's work indicated that TDOA is more accurate, the architecture identified flaws in timing that influenced the results in the TDOA algorithm. This then provides a research opportunity to evaluate ways time can be enhanced in the TDOA algorithm to produce even better accuracy than the one proven previously.

To build on the work that has been done already, this review focuses on how to optimize a LoRa geolocation solution that consumes less power as compared to the popular Global Positioning System (GPS) technology. Also, this seeks to learn about how to reduce location errors when the Time Difference of Arrival (TDOA) method is used in determining the position of end nodes given that TDOA produces a smaller error interval compared to alternative methods such as RSSI.

## 2.2. LoRa Vs GPS

Global Positioning System (GPS) is by far the most accurate localization technology and it covers a wide area. While GPS is the conventional localization technology, it consumes ten times more energy than LoRa when localization packets are sent at the same rate and requires an extra communication module such as Global System for Mobile Communications(GSM) [ 6]. On the other hand, LoRa can perform both localization and communication and consumes minimum power compared to GPS. Hence, it acts as a more feasible alternative localization module compared to GPS.

LoRa technology is a cost-effective, low-power, and long-distance wireless communication technology. It operates per the LoRaWAN protocol and is suitable for IoT networks. LoRa technology enables a low-cost chip module, high receiver sensitivity, license-exempt bandwidth and it uses the full bandwidth [7]. LoRa architecture consists of three main sections namely, end nodes, gateways, and network servers. The end nodes could be vehicles, or mobile devices and they generate data on the parameters of the position and use LoRa to send the generated data to the base station or gateway. A gateway functions as a base station and receives signals or data from all the end nodes in its proximity. A gateway must be in a fixed place where it acts as an interface between end nodes and the network server. The gateway receives data packets from the end nodes through the LoRa modules and forwards them to the network server using the internet. The network server interprets the data from the end nodes.

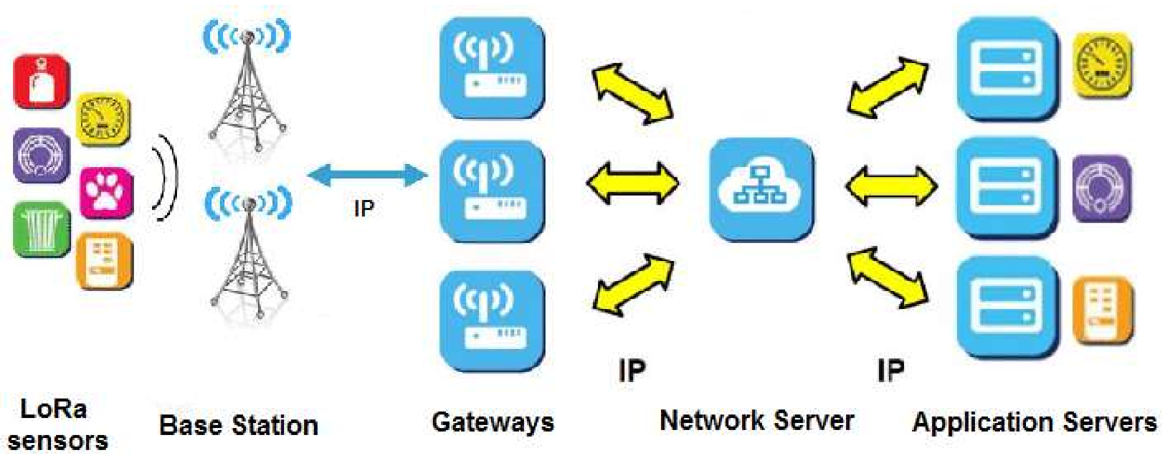


Figure 2.1. LoRa technology architecture and its main components.

Source: Adapted from [8]

Lora technology coverage ranges between 2-15 km in the urban areas and 2-30km in rural areas [7]. The interval differs in different settings because in the urban areas more obstacles obstruct the line of sight, hence affecting the length at which the LoRa module can send signals or data to the base stations. In terms of power, LoRa modules consume between 10- 24 mW [7].

## 2.3. Location Determination Methods

To determine the location of a mobile device such as a vehicle, some different methods or approaches can be used. However, these methods vary based on the nature of the environment and what would be suitable given a specific circumstance. Some of the methods that can be used include trilateration and multilateration. Trilateration algorithm functions with 3 or more base stations. A mobile device sends to the base stations in its proximity and based on the intersection point of these base stations, the algorithm determines the position of the mobile device. The trilateration algorithm depends on the Time of Arrival (TOA) or Received Signal Strength Indicator (RSSI) to determine the distance between the transmitter and receiver [9]. Time of Arrival (TOA) provides the absolute time packets are received. In the Trilateration algorithm, there needs to be synchronization between transmitter and receiver [9]. On the other hand, the multilateration algorithm uses the Time Difference of Arrival [TDOA] method to determine the position or position of a mobile device. Contrary to the trilateration algorithm, multilateration does not require synchronization between transmitters and receivers.

Time difference of arrival (TDOA) has been proven to be more accurate than RSSI, however, Time of Arrival (TOA) which uses absolute time is the most accurate. The Global Positioning System(GPS) uses TOA to determine the position of the nodes. In terms of the positioning errors produced by other methods in comparison to TOA, TDOA gives an error of around 300 meters, while RSSI gives an error in the range of 1000 – 2000 meters. Other methods such as fingerprinting produce an error of 398.4 m [1].

## 2.4. Vehicle Tracking Mechanism

The localization of vehicles or mobile devices is often done by the Global Positioning System (GPS). This system depends on satellites to determine the position of a vehicle or a node and depends on the absolute time signals that are sent using the TOA method. GPS covers a range of about 20000 km, which is the longest there is. While GPS is very accurate, it is an expensive technology that consumes a lot of power and requires frequent charging of the battery of the GPS receivers.

Alternatively, LoRa technology is explored to be used in the tracking of vehicles or mobile devices. This is based on its features of consuming low power, high sensitivity receivers, and affordability. LoRa can use the RSSI, TDOA, or TOA method in determining the position of mobile devices. Time of Arrival (TOA) is the most accurate among these with a two-way ranging mechanism where both the transmission site and receiving site communicate back and forth. However, even though this approach is synchronized and produces high accuracy location measurements its location update frequency is low [11]. On the other hand, Time Difference of Arrival uses a one-way ranging with data traveling from the transmission site to the receiving site, however, it can maintain location update rate constant [11].

The Time Difference of Arrival (TDOA) algorithm produces the time difference of the arrival curve derived from the transmission or receipt of signals between a mobile device and three or more stations [7]. This location determination mechanism requires synchronization between the base stations to produce accurate positioning of mobile devices. TDOA algorithm determines the moving device position by evaluating the difference in time of arrival at three or more base stations [12]. Below is a sketch of the TDOA architecture for position determination.

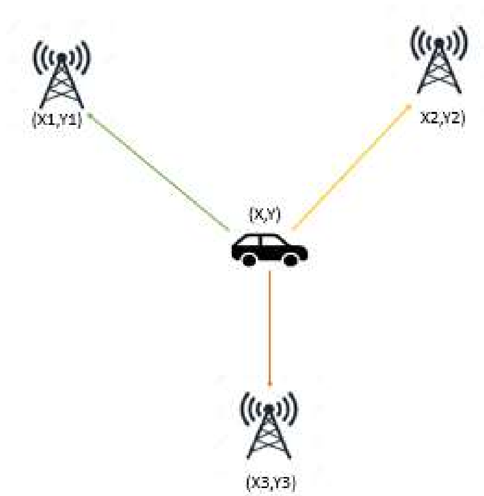


Figure 2.1. Illustration of TDOA Algorithm setup

According to Figure 2.1 above, for i number of base stations, a known position coordinate of ( $x_i, y_i$ ), the algorithm calculates the coordinate (x,y) of the moving vehicle. To determine this coordinate position, TDOA first determines the time difference of arrival between the moving vehicle and each base station using the formula below.

$$Distance = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

After determining distance for every base station, you then find the difference between the distances values as shown below:

$$Distance_i - Distance_{i+1} = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x - x_{i+1})^2 + (y - y_{i+1})^2}$$

## 2.5. Role of Machine Learning in Geolocation

To optimize LoRa technology for device tracking, research has been conducted to fill gaps existing in this system. One of the approaches is to reduce TDOA positioning error using

LoRa and a deep learning network. The positioning errors in TDOA are often based on transmission delay between end nodes and gateways [1]. In the LoRa and deep learning network approach, reference nodes are used to represent the static nodes. These reference nodes are positioned in a known place to collect data on mobile devices in their proximity using LoRa. The mobile nodes send the packets to base stations or gateways. A neural network is built, and it consistently learns the time of arrival of packets between the reference nodes and gateways. Based on the data of the time of arrival, the neural network calibrates it in the operation phase and produces a more accurate position coordinate. Tools used include Keras, SX1272 chip, and python modules [7].

This research corrected about 50% of positioning errors. However, the reduction of errors highly depended on the number of reference nodes used. Using one reference node produced more errors. Based on the simulation using four reference nodes, the error was reduced from 239.224 meters to 61 meters[7].

The gaps in this approach are on testing the accuracy of this system in different settings i.e., rural areas versus urban areas. There should be adjustable means that would make the neural network adaptable to any kind of environment and still produce accurate localization.

Another approach to optimize localization using LoRa and machine learning uses fingerprinting and to tackle limitations of LoRa technology to produce accurate localization in urban areas where the line of sight is blocked by buildings. TDOA algorithm is used to determine position, where it computes the difference between the TOAs of two gateways [3]. In this approach, a reference map is created to train the machine learning model. This reference map consists of details on the mobile node, LoRa, and GPS receiver. The presence of both GPS and LoRa receivers helps in comparing accuracy. To test this solution, a LoRa simulator was developed. The LoRa simulator generated data and created a GPS dataset and a LoRa dataset based on noisy data to represent real-life data. These datasets were used to train the algorithms of machine learning which help in predicting positions of mobile nodes or devices. For this solution to produce accurate localization, the machine learning models would require large datasets and around 10 gateways that would always be visible to the end nodes [10].

Even though this solution seems viable from the simulation and the logic, it was not tested in real life using real datasets. This, therefore, presents a gap in how the system would behave with real data. It is also projected that in case a base station or gateway changes location, a new reference map must be created, and this is a common issue of the fingerprinting method [10]. Future work on this approach would be to explore ways to reuse reference maps instead of creating new reference maps whenever a gateway moves.

# Chapter 3: Methodology

## 3.1. Requirements Specification

To build an alternative vehicle tracking system to the GPS GSM system, there are system requirements to comply with. This research project seeks to provide a more accurate, affordable means of tracking vehicles that would be adopted by local public transportation across African countries in all kinds of environments i.e. rural versus urban areas.

Therefore, the system requirements are:

- Creating a low-cost GPS-free geolocation system.
- Creating a system that consumes low power within the range of 3V to 5V.
- Designing a power system that will not require a frequent battery change.
- Designing a system that has high sensitivity to facilitate the transmission of data that is essential in calculating vehicle position.
- Long-range transmission technology to facilitate the transmission of data from a moving vehicle to a stationary gateway.
- Synchronizing the receiving antennas to ensure accurate TDOA algorithm results.

## 3.2. System Architecture Overview

To optimize the vehicle tracking system using LoRa technology, there are design decisions to make that would help in implementing and testing the solution. Based on the system outline, there are two major parts of the system namely the moving vehicle with which the system calculates the position and the station which facilitates as the gateway to collect data from the moving vehicle and eventually determining its position. Below is the picture that outlines the system architecture and the major design areas of the vehicle tracking system:

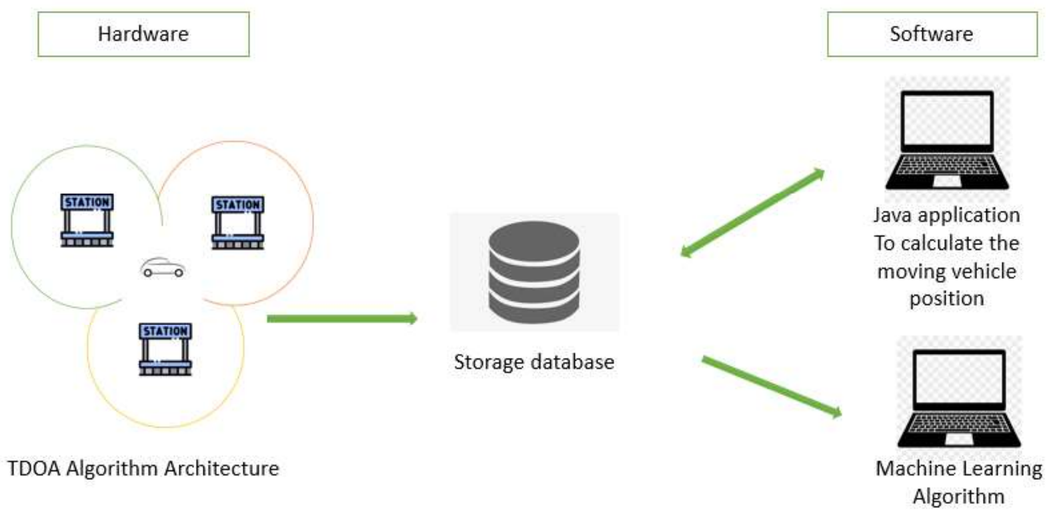


Figure 3.1. System architecture overview of the system

Figure 3.1 above, shows all the crucial sections of the project and how they interconnect. To begin with, there is a moving vehicle. This vehicle sends signals about its position and sends them to three stations near it. The moving vehicle uses LoRa in sending the data. Since LoRa transmits in a range of 0 to 20km, the vehicle transmits the signals to stations gateways within that distance. From Figure 3.1 above, the moving vehicle must be in the intersection of the three stations to allow calculation of vehicle position using the Time Difference of Arrival (TDOA) algorithm. Once, the stations receive data signals from the moving vehicle, they use the LoRaWAN protocol to transmit the data to a database which is then accessed by a java application to calculate the vehicle position based on the TDOA algorithm. The data from the database is then processed and analyzed using machine learning. The data gathered by a machine learning algorithm can be used in optimizing vehicle tracking systems and allow LoRa vehicle tracking application in real-time.

### 3.2.1. Moving Vehicle Design

To gather information on a moving vehicle, there are design components that are required to create the system. These include a power system, sensors or transceivers, a microcontroller, and means of communication.

To begin with, several microcontrollers on the market could be used in implementing a moving vehicle system that would gather data to be used. Some of the microcontrollers include Atmega 328p and ESP8266. To choose which of the microcontrollers is suitable for this system, the following factors were evaluated: cost, availability, compatibility, and complexity in addition to technical aspects such as the power consumption, the operating temperature range, and the availability of peripheral features that would allow the connection of the transceivers and microcontroller. The baseline used in making the microcontroller design decision is Raspberry pi, low power and low-cost single-board computer with the ability to communicate to UART devices. This is a low-power computer with the ability to connect UART peripheral devices. Below is a Pugh chart detailing the process of selecting a microcontroller.

Table 3.1. Pugh chart for microcontroller selection

| **Selection criteria** | **Baseline** | | |
|---|---|---|---|
| | **Raspberry Pi** | **Atmega 328p** | **ESP8266** |
| 1. Cost | 2 | 4 | 2 |
| 2. Availability | 4 | 4 | 4 |
| 3. Compatibility | 2 | 3 | 2 |
| 4. Low Complexity | 2 | 2 | 2 |
| 5. Low Power | 4 | 4 | 4 |
| 6. Operation Temperature range | 4 | 4 | 4 |
| 7. Peripheral Features | 4 | 4 | 4 |
| Total | 22 | 25 | 22 |

In table 3.1. above, three microcontrollers are rated on a scale of 4 with 4 meaning that cost is affordable, that it is available on the market, that is compatible with LoRa modules, and that it is less complex to learn. In addition, the design requires a preference for a low-power microcontroller. The Atmega 328p requires an operating voltage between 1.8 and 5.5 V while ESP8266 requires 3.3 V. Hence, they both meet the low power requirements. Both microcontrollers also have a good operating temperature range between -40 to 85 °C and UART digital communication peripherals which are essential in connecting the microcontroller to the transceivers. Since both options meet the technical requirements, the differentiating factors are on the complexity and system compatibility.

The above analysis shows that Atmega 328p rates better than the others. Therefore, Atmega 328p will be used in designing the moving vehicle system.

In addition, another design decision is on the LoRa module. LoRa module is a transceiver that would be used to transmit the signals from the moving vehicle to the station gateways. Selecting which LoRa module to use is also based on the cost, availability, and complexity. Below is a pugh chart comparing some of the LoRa modules on the market.

Table 3.2. Pugh chart for LoRa module selection

| **Selection Criteria** | **Baseline** | | |
|---|---|---|---|
| | **GPS Module GPS NEO-6M** | **Reyax (rlyr896) LoRa Module** | **RF1276** |
| 1. Cost | 4 | 4 | 4 |
| 2. Availability | 4 | 3 | 2 |
| 3. Complexity | 3 | 2 | 2 |
| 4. Sensitivity | 4 | 4 | 4 |
| 5. Transceiver Capabilities | 2 | 4 | 4 |
| 6. Low power | 4 | 4 | 3 |

| | | | |
|---|---|---|---|
| 7. Total | 21 | 21 | 19 |

In designing an alternative vehicle tracking system to the conventional GPS GSM system, the baseline for a transmission/receiver module is a GPS module. A LoRa module used must be able to accomplish several criteria that a GPS module has such as high sensitivity, ability to transmit/receive data. In this case, a GPS Neo-6M is used to facilitate the design decision. GPS Neo-6M module is a highly sensitive, low-power, and low-cost receiver which costs 10$. It requires a maximum operating voltage of 2V, and it can be stored at a temperature within the range of -40 to 85 degrees Celsius.

According to table 3.2. above, two kinds of LoRa modules are rated on a scale of 4 in comparison to the GPS Neo-6M module. The rate 4 implies an affordable cost, availability on the market, and less complexity to learn. In addition, the LoRa modules are evaluated based on technical factors such as transceiver capabilities, sensitivity, and power consumption. Transceiver capabilities are essential to ensure the system is low cost. A transceiver combines a transmitter and receiver in one device, hence reducing cost. LoRa module options evaluated are both transceivers. In terms of sensitivity, which is essential in data transmission, both modules have a sensitivity of -148 dBm making both suitable. Power requirements for the Reyax LoRa module range between 2.8 to 3.6V while RF1276 is 4.5 to 5.5V. Since power consumption is crucial to the system, we opt for a module with the least power requirements. Therefore, the Reyax LoRa module will be used in this system as a transceiver.

Below, is a diagram showing the main parts of the moving vehicle design based on the design decisions made above. A system made of a microcontroller, a LoRa module, and a power system will be created and installed on a vehicle. As the vehicle moves, the system will extract signals and send them to at least three stations in the vehicle's proximity.

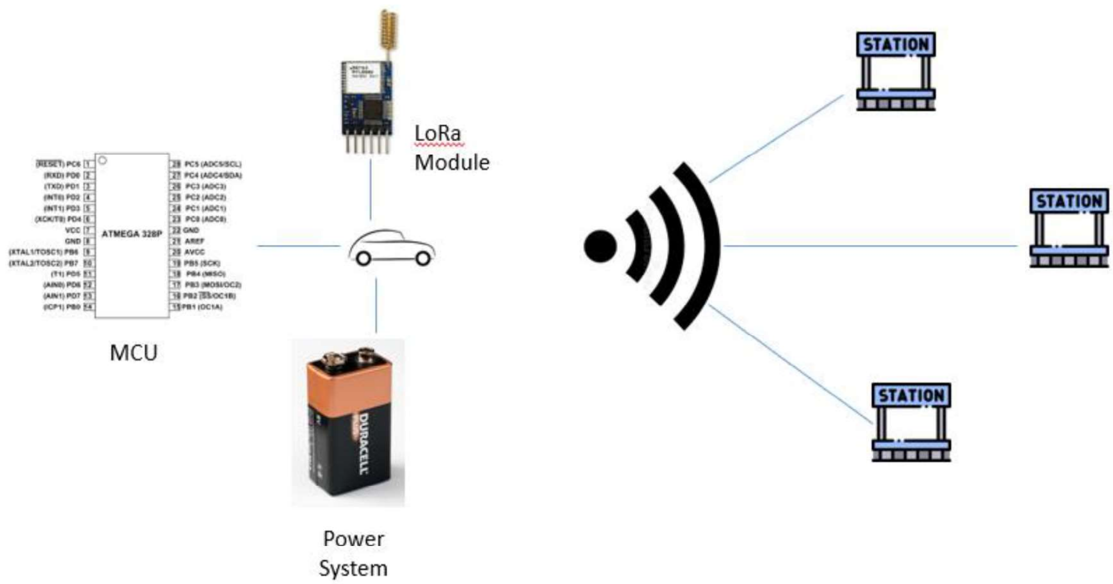


Figure 3.2. Moving vehicle system design

### 3.2.2. Station Gateway Design

Once a moving vehicle system collects signals from the vehicle, it needs to transmit them to the station gateways. The stations work as gateways that receive the signals from the moving vehicle and store their information on their time of arrival in a database to calculate the vehicle position. To design the station gateway, we need a microcontroller, a LoRa transceiver, and a power system, and a communication system to the database.

In this system, the Reyax LoRa module will be used as a transceiver to receive signals from the moving vehicle. One of the shortcomings of implementing the TDOA algorithm is the inaccuracy in capturing the right time of arrival of data packets from the transmitter hardware. According to Kelvin, for TDOA to accurately determine the position or location of a moving vehicle, the microcontroller must have a nanosecond precision [5]. However, even though it can be challenging to get this high accuracy we can consider some microcontrollers that have high performance and can improve the accuracy of the LoRa geolocation system.

Among the high-performance microcontrollers include Raspberry Pi. This single-board computer (SBC) consists of a flexible clock running up to 133MHz, consumes low power with a voltage requirement ranging between 1.8 to 3.3V, and contains the UART peripherals which are essential in connecting the LoRa module to the system. On the other hand, there is a NodeMCU microcontroller that operates at a clock frequency between 80 MHz and 160 MHz. NodeMCU is low power with an operating voltage of 3.3V and contains UART peripherals that can be used in connecting the LoRa module. The baseline considered is an Arduino Nano, a low-power microcontroller with the UART peripheral features that can facilitate the connection to the LoRa transceiver.

Below is a pugh chart indicating which of the two microcontrollers would be suitable for this project.

Table 3.3. Pugh chart for station gateway microcontroller selection

| **Selection criteria** | **Baseline** | | |
|---|---|---|---|
| | **Arduino Nano** | **NodeMCU** | **Raspberry Pi** |

| | | | |
|---|---|---|---|
| 1. Cost | 4 | 4 | 2 |
| 2. Availability | 4 | 4 | 4 |
| 3. Compatibility | 3 | 3 | 2 |
| 4. Low Complexity | 4 | 3 | 2 |
| 5. Low Power | 4 | 4 | 4 |
| 6. Peripheral Features | 4 | 4 | 4 |
| 7. Ability to store data on a database | 2 | 4 | 4 |
| Total | 25 | 30 | 26 |

From table 3.3. above, the NodeMCU is more suitable for this project. This is because NodeMCU has the highest clock frequency which is essential in synchronization. NodeMCU is also low-cost compared to Raspberry pi. NodeMCU costs around 2 USD while Raspberry is 4 USD. NodeMCU is also suitable due to its ability to connect to the database through its ESP8266 WiFI module.

Below is a diagram showing the main parts of the station gateway design. Hardware will be built consisting of the LoRa module, the NodeMCU, and a power system. This hardware will be installed at the stations. The role of this hardware will be to collect the signals from the moving vehicle and their time of arrival. This quantitative data will then be transferred to the database to proceed with vehicle position calculation.

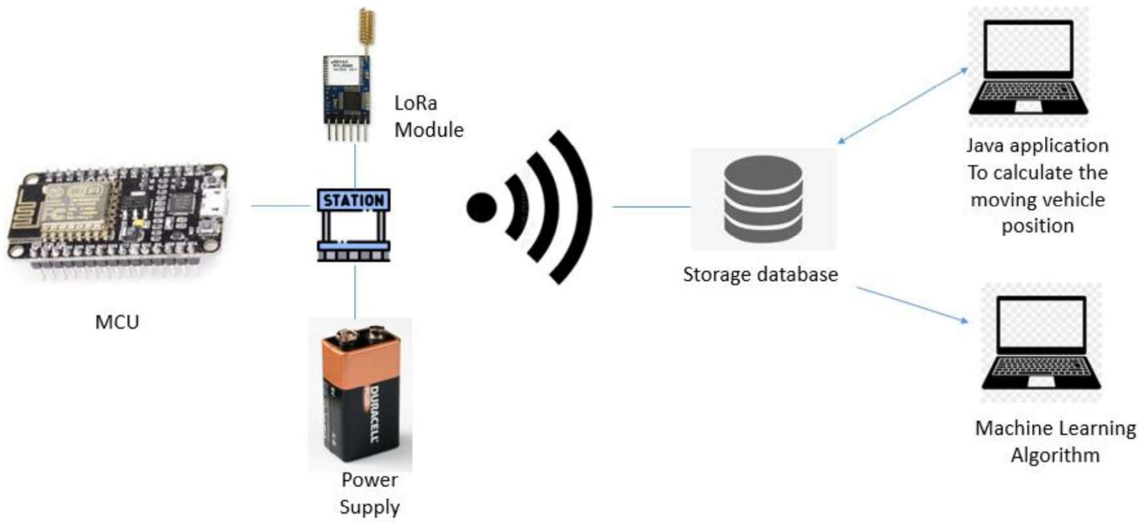


Figure 3.3. Station gateway system design

### 3.2.3. Power System

Power consumption is an important aspect in designing an alternative geolocation system. GPS GSM system consumes a lot of power and requires frequency battery change to keep it working. Therefore, in designing the LoRa vehicle tracking system, the power consumption must be relatively low compared to the GPS GSM system to ensure the system is low cost.

In powering the system, the interest is on a power system that is small in size so that it does not occupy a lot of space. The power system must be able to sustain system for a long period without requiring a battery change.

Based on the design decision above, the microcontroller Atmega 328p, Reyax LoRa module, and KL25Z freedom board are low-power components that can be powered by 5V. To extract this voltage to power the system, putting into consideration the size of the system, a 9V battery will be used. A voltage regulator will be applied to produce 5V that will power the design components.

Below is a schematic showing the voltage regulator circuit.

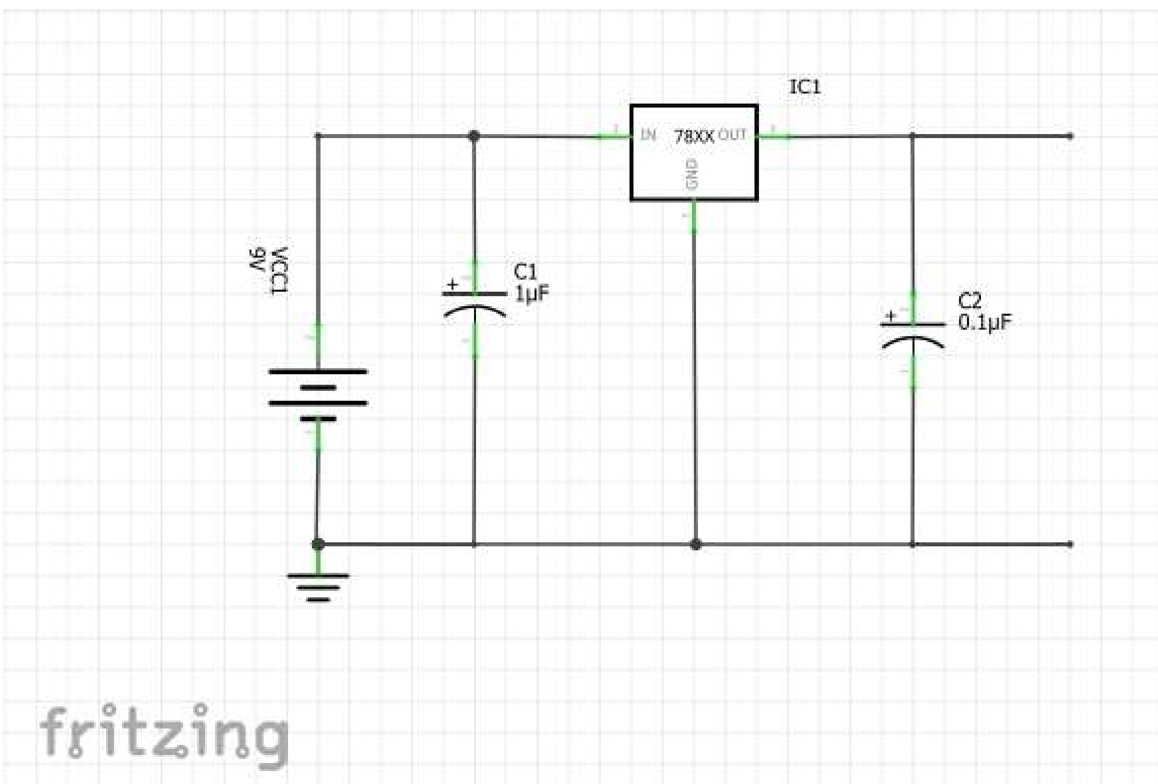


Figure 3.3. Voltage Regulator Circuit

According to Figure 3.3. above, the main electronic components required to produce power for the system are the 9V battery, two capacitors, and LM7805. The capacitor's role is to eliminate noise from the wires to produce a constant 5V. The LM7805 ensures that the 9V is stepped down to 5V. Based on the system requirements, this power system will ensure that the system consumes low power efficiency for vehicle tracking.

Reyax LoRa module requires a typical voltage of  3V. Therefore, a breadboard power supply can be used to ensure that the microcontrollers that require a 5V are satisfied as well as the LoRa module that requires a smaller voltage of 3 V. The breadboard power supply would be powered by a battery.

### 3.2.4.  Database Storage

Once both the moving vehicle or a mobile device system and a station gateway are set up, they send data packets and details on their time of arrival are stored in a database for quick and easy access for further analysis.

To choose a suitable database for this system, there are some criteria to follow. Based on the potential applications of this system such as criminal investigations and delivery tracking, the data needs to have security and a backup to avoid data loss, and it should be easy to scale up since we expect large amounts of data entered. The database should also be easily accessible by the machine learning algorithm for analysis. Based on the hardware design of the station gateway, the database must facilitate a way to write to it from the hardware design. According to these criteria, MySQL will be used due to its security layers to ensure data protection.

### 3.2.5. Machine Learning Algorithm

To improve the LoRa technology, machine learning is used to learn the system and make predictions in location determination for a moving vehicle or a mobile device. Based on the nature of the LoRa geolocation system and the location determination algorithm TDOA, the system generates quantitative data on the time of arrival of data packets from the transmission side to the station gateways.

GPS GSM geolocation system sets standards on the accuracy of location determination of a mobile device or a moving vehicle. Therefore, in training the machine learning algorithm, both GPS and LoRa datasets should be used to compare the behavior of the LoRa data to that of GPS. The LoRa dataset consists of output location position data given inputs of the mobile device. This is done through supervised learning and it allows the algorithm to generate pattern recognition to inform the prediction of the system.

The machine learning algorithm suitable to optimize the LoRa geolocation should have the ability to accurately predict the location of a moving vehicle or a mobile device once it is well trained.  The machine learning algorithm should also be able to maintain a certain level of accuracy with different data fed to it. With this, machine learning algorithms that meet the standards of the system requirements are Random Forest Algorithm and Decision Tree Algorithm.

Random Forest Algorithm is a machine learning which combines multiple subsets from a given dataset for training and derives a pattern to facilitate accurate prediction [13]. On the other hand, the decision tree algorithm is a machine learning algorithm that predicts the output given a certain input. The advantage of using the decision tree algorithm is that it takes into consideration all possible results before deciding.

Based on the criteria of a machine learning algorithm suitable to predict and optimize geolocation with LoRa, this project will explore the Random Forest Algorithm based on its accuracy in predicting values after using various datasets for training.

# Chapter 4: Implementation

## 4.1. Moving Vehicle Design Implementation

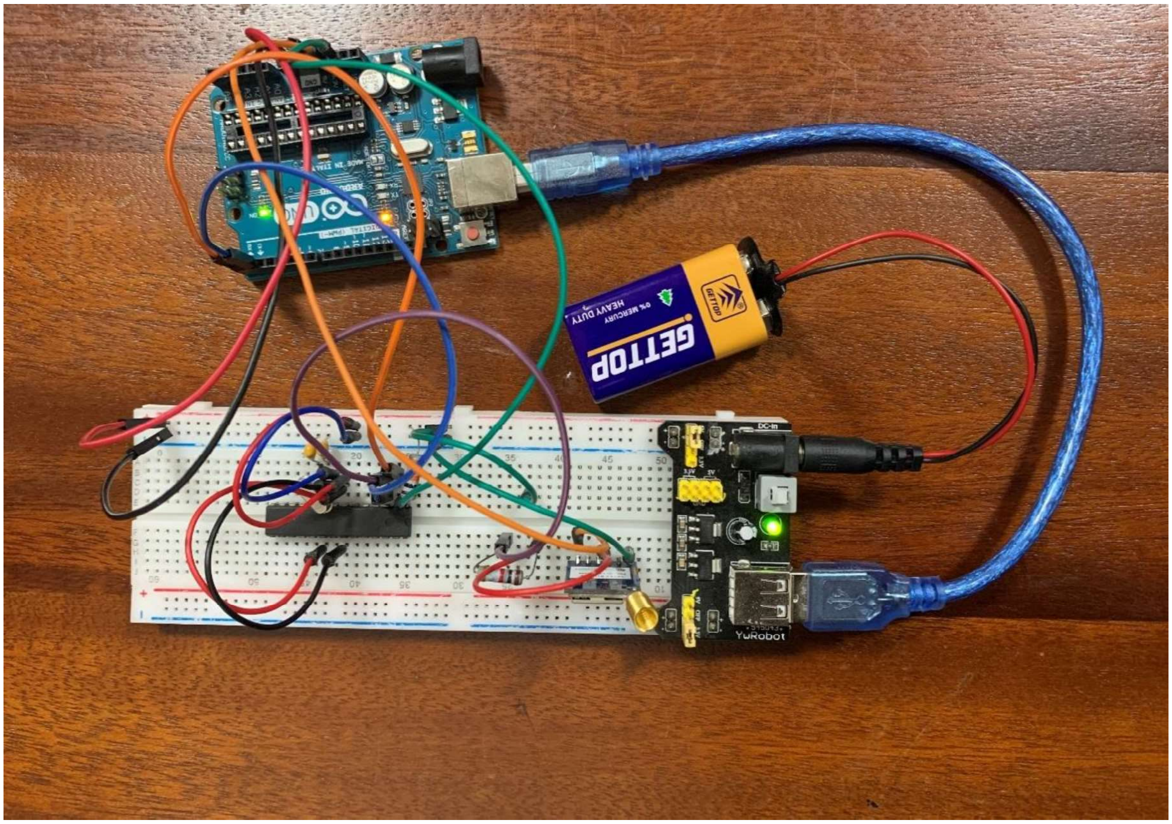


Figure 4.1. Transmitter Hardware Circuit

Based on the design decision from the previous chapter, Figure 4.1 above is a built circuit for transmission which is installed in a moving vehicle. The components used include:

- Reyax LoRa module as a transceiver.
- Atmega 328p chip as a microcontroller with a 16 MHz crystal clock to make the Atmega 328p chip run fast.
- Two 22pF capacitors.
- Jumper wires.

- Arduino UNO as an In-System Programmer (ISP) for the Atmega 328p.
- A power system made out of a 9V battery connected to a breadboard power supply. This breadboard power supply distributes 5V to power the Atmega 328p and 3V to power the Reyax LoRa module.

This hardware is programmed using an Arduino IDE. The Arduino IDE is then used to write the AT commands to the LoRa module instructing it to send data packets at an interval of 10s at a time. Data packets from the LoRa module transceiver are then sent to the receiver hardware circuit. Refer to Appendix for the transmitter circuit code.

## 4.2. Station Gateway Design Implementation

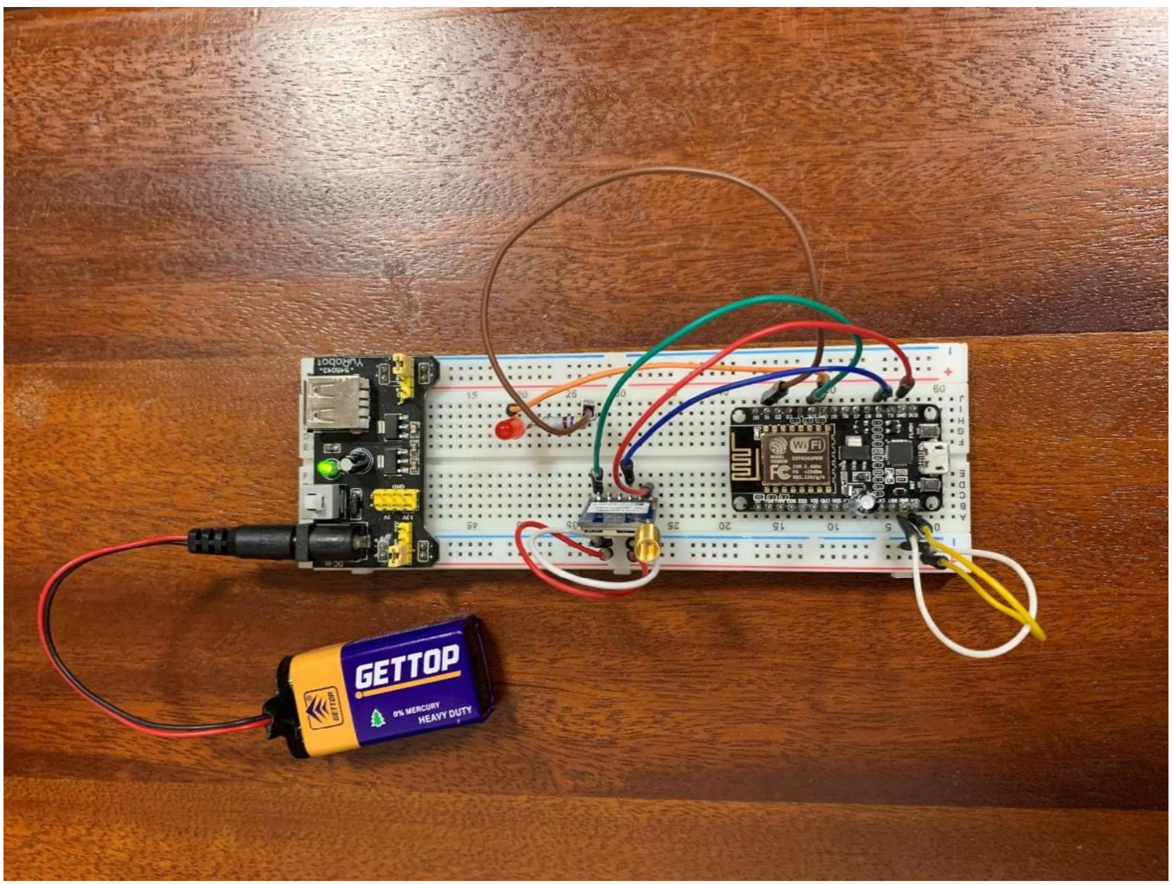


Figure 4.2. Receiver Hardware Circuit

The above circuit in Figure 4.2. illustrated the receiver hardware circuit. Based on the design decision in the previous chapter, the components used include:

- Reyax LoRa module transceiver as a receiving device
- NodeMCU microcontroller with an ESP 8266 WiFi module to communicate to the database.
- Jumper wires.
- Breadboard.
- A power system made out of a 9V battery connected to a breadboard power supply. This breadboard power supply distributes 5V to power the NodeMCU and 3V to power the Reyax LoRa module.

This circuit is programmed using an Arduino IDE. To interface Arduino with the NodeMCU, the ESP8266 library is used to facilitate the communication between the NodeMCU and the database. Arduino IDE also facilitates the programming of AT commands to the LoRa module to ensure that it receives data packets from the transmission hardware circuit. Refer to the Appendix for the receiver circuit code.

## 4.3. Database Implementation

The data to be collected on the moving vehicle can be confidential based on the area of application. To avoid any data leakage and data loss, a MySQL database is created to store this data and ensure its security.

In creating the MySQL database, phpMyAdmin was used. Based on the location determination requirements, two tables were created. One to store data on stations which in this case act as reference or gateways. The second table stores data on the vehicle including Timestamps on when data packets are received by the station gateways. These timestamps will then be used to calculate the vehicle position using the Time Difference of Arrival (TDOA) Algorithm.

Below is a sketch showing the relational database created based on the two tables mentioned above.

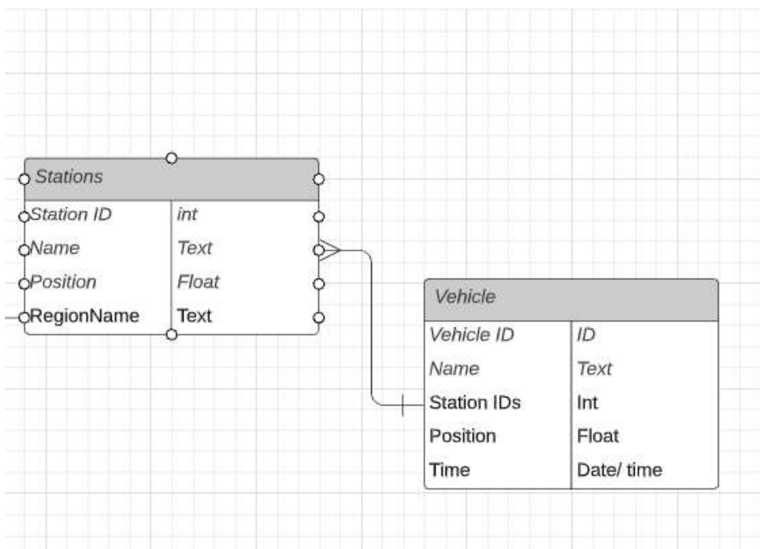


Figure 4.1. Sketch of the MySQL relational database

From Figure 4.1. above, there are two tables which will both contain information to be used in the location determination of a moving vehicle. The vehicle table requires an input of stationID from the stations' table hence creating a relationship between the two tables. Recording the stationID in the vehicle table ensures that we store information from at least three station gateways close to the moving vehicle to determine its location.

## 4.4. TDOA Algorithm Program

To implement the TDOA algorithm, an application program connects to the database and retrieves Time of Arrival (TOA) data from three station gateways in the proximity of the LoRa module at a time. The application program to calculate the location or position of the moving vehicle is written in Java language. Java is used because of its phpMyAdmin database connector which is relatively easy to implement. Using the TDOA Algorithm, the Java program will calculate the location of a moving vehicle and populate the position column in the vehicle table in the database. Below is pseudocode to indicate what the program does to determine the vehicle location.

*Connect to the database to access timestamps or time of Arrival inputs for the moving vehicle.*

*retrieve data for four different station IDs that a moving vehicle sends data signals to*

*Compute the distance difference using time of arrival and speed of light*

*Compute the x,y coordinates of a moving vehicle*

*Convert the x,y coordinates to latitude and longitude*

*Store the location coordinates in the database*

## 4.5. Machine Learning Algorithm Implementation

To implement a machine learning model, a dataset from the UCI Machine Learning repository was used. This dataset is called go_track_trackspoints, it contains 18107 data points, and it is based on GPS technology. This dataset records the position of a moving device in latitude and longitude and the timestamp of when this device is in a specific position. The nature of this dataset is similar to the expected dataset generated by the LoRa module in this system.

Using google colab, essential libraries pandas and NumPy were imported before training the model. These libraries are open-source python libraries for data analysis and manipulation. After importing the libraries, the dataset is uploaded to colab. Before proceeding, it is important to ensure that missing data in the dataset are taken care of. Using function info(), I checked for missing data in the dataset. The info function returned a list of all the columns in the dataset indicating that every column is non-null.

After checking that the dataset has no missing data, we proceeded to indicate the dependent and independent variables. In this project, we are trying to determine the position of a moving car based on the time of arrival of data signals to four station gateways. Therefore, the position of a moving vehicle i.e., latitude and longitude are the dependent variables, while the time and trackID variables are the independent variables. Once the independent and dependent variables, the dataset split the data into a training set and testing set. The training set was 75% of the dataset while 25% was the testing set. Once the dataset is split, I then moved on to train the model using the Random Forest algorithm from the sci-kit learn library for machine learning.

# Chapter 5: Test and Results

## 5.1. Hardware Testing

Based on the Multilateration method of Time Difference of the Arrival Algorithm, there should a minimum of four station gateways to improve the location determination. To test the setup of both the moving vehicle and the station gateway, four-station gateways were chosen around the Ashesi University Campus which is the testing ground. The station gateways chosen needed to be within the Lora range of 0-15 km to ensure that the LoRa transmitter can be able send data to the receiver side.

Below is the list of the station gateways on the Ashesi University Campus used in the hardware testing with their corresponding Google maps location to facilitate as a reference during the calculation of moving vehicle position.

Table 5.1. Table showing the chosen station gateways for testing.

| **StationID** | **StationName** | **Latitude** | **Longitude** | **X (km)** | **Y (km)** |
|---|---|---|---|---|---|
| 1 | Ashesi Engineering Block | 5.75995 | -0.21986 | 6338.779 | -24.324 |
| 2 | New Student hostels | 5.75828 | -0.22113 | 6338.805 | -24.464 |
| 3 | Fab Lab (Garage) | 5.75961 | -0.21824 | 6338.791 | -24.1446 |

| | | | | | |
|---|---|---|---|---|---|
| 4 | Big Ben Cafeteria | 5.75913 | -0.22146 | 6338.795 | -24.5009 |

The positions of station gateways are located in terms of latitude and longitude. However, in calculating the TDOA position we depend on 2D coordinates. Therefore, the latitude and longitude are converted to x,y coordinates. The corresponding x,y coordinates of the station gateways is indicated in Table 5.1. above. Below are formulas used in obtaining the 2D coordinates:

$$x = Radius\ of\ Earth * \cos(\ latitude) * \cos(longitude)$$

$$y = Radius\ of\ Earth * \cos(latitude) * \sin\ (longitude)$$

∴ Where the Radius of Earth is 6371km.

In testing the hardware, a clear line of sight was assumed and in absence of a real car to test with, a transmitter hardware circuit was located in the school's main car park and the receiver hardware circuit was installed at the four-station gateways stated in Table 5.1 above. Below is a google map picture indicating the position of the transmitter site and receiver sites.

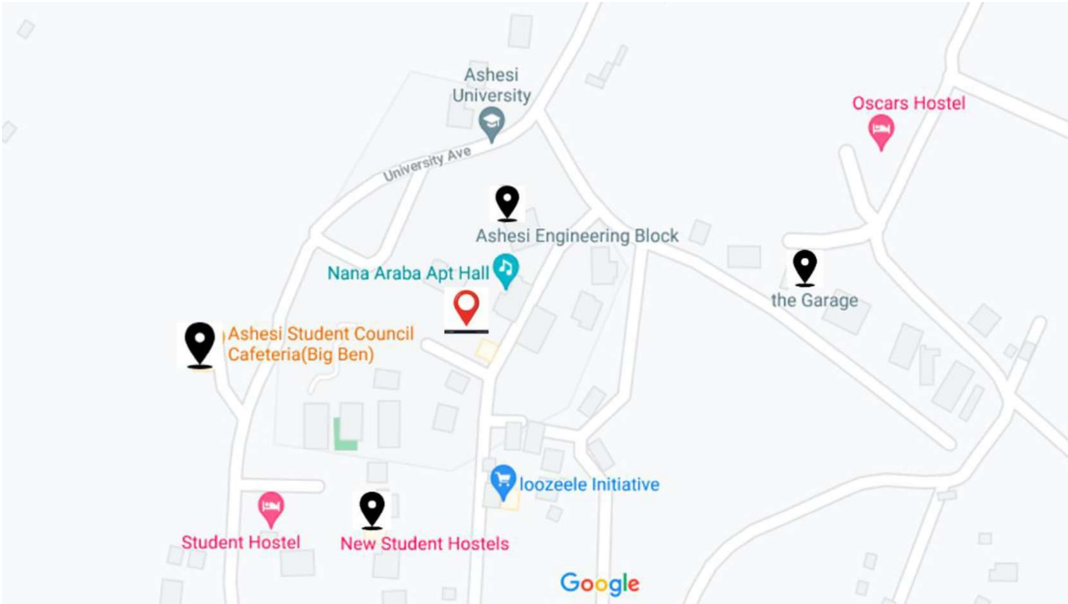


Figure 5.1. Diagram showing positioning of the station gateways and transmitter.

Figure 5.1. above shows the positioning of the four receiver hardware circuits for the station gateways and transmitter at the Ashesi University Campus. The locations with a black location pointer are the four-station gateways that receive data from the transmitter while the red location pointer determines where the transmitter is installed. Once both hardware setups are powered, the LoRa transceiver at the transmitter site sent data signals to the LoRa transceiver at the station gateways. The station gateways recorded timestamps of when the data is received. These timestamps are then used in computing transmitter location.

During testing, one receiver circuit and one transmitter circuit were used. This is because there were no three more LoRa modules to set up for the four-receiver station gateways at the same time. This, therefore, implies that there was no simultaneous sending of data signals to all the station gateways at the same time. For data collection purposes, a reference time was used to calibrate the data from station gateways as though data was sent to all four of them at the same time.

From the station gateway positioning above, it is clear that the testing ground consisted of obstacles such as buildings which made it challenging to find a clear line of sight. Another testing was done at the Ashesi football pitch for a clear line of sight. Four random positions were chosen within the LoRa distance range. Since there is a weak internet connection in this space, a computer was used to verify the sending of data signals using a serial monitor on the Arduino IDE. Even though there was a clear line of sight in this space, it was challenging to get timestamps even in very short distances. This test was done when there was a lot of wind, and this potentially distorted the Reyax LoRa module antennas preventing them to send data signals as expected.

## 5.2. TDOA Approach

TDOA algorithm does not require synchronization between the transmitter and receiver. However, to achieve an accurate TDOA reading it requires synchronization between station gateways to ensure that data is received at the same time. In the past research done in this field, receiver synchronization is mostly assumed. In real-time testing, huge country-wide LoRa Networks such as the KPN network in the Netherlands, and Sagemcom are used during testing to achieve the desired accuracy. These networks use a multilateration algorithm that requires four or more gateways to be installed and they can record timestamps up to nine decimals. For instance, Sagemcom used 42 gateways to achieve an accuracy of 4 meters [9]. To get this accuracy, it required Sagemcom a below timing:

$$Time = \frac{Speed}{Distance}$$

$$Time = \frac{\frac{(3 * 10^8)m}{s}}{4m} = 0.75 * 10^{-8} seconds$$

Therefore, we must ensure a nanoseconds timing to ensure a good accuracy as that of Sagemcom.

In this project, the microcontroller used at the station gateways recorded the timestamps up to 3 decimal points. Below is sample raw data retrieved during testing from the Arduino IDE serial monitor.

Table 5.2. Sample data received at the station gateway Ashesi Engineering Block during testing

| **Dummy data printed to the Serial Monitor** | **Timestamps** |
|---|---|
| Data Received | 17:47:42.911 |
| Data Received | 17:47:54.019 |
| Data Received | 17:48:05.119 |
| Data Received | 17:48:16.219 |
| Data Received | 17:48:27.317 |
| Data Received | 17:48:38.431 |

Table 5.2. above shows sample data at a station gateway Ashesi Engineering Block during testing. The timestamps indicate hour: minutes: seconds. The microcontroller stored timestamps with milliseconds precision and this does not facilitate an accurate TDOA localization.

## 5.3. Machine Learning Algorithm

During the implementation of the machine learning algorithm, a dataset from UCI Machine Learning Repository was used. This yielded a 70.8% score. This, however, consisted of data generated by GPS. Given that GPS is a standard technology in determining location or position, this does not represent what the LoRa dataset would produce.

Ideally, a LoRa dataset generated from the database should be used during training. This would allow us to compare the training score of the LoRa dataset as compared to that of GPS and hence determine the effect of using LoRa in location determination systems.

# Chapter 6: Discussion and Conclusion

## 6.1. Conclusion

This project studied geolocation with LoRa and designed a tracking system intending to optimize the accuracy of the TDOA Algorithm using machine learning and timing. This project came up with a low-cost, low-power design for hardware. However, this design had a millisecond precision which is not suitable to ensure accurate TDOA algorithm localization.

On the other hand, to train a machine learning model, a more accurate dataset generated by GPS was used. Ideally, a LoRa dataset would be used to ensure the evaluation of the machine learning effect on the system. However, since there were challenges such as a millisecond precision timing system, limited LoRa modules for simultaneous data transmission to station gateways, we did not rely on the locations generated by the system for the LoRa geolocation system.

## 6.2. Limitations

As stated in the subsection above, implementation of this project faced some limitations which hindered the completion of this project and the improvement of the LoRa geolocation system. Geolocation with LoRa and machine learning has a lot of room for improvement. Below are some limitations faced that could be tackled in the subsequent projects in this field.

In testing TDOA, a timer with a nanosecond's precision is ideal in ensuring accurate TDOA inputs. The microcontroller used in this project gave a millisecond precision. This implies that TDOA inputs were not as accurate in capturing the slightest time changes in time of arrival to ensure accurate localization.

The failure to get an accurate TDOA position affected the training of the machine learning model. Initially, the results from TDOA positioning of the moving vehicle would contribute to the dataset used in training the machine learning model. However, this project failed to get an accurate LoRa geolocation position from real-world testing, hence affecting the implementation of the machine learning model.

To ensure a clear line of sight, the LoRa geolocation system was tested at the Ashesi University football pitch. However, having a clear line of sight did not guarantee the transmission of data signals. The openness in this testing ground allowed a heavy wind on the LoRa modules used and that interfered with the transmission of data signals even in very short distances.

## 6.2. Future Work

To fulfill the requirements of this system in designing an efficient low cost and low power geolocation system using LoRa, below are potential future work:

- To ensure accurate testing of the multilateration algorithm, four-station gateways will be built to ensure a simultaneous transmission of data signals to all four station gateways.
- Statistical analysis will be carried out to evaluate the effect of multilateration algorithms and machine learning on the LoRa vehicle tracking system. Given the limitations faced during the implementation of this project, a statistical analysis was not carried out. However, it would be relevant for future research to ensure that this is done to evaluate the impact.
- The station gateways design complies with a NodeMCU which allows TCP/IP communication between the microcontroller and the LoRa module. The NodeMCU plays a big

role to ensure that the time of arrival of signals is stored in a database to be used to calculate location by a program. To do that, it requires connection to WiFi to be able to store that data online. As part of future work, a protective case will be designed to contain the hardware as well as a router that the NodeMCU can connect to. This is important because this can be easily portable and installed in various places to create a station gateway and it can also prevent obstacles such as heavy wind from affecting the LoRa modules antennas and interfering with data transmission.

- Time synchronization is essential in achieving accurate positioning using the TDOA algorithm in a LoRa geolocation system. More research will be done to design an affordable means of implementing this and improve geolocation in LoRa using the TDOA algorithm.

# References


[1] B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANs," *2017 Global Internet of Things Summit (GIoTS)*, Geneva, 2017, pp. 1-6, DOI: 10.1109/GIOTS.2017.8016251.

[2] F. Carrino, A. Janka, O. Abou Khaled, and E. Mugellini, "LoRaLoc: Machine Learning-Based Fingerprinting for Outdoor Geolocation using LoRa," *2019 6th Swiss Conference on Data Science (SDS)*, Bern, Switzerland, 2019, pp. 82-86, DOI: 10.1109/SDS.2019.000-2.

[3] A. Lavric and V. Popa, "Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey," *2017 International Symposium on Signals, Circuits, and Systems (ISSCS)*, Iasi, 2017, pp. 1-5, DOI: 10.1109/ISSCS.2017.8034915.

[4] F. Zafari, A. Gkelias and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568-2599, third quarter 2019, DOI: 10.1109/COMST.2019.2911558.

[5] A. Kelvin, "Location Determination Algorithms", 2020 Ashesi University Engineering Capstone Project, 2020, pp.29-34.

[6] D. Bissett, "Analysing TDOA Localisation in LoRa Networks," Delft University of Technology Master's Thesis in Embedded Systems, 2018, Retrieved from https://repository.tudelft.nl/islandora/object/uuid:bea423b1-6f04-4708-8ed4-e8663dd51cde

[7] J. Cho, D. Hwang, and K. Kim, "Improving TDoA Based Positioning Accuracy Using Machine Learning in a LoRaWan Environment," 2019 International Conference on Information

Networking (ICOIN), Kuala Lumpur, Malaysia, 2019, pp. 469-472, DOI: 10.1109/ICOIN.2019.8718160

[8] J.Silva, J. Rodrigues, A.M. Alberti, et al, " LoRaWAN – A Low Power WAN Protocol for Internet of Things: a Review and Opportunities", 2017 Research Gate, Retrieved from researchgate.net/publication/318866065

[9] B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANs," 2017 Global Internet of Things Summit (GIoTS), Geneva, 2017, pp. 1-6, DOI: 10.1109/GIOTS.2017.8016251.

[10] F. Carrino, A. Janka, O. Abou Khaled, and E. Mugellini, "LoRaLoc: Machine Learning Based Fingerprinting for Outdoor Geolocation using LoRa," *2019 6th Swiss Conference on Data Science (SDS)*, Bern, Switzerland, 2019, pp. 82-86, DOI: 10.1109/SDS.2019.000-2

[11] B. Choi, K. La and S. Lee, "UWB TDOA/TOA measurement system with wireless time synchronization and simultaneous tag and anchor positioning," *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Ottawa, ON, Canada, 2018, pp. 1-6, DOI: 10.1109/CIVEMSA.2018.8439949.

[12] S. Kim, J. Chong, " An Efficient TDOA-Based Localization Algorithm without Synchronization between Base Stations," *International Journal of Distributed Sensor Networks*,2015, DOI: 10.1155/2015/832351

[13] S.Uddin, A. Khan, M. Hossain, *et al.* Comparing different supervised machine learning algorithms for disease prediction. *BMC Med Inform Decis Mak* **19,** 281 (2019). https://doi.org/10.1186/s12911-019-1004-8

# Appendix

The code below is for the transmitter circuit.

```
/*Transmitter Circuit
 *
 *
 * This code transmits sample data to the receiver circuit using AT commands.
 * The sample data is "DataIsTransmitted"
 * Once received, the receiver circuit checks if it gets this data packet
 * To ensure that it is receiving from the right transmitter
*/

String data;
unsigned long lastTransmission;
const int interval= 10000;

void setup() {
  Serial.begin(115200);

}

void loop() {
  // put your main code here, to run repeatedly:

 if(millis() > lastTransmission + interval ){
     data = Serial.readString();
     Serial.println("AT+SEND=0~65535,17,DataIsTransmitted");
     lastTransmission = millis();
   }
```

```
}
```

The code below is for the receiver circuit. It receives data packets from the transmitter and sends information about the vehicle onto the database to be used for analysis and TDOA calculations.

```
/*Receiver circuit
 *
 * This code receives the data packet from the transmitter circuit and populates information about a transmitter onto
 * a MySQL database to record time of arrival that is used to calculate TDOA location.
*/

// libraries
#include <ESP8266WiFi.h>
#include <WiFiClient.h>



//hotspot details

const char* ssid = "cap_wifi";
const char* password = "wnaif0my4a35f";

char server []= "localhost"; // or the PC's local ip

WiFiClient client;

// vehicle info

int VehicleID=1;
String vehicle_Name ="vehicle1";
```

```
#define led D1
String receivedData;
int StationID=1;

void setup() {
  // put your setup code here, to run once:

  Serial.begin (115200);
  pinMode(led, OUTPUT);
  delay(10);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  Serial.println("Server started");

  // Print the IP address
```

```
  Serial.println(WiFi.localIP());
  delay(1000);
  Serial.println("Connecting...");

}

void loop() {
   //put your main code here, to run repeatedly:
  if (Serial.available()){
      receivedData= Serial.readString();
      if (receivedData.indexOf("DataIsTransmitted")>0){
          Serial.print("Data Received \t");
          Serial.println(stationID);
          digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
          delay(1000);                       // wait for a second
          digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
          delay(1000);                       // wait for a second
          send_to_database();   //Send data to database
      }
    }
}

void send_to_database(){

  if (client.connect(server, 80)) {
      Serial.println("connected");
      // Make a HTTP request:
      client.println("GET
/LoraData_Capstone/insertVehicle.php?insert&vehicleID=VehicleID&VehicleName=vehicle_Name&Sta
tionID=StationID HTTP/1.1");
      client.println("Host: localhost");
```

```
      client.println("Connection: close");
      client.println();
    } else {
      // if you didn't get a connection to the server:
      Serial.println("connection failed");
    }
}
```