*LoRa Proximity Model for Indoor Space*

B.S Electronics Engineering

Norfolk State University

2020 - 2021

Project Mentor Submitted by

Dr. Hongzhi Guo Mario Alcaraz

Bruna Goncalves

Submission Date

11/17/2020

Declaration

We hereby certify that the project entitled “LoRa Proximity Model for Indoor Space” by Mario Alcaraz (0436290) and Bruna Goncalves (0446033), in partial fulfillment of requirements for the award of degree of B.S. (Electrical and Electronics Engineering Engineering) submitted in the Department of Engineering at NORFOLK STATE UNIVERSITY, is an authentic record of our own work carried out under the supervision of Dr. Hongzhi Guo. This material has not been submitted by me in any other University for the award of B.S. Degree.

Mario Alcaraz Bruna Goncalves

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Hongzhi Guo,

Mentor’s Designation

Acknowledgement

First, we are grateful for Norfolk State University for creating such a wonderful environment not only to learn but to develop soft and hard skills. As well, we would like to highlight the extraordinary job that the Engineering department has done guiding us through the development of our engineering careers. We would like to thank Dr. Hongzhi Guo, our professor and mentor. His guidance, extraordinary advice and encouragement are big contributions for the development of this project. We also want to thank Dr. Prathap Basappa for his advice and guidance on the first steps of the project. Additionally, we would like to thank Dr. Patricia Mead, Dr. Michael Kozhevnikov and Dr. Hargsoon Yoon for providing important feedback and advice throughout the process which encouraged us to find solutions and a better approach. Last but not least, we would like to thank our friends and classmates for the whole support they gave to us and help to many problems we had in the way.

Executive Summary

In recent years, LoRa technology has been developing rapidly. By the hand, the necessity to understand its performance and applications. The “LoRa Proximity Model for Indoor Space” is a project intended to develop a proximity system based on emergent technology LoRa (Long Range). The system will be able to proportionate the approximate distance between a client (end node) from its server (gateway) in a one-dimensional plane. Data analysis is performed to develop a path loss model prediction. Once again, experiments and data collection are kept on determining accuracy and make adjustments to the path loss model. The result leads to an automated system able to track the proximity of the client in a simple program which prepares to operate at user level. The system can be developed more than a one-dimension plane by implementing two more gateways at different points so it will collect more data and be able to generate a more precise system in a multi-floor environment.

Introduction

LoRa (short for long range) is a spread spectrum modulation technique derived from spread spectrum (CSS) technology developed by Semtech. LoRa devices and the open LoRaWAN protocol enables IoT applications that solve the biggest current challenges in the world, such as energy management, pollution control, etc. LoRaWANs are designed to fill the gap between short-range, high-bandwidth networks and cellular networks. It communicates through low power (928- MHz in the US) and enables long range transmissions up to 20 km (in open rural environments) and up to 2-3 km in urban environments with an outdoor gateway[1][2]. This opens up a new world of possibilities of implementations to solve problems in rural and urban areas. Therefore, we decided to give our project a focus on the urban implementation side. When a transmitter and a receiver (client/server) communicate, they do it through electromagnetic waves at a certain frequency. Although not all the time the signal sent is the signal received. When this happens, there is a determined loss of the signal that depends on multiple factors such as the connectivity available, the space between client and server, objects that may interfere in between, etc. The goal of our project is to measure this signal losses (Db power loss in our system) and through multiple data collection and analysis, find the relationship between distance and path loss of the system for a determined indoor space of one dimension. With the implementation of this model and a sensor for a determined goal, it would be possible to tell the approximate distance from the gateway from which the sensor signal was sent. This could be used to track human activity in a room, detect fire and have information about its approximate location, etc. Meaning this that decisions can be taken faster and smarter when needed or for an emergency. This allows a better and more efficient management of the implemented industry.

Literature and Review

Significant understanding of LoRa technology was required for the development of the project. LoRa (Long Range Communication) was released 10 years ago to the market. Since its invention, the development applications of it has accelerated due to the global adoption of IoT (Internet of things). LoRa is a method for transmitting radio signals that uses a chirped, multi-symbol format to encode information. It is a proprietary system made by chip manufacturer Semtech; its LoRa IP is also licensed to other chip manufacturers. These chips are standard ISM band radio chips that can use LoRa (or other modulation types like FSK) to convert radio frequency to bits, without any need to write code to implement the radio system. LoRa is a lower-level physical layer technology that can be used in all sorts of applications outside of wide area [17]. The biggest goal of this technology is to add wireless communication to track data from things in a long range and use IoT for gas, water and electricity meters, etc. Therefore, an application for CSS (Chirp Spread Spectrum), modulation technology, has been developed. These chips are used for sonar in the maritime industry and radar in aviation. The finality is to send data between gateway (Server) and end-node (Client).

After understanding the characteristics of LoRaWAN communication, we applied this characteristic in our project to track the distance between Server and Client. Due to data collection and dBm loss signal receive in the monitor we were able to proceed this project. Where dBm is used as a reference of how well from the Client can hear from the Server.

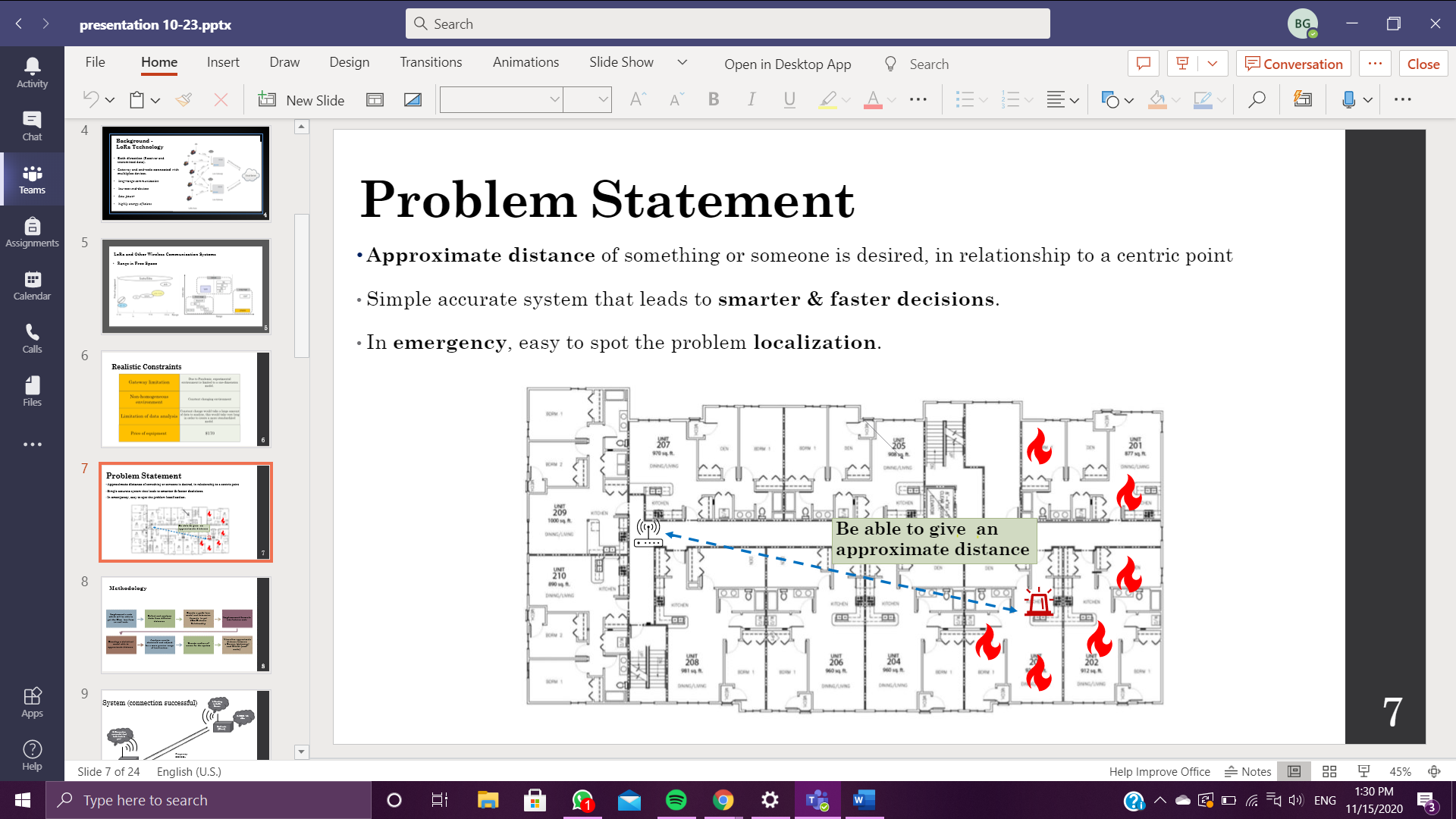
Additionally, documentation on published papers on its current applications and behaviors was performed [1][7][11][12][13]. This enabled us to have a better view of the capabilities of LoRa and its uses. It also provided us with information on possible constraints presented and analyze the processes followed by prior researchers. Then, we were be able to trace our steps and gave us idea how to implement LoRa and the necessary implementation to get our goal in this research.

The information acquired from the resources mentioned was complemented with our mentor’s help to have an understanding of it in a simple way their paper. Our mentor helped us with any problem it came during the process and discussed the best approach. Experimentation with Dragino LoRa kit was performed from the beginning of the project to get familiar with it and have hands-on experience be able to start understanding the unique material we have in our hand.

Objective

The goal of the project is to get the approximate distance in a one-dimension space indoor environment using LoRa wireless communication. Also, the model should provide the final results in a simple user-level way. Therefore, no engineering background should be required to operate the model and understand the results

Knowledge in wireless communication, Arduino software, microcontrollers and mathematics was required for the desired outcome. The model can be used to implement smart systems that provide important information to the user. Aa an example, a fire situation. The approximate localization of the fire can be tracked easily.

Figure Ⅰ. Able to visualize our goal 

Project Management and Workplan

A timeline for the development of the project was created in order to have a schedule to follow and guarantee progress as well as the realization of the project on time (Table Ⅰ).

TABLE Ⅰ: TIMELINE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Date of Completion** | **Material** | **Place** | **Problem Responsible** |
| **Get documented on**  **LoRa technology, Dragino kit and RF signals** | Feb 15 | Internet and Draigno kit | School | Mario Alcaraz  Bruna Goncalves |
| **Test Dragino kit to set**  **up a simple connection between end node and gateway** | March 6 | Dragino kit | School | Mario Alcaraz |
| **(Due to pandemic we could not start running Client/Server network) get documented on Research papers to analyze the possible outcomes, difficulties and challenges when setting a network for indoor environment** | March 13 - May | internet | Home | Mario Alcaraz  Bruna Goncalves |
| **Set up a Server - Client network with Dragino kit and get the software needed to complete its task as well as proving a successful network establishment.** | Sep 7 - Sept 11 | Dragino IoT network kit, Arduino software, 2 laptops, 2 Ethernet wires. | Residential Building | Bruna Goncalves |
| **Run network tests and implement.** | Sep 12 - Sep 19 | Dragino IoT network kit, Arduino software, 2 laptops, 2 Ethernet wires. | Residential Building | Mario Alcaraz  Bruna Goncalves |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Date of completion** | **Material** | **Place** | **Problem Responsible** |
| **Organize a data collection map to identify different areas tested and develop C code that automatize data collection in txt file.** | Sep 20 - Sep 27 | Dragino IoT network kit, Arduino software, C software, computer. | Residential Building | Mario Alcaraz |
| **Analyze the data collected from the different points to get a model for estimated distance from gateway. Test model accuracy** | Sep 28- Oct 18 | Txt files with data, C software and Arduino | Residential Building and school. | Bruna Goncalves |
| **Run another network experiment to collect data. Mix data and create a new path loss model and check the accuracy** | Oct 19 - Oct 23 | Txt files with data, C software and Arduino | Residential Building and school. | Mario Alcaraz  Bruna Goncalves |
| **Work on the accuracy of the model** | Oct 24 - Oct 30 | C software and Arduino | Residential Building & school. | Mario Alcaraz  Bruna Goncalves |
| **Develop and implement the path loss model prediction into C++ code** | Nov 1 - 6 | C++ Software and Arduino | Residential Building | Mario Alcaraz |
| **Test the code with the LoRa network together and work on last touches to make it as simple as possible** | Nov 7-10 | C++ Software and  Arduino and Dragino kit | Residential Building | Mario Alcaraz  Bruna Goncalves |
| **Work on final presentation and report** | Nov 11 - 17 | Laptop and results obtained | Residential Building | Mario Alcaraz  Bruna Goncalves |

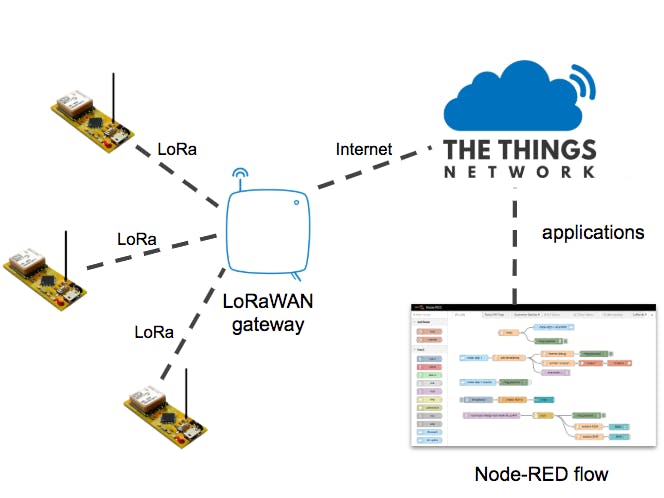
One of the biggest challenges faced was the inability to have physical interaction due to pandemic. To solve this problem, multiple meetings were to be carried on via Zoom. Once a week with mentor Dr. Hongzhi Guo and a minimum of 3 times a week between project members. Also, use of Microsoft Teams allowed cooperation for the development of presentation, analysis and research. Lastly, an in-person meeting was carried on once a week for experimental purposes, extra data collection needed and Path Loss Model testing. The work was distributed according to each member’s strength for specific short tasks and collaboration for longer tasks.

Design and Evaluation

The intended specification for the project is to provide a model for an indoor environment that allows a user to visualize in a simple way the proximity of a Client in relationship to its Server. The model must be simple and easy to understand for any user that does not have an engineering background. The approach to complete the project was divided in three general steps: data collection and environment experiment, analysis and model development, and implementation of model into real system and error analysis.

Data collection and analysis

A proper setup of the Dragino network was the first step on the project. The goal was to set up a Server/Client network between the gateway and end-node. As well as ensure that a two-way communication happens (Fig. ⅠⅠ). The way to do it was by setting up both at the same frequency (868 MHz) via Arduino Software.

FIGURE ⅠⅠ [3]. LoRaWAN connection.

The next step to take was to implement a code to the Arduino software that would display a “successful connection” message and print the Received Strength Indication (RSSI) by the Client (end-node) in dBm. Then, we could test the system by moving around and observe the behavior of the RSSI received as we move close to or separate from the gateway [eq.Ⅰ][eq.ⅠⅠ]. An automated system that provides data every determined time was created

(Fig.ⅠⅠⅠ).

(eq.Ⅰ) (eq.ⅠⅠ)

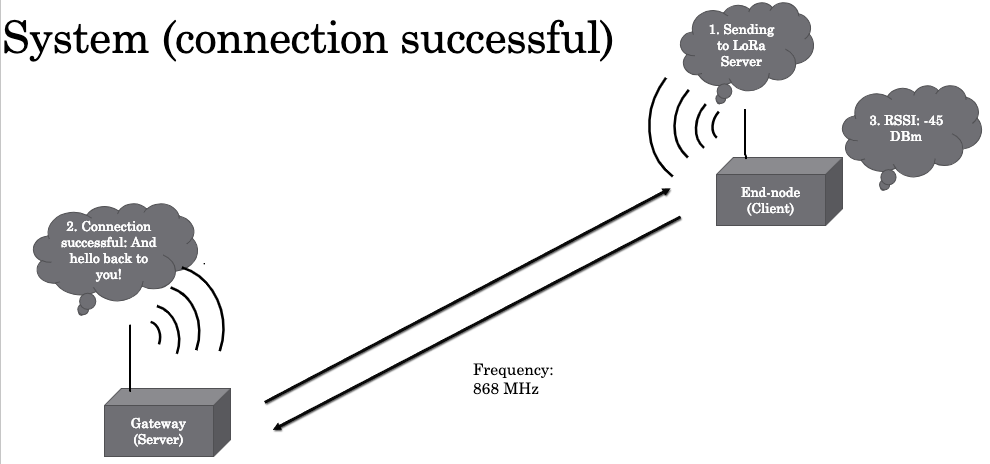


FIGURE ⅠⅠⅠ. Flowchart in a connection successful

The last step of the first segment of the project was to utilize the system and collect data from different distances from the Server. The end-node was placed at a measured distance in each 2 meters (2 meters, 4 meters, 6 meters, etc.) where we would receive the RSSI for that point. The data from each point was stored in a txt file for future analysis to develop our first Path Loss Model. The Software CoolTerm [4] was used to collect the data. This allowed us to automatically save the data at the same moment it was provided. The experimental environment as well as the points used for data collection are presented (Fig.ⅠV).

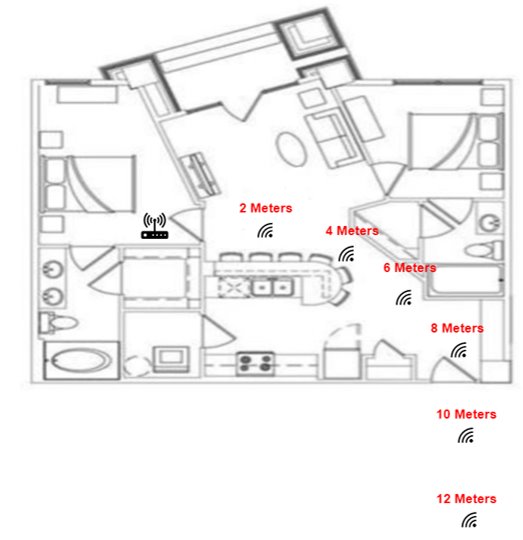


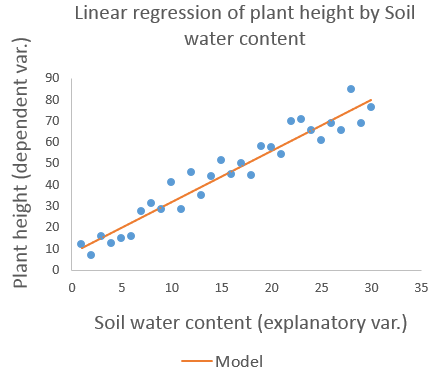
FIGURE ⅠV. Experiment Environment

Analysis and Model Development

The next step in the project was to analyze the data obtained by plotting it in a graph for a better visualization of the system's behavior. The data was divided among the different distances with the respective data collected for each of them. Due to environmental degradation we added procedures to give us more accuracy. First, we took a 10 point data set and removed the high and low outliers, and then averaged the remaining points to get a single point, and we repeated this procedure 10 times. After we got the full 10 points set, we also took out the high and low set with the 8 remaining points and plotted it in a scatter graph.

Then, we performed linear regression to the graph to obtain an average of the data of each point. This would allow us to get an equation for a “Path Loss Model Prediction” that would serve as our statistical model (Graph Ⅰ) [5]. Then, a predictive function would be obtained (eq.ⅠⅠⅠ).

GRAPH Ⅰ

STATISTICAL MODEL EXAMPLE

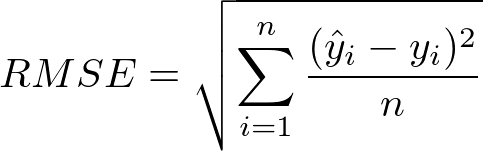
(Eq.ⅠⅠⅠ)

Model Implementation

The last step is the implementation of the model created, via Arduino. The approach for this step was to develop a new function inside our current Arduino code. This function would require only one parameter. Then, our current code would pass the RSSI dBm received to the function as the required argument. The function would use that value to substitute for ‘y’ in the predictive function previously generated (Path Loss Predictive Model) . Lastly, the function would return the predicted distance and print the value for the user in a simple message such as “The device is 10 meters away”. This would be the final model to be tested for our project. Therefore, the next step was to test the implementation of the model in a real environment to check its accuracy.

Error Analysis

Analysis of error is crucial in any engineering project. Therefore, testing the results obtained to check for its accuracy is an important step. In order to do it, we were to test it in the same experimental environment and compare the output received from the system with the measured distances. Once the testing was done, we developed a method that would include the measured distance against the calculated distance predicted by the model created. To define the error of the model, we utilized a normalized mean square root error (NRMSE)(eq. ⅠV)[6] approach. NRMSE provides a normalized RMSE which is a common technique used in regression analysis to verify experimental results with different scales. This provides us a view of how accurate the model is.

 (Eq. ⅠV)

Relevance

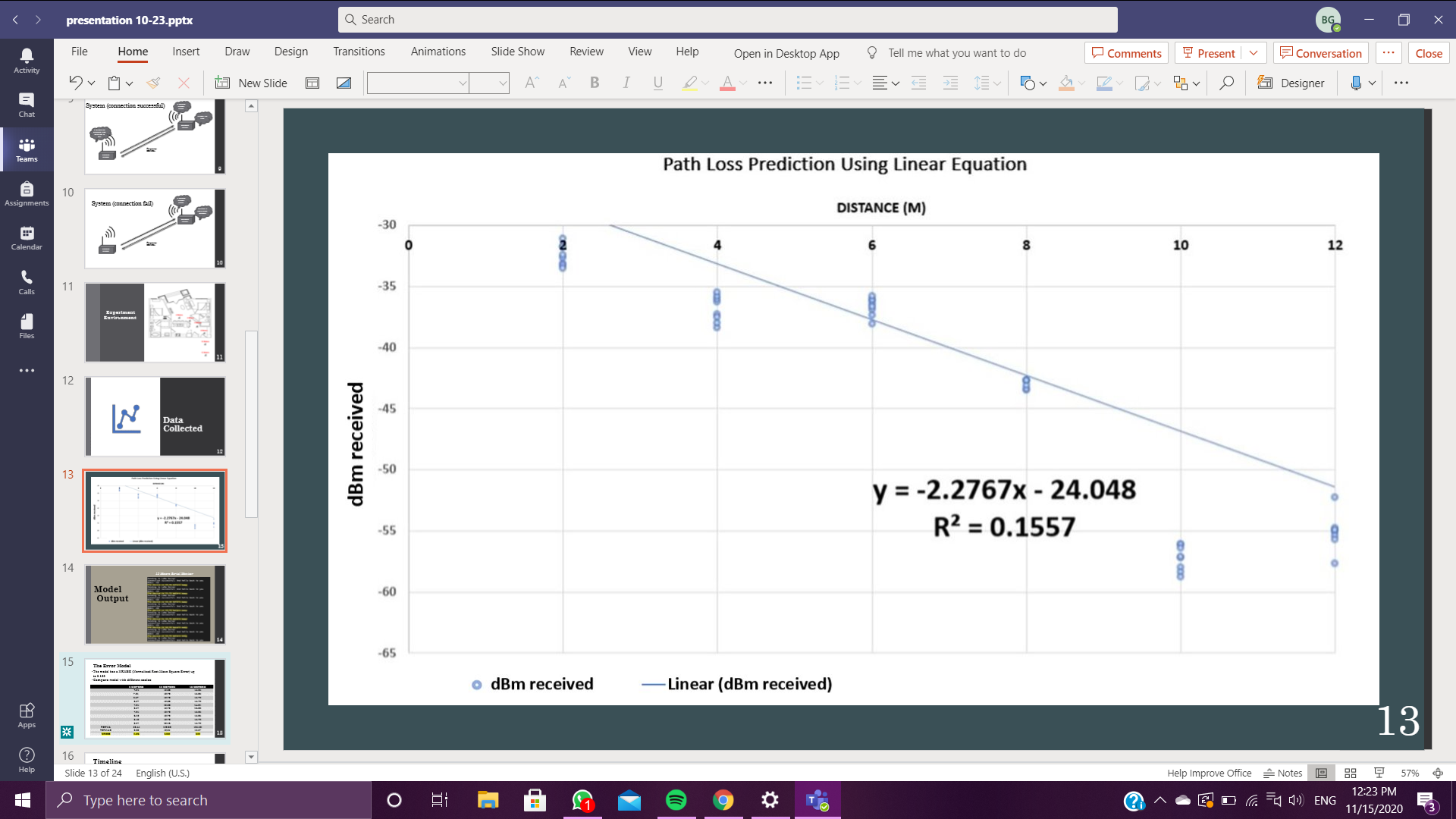
As Norfolk State University engineering students, deep understanding of engineering concepts is expected. This project provided us with the opportunity to have hands-on experience in a real engineering project that could be used for real applications. Application of knowledge is an important step for a better understanding and self-development. The project we worked on required knowledge on electromagnetic waves for communication, understanding LoRa recent technology, coding skills on C++, statistics, data analysis, ability to perform proper research, time management, microcontrollers knowledge and more. Therefore, there is high relevance to the formal NSU Engineering curriculum. Applying the knowledge acquired these years at NSU into a final project, shows proper understanding of engineering and ability to use them. On the other hand, it also represents one, if not the first, project that will add to our engineering portfolio.

Results and Discussion

The Path Loss Prediction Model (Graph. ⅠⅠ), is the result from plotting the data collected into a scatter graph and performing linear regression to create a predictive equation (eq. V) for the distance based on the RSSI dBm received. Where ‘y’ represents the dBm received and ‘x’ the distance. Therefore, we solved for ‘x’ to determine the equation (eq.VⅠ) to be implemented into the code for distance tracking.

GRAPH. ⅠⅠ

THE PATH LOSS PREDICTION MODEL



(eq. V) Predictive Equation by linear regression

(eq.VⅠ) Predictive equation to use

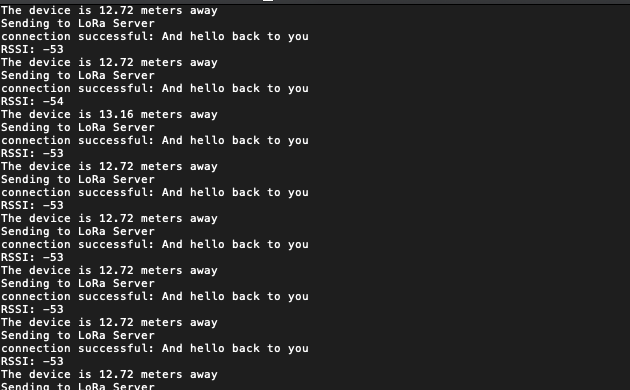
Once the equation was implemented into the Arduino code and uploaded to the end-node and microcontroller, we were able to obtain the output for the distance calculated using the predictive equation (Fig. V). First, we would get a successful connection message between the Client and Server. Next, the received RSSI would be printed. Last, the approximate distance between the gateway and Client is shown to the user. The information updates every 3 seconds, but it can be adjusted to any other time.

Figure V. Testing example (12 meters away)

By looking at the results from the model developed it was clear that it was not 100% accurate. Therefore, it was necessary to perform an analysis to determine the error margin of the model. Data collection of the distance results from the model and the measured distance was done. To predict a better accountancy in our average and error margin, we also disregarded the high and low outlines. Then, we were able to plot the results in a table (Table ⅠⅠ) and solve for the Normalized Mean Square Root Error (NMSRE) [6] to determine its accuracy and error probability. We found differences NMRSE, up to 0.125 in our model.

TABLE ⅠⅠ

RESULTS OBTAINED BY PREDICTIVE MODEL

|  |  |  |  |
| --- | --- | --- | --- |
|  | 6 METERS | 12 METERS | 16 METERS |
|  | 7.01 | 12.28 | 15.35 |
|  | **7.01** | 12.72 | 15.35 |
|  | 6.57 | 12.72 | 15.79 |
|  | 6.57 | 12.28 | 15.79 |
|  | 7.01 | **12.28** | **14.91** |
|  | 6.57 | 12.72 | **16.23** |
|  | 7.01 | 12.72 | 15.35 |
|  | **6.13** | 12.72 | 15.35 |
|  | 6.13 | 12.72 | 15.79 |
|  | 6.57 | **13.16** | 15.79 |
| TOTAL | 53.44 | 100.88 | 124.56 |
| TOTAL / 8 | 6.68 | 12.61 | 15.57 |
| NRMSE | 0.125 | 0.053 | 0.03 |

Communication takes place in random environments. As an example, in LTE signals (used for cell phones) a deterministic model cannot be developed for standard use in any environment. The conditions where the communication takes place are infinite and random. Translating that to our model, multiple factors can interfere on the performance of the model such as a neighbor, materials, or interfering signals. This was presented in the data collection, where we had variations of the dBm received on different days. We approached that problem by collecting more data for more accuracy. Even though we collected a good amount of data, the model was not completely accurate. This happens because of multiple possible factors that present on a determined day and alter the performance (noise, objects in between, other signals, etc.). The amount of data needed to improve the model is very large and it would still be very hard to make it 100% accurate.

Conclusions

It is crucial for its implementation and operation to consider the productivity and characterization of LoRa technology in indoor buildings. In this project, we have presented a comprehensive study how to utilize and predict LoRa in an indoor environment. It was a complete challenge to work in a project by using a new technology we had not utilized before. Looking at the results and after performing experiments testing the model, we realized that a perfect distance measurement is a tough task. When it comes to communication, it generally occurs in arbitrary conditions. This leads us to the fact that the model should be taken as a reference but not as a 100% accurate product. The findings presented in this paper can provide highlights into practical LoRa-based indoor applications.

Many studies have worked to try to understand the performance and limitations of LoRaWAN. For example, the researcher in [11] studied the LoRa range in an outdoor environment, its sensibility, and the difference between LoRa and other wireless communication. In a similar work [12], the researchers analyze packet payloads, radio-signal efficiency, and spatiotemporal characteristic, to model and estimate the performance of LoRaWAN in an outdoor environment. Different from these works, our work focuses on the path loss model in the indoor environment. The further scope for this project should be to apply this problem statement in a multi floor build. The best approach in this circumstance is trying to predict the dBm lost in each floor. A good example material is the [7] research, which demonstrates path loss models in different type builds and how material can affect each model. It also used a multi floor build and many collections of data to try to predict the outcome results, a similar approach we have done in this project.

Bibliography

[1] Goldstein, P., 2020. LoRa and LoRaWAN: How Technology Helps Smart Cities. *StateTech*, [online] Available at: <https://statetechmagazine.com/article/2020/01/lora-and-lorawan-how-technology-helps-smart-cities-perfcon> [Accessed 3 November 2020].

[2] Meaney, D., 2020. *Lorawan*. [online] Ecsxtal.com. Available at: <https://ecsxtal.com/lora-timing> [Accessed 3 November 2020].

[3] design, n., 2019. *Using Lorawan End Devices On The Things Network*. [online] Hackster.io. Available at: <https://www.hackster.io/nootropicdesign/using-lorawan-end-devices-on-the-things-network-206a86> [Accessed 8 November 2020].

[4]Meier, R., 2020. *Roger Meier's Freeware*. [online] Freeware.the-meiers.org. Available at: <http://freeware.the-meiers.org/> [Accessed 17 October 2020].

[5]Help.xlstat.com. 2020. *XLSTAT Support Center*. [online] Available at: <https://help.xlstat.com/s/article/what-is-statistical-modeling?language=en\_US> [Accessed 23 October 2020].

[6] SASKIA, 2019. HOW TO NORMALIZE THE RMSE. [Blog] *Marine Data Science*, Available at: <https://www.marinedatascience.co/blog/2019/01/07/normalizing-the-rmse/> [Accessed 13 November 2020].

[7] Xu, W., Kim, J., Huang, W., Kanhere, S., Jha, S. and Hu, W., 2020. Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings. IEEE Internet of Things Journal, 7(1), pp.298-310.

[8] "What is LoRa? | Semtech LoRa Technology | Semtech", *Semtech.com*, 2020. [Online]. Available: https://www.semtech.com/lora/what-is-lora. [Accessed: 15- Nov- 2019].

[9]]B. Ray, "What Is LoRa? A Technical Breakdown", *Link Labs*, 2018.

[10] [8]"Yun Shield - Wiki for Dragino Project", *Wiki.dragino.com*. [Online]. Available: https://wiki.dragino.com/index.php?title=Yun\_Shield. [Accessed: 22- Jan- 2020].

[11] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology*," 2015 14th International Conference on ITS Telecommunications (ITST), Copenhagen, 2015, pp. 55-59, doi: 10.1109/ITST.2015.7377400.*

[12]N. Blenn and F. Kuipers, “Lorawan in the wild: Measurements from the things network,” *arXiv preprint arXiv:1706.03086, 2017.*

[13] T. Hardwood, "Long Range Wireless IoT | 2019 Guide to LoRa and Other LPWAN Technologies", Postscapes. [Online]. Available: https://www.postscapes.com/long-range-wireless-iot-protocol-lora/. [Accessed: 16- Oct- 2019]

.

[14] "dragino/RadioHead", GitHub, 2018. [Online]. Available: https://github.com/dragino/RadioHead. [Accessed: 16- Nov- 2020].

[15] Dragino and Engineer-lin, "dragino/Arduino-Profile-Examples", GitHub. [Online]. Available: https://github.com/dragino/Arduino-Profile-Examples/blob/master/libraries/Dragino/examples/LoRa/LoRa\_Simple\_Client\_Arduino/LoRa\_Simple\_Client\_Arduino.ino. [Accessed: 15- Feb- 2020].

[16] ]"dragino/Arduino-Profile-Examples", GitHub, 2016. [Online]. Available: https://github.com/dragino/Arduino-Profile-Examples/blob/master/libraries/Dragino/examples/LoRa/LoRa\_Simple\_Server\_Yun/LoRa\_Simple\_Server\_Yun.ino. [Accessed: 16- Nov- 2020].

[17] “What is LoRaWAN”, Link Labs, October 29, 2018. [Online]. Available:

<https://www.link-labs.com/blog/what-is-lorawan>

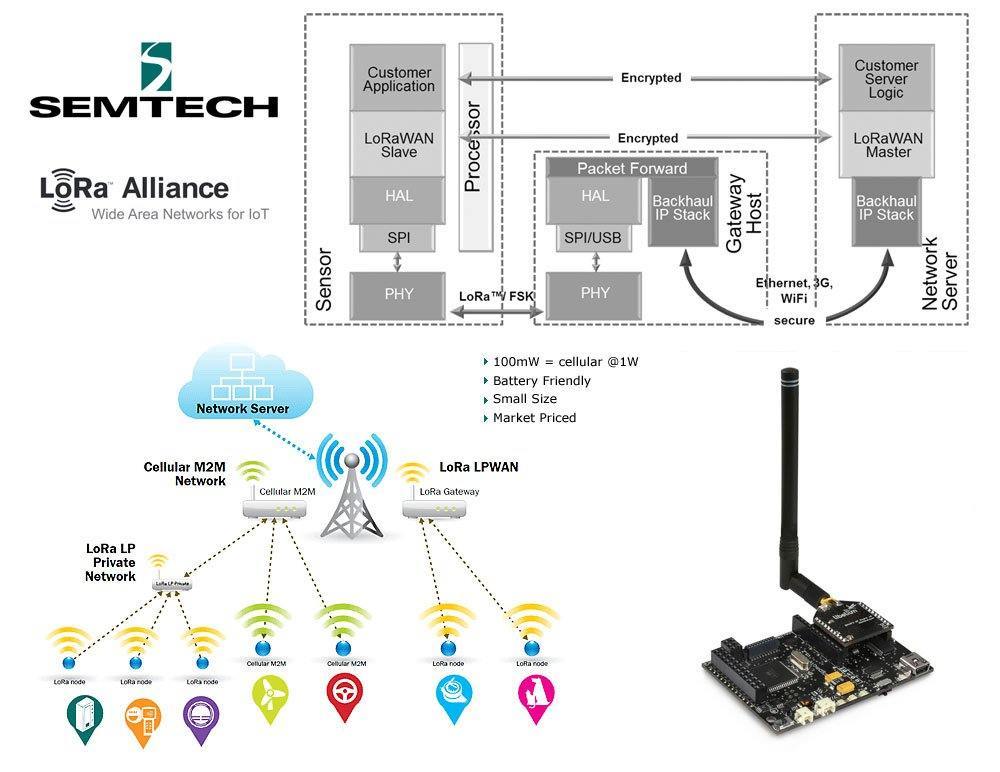
Appendix

Industrial Standards

LoRa: PHY layer

LoRaWAN: MAC, NWK, & APP built on LoRa

LoRa is a patented ( [EP2763321](https://www.google.com/patents/EP2763321A1?cl=en&dq=EP2763321&hl=en&sa=X&ved=0CB0Q6AEwAGoVChMI0J6LpfWJxwIVhdUUCh0vAwMV) from 2013 and [US7791415](http://worldwide.espacenet.com/publicationDetails/biblio?FT=D&date=20100907&DB=&locale=en_EP&CC=US&NR=7791415B2&KC=B2&ND=2) from 2008 ) spread-spectrum radio modulation originally developed by Cycleo (Acquired in 2012 by Semtech) [13]

Semtech owns the LoRa IP, but have licensed the IP to other chip manufacturers like [STMicro in 2015 in 2015](http://rethink-iot.com/2015/12/18/semtech-announces-stmicro-lora-licensing-deal-microchip-launches-11-module/).

Semtech’s LoRa chips transmit in the sub-gigahertz spectrum (109MHz, 433MHz, 866MHz, 915MHz), which is an unlicensed band that has less interference than others (like the 2.4 GHz range used by Wi-Fi, Bluetooth, and other protocols). At those frequencies, signals penetrate obstacles and travel long distances while drawing relatively little power -- ideal for many IoT devices, which are often constrained by battery life.

Within the sub-GHz spectrum, LoRa chips use a spread-spectrum strategy to transmit at a variety of frequencies and data rates. That allows the gateway to adapt to changing conditions and optimize the way it exchanges data with each device.

Semtech produces transceiver chips for devices to be connected (nodes), and gateways to connect them. A single gateway can communicate with several hundred thousand nodes up to 20 miles away in unobstructed environments, and even in a city can penetrate buildings to achieve a range of several miles. End-nodes can remain operational for a supposed 10 years running on two AAA batteries (drawing 10mA for the receiver, under 200nA in sleep mode).

Arduino Code Templates 

A code template was used for the implementation of the Client- Server network. The Client template was imported from the existing Arduino-Dragino library [15]. The Server template was imported from the existing Arduino-Dragino library [16]. Extra to the code templates, a Radio head library was imported [14]. This library allows sending and receiving packetized messages via a variety of common data radios and other transports on a range of embedded microprocessors.

*Client Code Template*

#include <SPI.h>

#include <RH\_RF95.h>

// Singleton instance of the radio driver

RH\_RF95 rf95;

//The parameters are pre-set for 868Mhz used. If user want to use lower frequency 433Mhz.Better to set

//rf95.setSignalBandwidth(31250);

//rf95.setCodingRate4(8);

float frequency = 868.0;

void setup()

{

Serial.begin(9600);

//while (!Serial) ; // Wait for serial port to be available

Serial.println("Start LoRa Client");

if (!rf95.init())

Serial.println("init failed");

// Setup ISM frequency

rf95.setFrequency(frequency);

// Setup Power,dBm

rf95.setTxPower(13);

// Setup Spreading Factor (6 ~ 12)

rf95.setSpreadingFactor(7);

// Setup BandWidth, option: 7800,10400,15600,20800,31250,41700,62500,125000,250000,500000

//Lower BandWidth for longer distance.

rf95.setSignalBandwidth(125000);

// Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)

rf95.setCodingRate4(5);

rf95.setSyncWord(0x34);

/\*

}

void loop()

{

Serial.println("Sending to LoRa Server");

// Send a message to LoRa Server

uint8\_t data[] = "Hello, this is device 1";

rf95.send(data, sizeof(data));

rf95.waitPacketSent();

// Now wait for a reply

uint8\_t buf[RH\_RF95\_MAX\_MESSAGE\_LEN];

uint8\_t len = sizeof(buf);

if (rf95.waitAvailableTimeout(3000))

{

// Should be a reply message for us now

if (rf95.recv(buf, &len))

{

Serial.print("got reply: ");

Serial.println((char\*)buf);

Serial.print("RSSI: ");

Serial.println(rf95.lastRssi(), DEC);

}

else

{

Serial.println("recv failed");

}

}

else

{

Serial.println("No reply, is LoRa server running?");

}

delay(5000);

}

*Server Code Template*

//If you use Dragino IoT Mesh Firmware, uncomment below lines.

//For product: LG01.

#define BAUDRATE 115200

//If you use Dragino Yun Mesh Firmware , uncomment below lines.

//#define BAUDRATE 250000

#include <Console.h>

#include <SPI.h>

#include <RH\_RF95.h>

// Singleton instance of the radio driver

RH\_RF95 rf95;

int led = A2;

float frequency = 868.0;

void setup()

{

pinMode(led, OUTPUT);

Bridge.begin(BAUDRATE);

Console.begin();

while (!Console) ; // Wait for console port to be available

Console.println("Start Sketch");

if (!rf95.init())

Console.println("init failed");

// Setup ISM frequency

rf95.setFrequency(frequency);

// Setup Power,dBm

rf95.setTxPower(13);

// Setup Spreading Factor (6 ~ 12)

rf95.setSpreadingFactor(7);

// Setup BandWidth, option: 7800,10400,15600,20800,31200,41700,62500,125000,250000,500000

rf95.setSignalBandwidth(125000);

// Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)

rf95.setCodingRate4(5);

Console.print("Listening on frequency: ");

Console.println(frequency);

}

void loop()

{

if (rf95.available())

{

// Should be a message for us now

uint8\_t buf[RH\_RF95\_MAX\_MESSAGE\_LEN];

uint8\_t len = sizeof(buf);

if (rf95.recv(buf, &len))

{

digitalWrite(led, HIGH);

RH\_RF95::printBuffer("request: ", buf, len);

Console.print("got request: ");

Console.println((char\*)buf);

Console.print("RSSI: ");

Console.println(rf95.lastRssi(), DEC);

// Send a reply

uint8\_t data[] = "And hello back to you";

rf95.send(data, sizeof(data));

rf95.waitPacketSent();

Console.println("Sent a reply");

digitalWrite(led, LOW);

}

else

{

Console.println("recv failed");

}

}

}

Final Client Code Implemented

#include <SPI.h>

#include <RH\_RF95.h>

// Singleton instance of the radio driver

RH\_RF95 rf95;

float frequency = 868.0;

void setup()

{

Serial.begin(9600);

//while (!Serial) ; // Wait for serial port to be available

Serial.println("Start LoRa Client");

if (!rf95.init())

Serial.println("init failed");

// Setup ISM frequency

rf95.setFrequency(frequency);

// Setup Power,dBm

rf95.setTxPower(13);

// Setup Spreading Factor (6 ~ 12)

rf95.setSpreadingFactor(7);

// Setup BandWidth, option: 7800,10400,15600,20800,31200,41700,62500,125000,250000,500000

//Lower BandWidth for longer distance.

rf95.setSignalBandwidth(125000);

// Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)

rf95.setCodingRate4(5);

}

void loop()

{

Serial.println("Sending to LoRa Server");

// Send a message to LoRa Server

uint8\_t data[] = "Hello, this is device 1";

rf95.send(data, sizeof(data));

rf95.waitPacketSent();

// Now wait for a reply

uint8\_t buf[RH\_RF95\_MAX\_MESSAGE\_LEN];

uint8\_t len = sizeof(buf);

if (rf95.waitAvailableTimeout(3000))

{

// Should be a reply message for us now

if (rf95.recv(buf, &len))

{

Serial.print("connection successful: ");

Serial.println((char\*)buf);

Serial.print("RSSI: ");

Serial.println(rf95.lastRssi(), DEC);

getDistance(rf95.lastRssi());

}

else

{

Serial.println("recv failed");

}

}

else

{

Serial.println("No reply, is LoRa server running?");

}

delay(1000);

}

void getDistance(float db){

float distance = ( db + 24.048) / (-2.2767);

String message = "The device is " + String(distance) + " meters away";

Serial.println(message);

}

Relevant Coursework

During the Electric and Electronics Engineering degree we have had many important courses that are supremely valuable to the completion of this project. One of them is Wireless Communication by Dr. Guo, which allows us to understand the characteristics and the best application of each wireless communication.

Additionally, Design of Robotic Systems course imparted by Dr. Mead. During the course we worked with Arduino software, microcontrollers and dragino technology. This expanded our knowledge in the area and gave us a deeper understanding of it.

Unquestionable that all the labs we have taken was extremely useful to be able to complete this project. Labs EEN 201, EEN 202, EEN 301, and EEN 231. Along the process, we learned how to write a report, how to deal with unexpected problems, the differences between theory and real world, and how much material interference can affect accuracy.

During our college career at Norfolk State University we had variable experiences, and real-life applications which will be extremely useful to our academy and work career.

Individual Contribution

Work and project tasks were distributed equally and according to each team’s member strength. Bruna Goncalves' individual tasks relied more on the hardware part of the project. Mario Alcaraz's individual tasks relied more on the software part of the project. Analysis, interpretation, and general development of the project was carried on working together. This includes data analysis, data collection, results discussion, calculations, etc.