The Hong Kong Polytechnic University

Department of Department of Electrical and Electronic Engineering

EIE4430 / EIE4433 Honours Project

1. Student Name: DINESH Gautam (Student No.: 20040968D)

2. Programme Code: 46402 / 46403

3. Project Title: LoRA Indoor Security System

4. Supervisor Name: Dr. Lawrence Chi-Chung CHEUNG

5. Project summary (State clearly the project objectives and results achieved by yourself.)

- Find a suitable arrangement of the nodes to ensure accurate detection.
- Improve upon the existing model to increase the accuracy of human detection.
- Improve upon the GUI to show more detailed location data.
- If needed, develop, and implement a new model to determine the location of the human more accurately.
- Find out if using more powerful antennas will improve the accuracy of the location.

DECLARATION OF ORIGINALITY

Except where reference is made in the text of this report, I declare that this report contains no material published elsewhere or extracted in whole or in part from any works or assignments presented by me or any other parties for another subject. In addition, it has not been submitted for the award of any other degree or diploma in any other tertiary institution.

No other person's work has been used without due acknowledgement in the main text of the Report.

I fully understand that any discrepancy from the above statements will constitute a case of plagiarism and be subject to the associated.

Signature

Contents page no

1. Objective …………………………………………………………………… 3
2. Introduction ………………………………………………………………… 3
3. Background …………………………………………………………………… 4
4. Methodology …………………………………………………………………... 6
5. Findings………………………………………………………………………… 8
6. Difficulties ……………………………………………………………………20
7. Future improvements …………………………………………………………21
8. Conclusion ……………………………………………………………………21
9. References …………………………………………………………………….22

1. Objectives:

This project aims to research the use case of LoRa in indoor security systems. The project is conducted using previously developed tools that use real-time RSSI values to detect the location of a human in a pre-determined area using LoRa sensors.

The tools have been developed to:

- Store the RSSI value between the senders and receivers using JAVA and Arduino programs to control the hardware.
- Determine the range of human detection based on the data collected.
- Determine the area of the human in the detection range.
- The detection area has an 80% coverage detection.
- The human location is displayed in a GUI.

While working on this project, we hope to:

- Find a suitable arrangement of the nodes to ensure accurate detection.
- Improve upon the existing model to increase the accuracy of human detection.
- Improve upon the GUI to show more detailed location data.
- If needed, develop and implement a new model to determine the location of the human more accurately.
- Find out if using more powerful antennas will improve the accuracy of the location.

2. Introduction:

LoRa technology was developed in 2009 to develop a long-range, low-power modulation technology. The technology uses CSS (Chip Spread Spectrum) modulation technology, widely used in radar and sonar applications [1]. The signals operate in the unlicensed frequency spectrum, less than 1 GHz. LoRa technology is widely used in IoT applications due to Its long range, low power consumption, and low manufacturing and installation cost [2]. Most of the applications of LoRa are in IoT; there are very few instances of it being used in security systems, as the most preferred medium in those cases is infrared sensors. One of the advantages of using LoRa over infrared sensors is that LoRa sensors cannot be fooled by the refraction of light beams. The previous project has been developed to use LoRa sensors in security systems. This project aims to optimize the previous project to provide a much more accurate location of humans and objects in the detection range, prevent shadow targets due to background noise, and provide a much more detailed GUI to showcase the locations of the objects and humans in the detection area.

3. Background:

LoRa is a proprietary radio communication method; the spread spectrum in LoRa modulation is achieved by generating a chirp signal with continuous variations in frequency. The chirp signal bandwidth is equivalent to the signal spectral bandwidth.[3] The LoRa modulation also includes a variable error correction scheme that improves the robustness of the signal at the expense of redundancy.[3] The equation for the nominal bit rate is given below:

$$Rb = SF * 1 \,/\, [\, 2^{SF}/BW] * [4 \,/\, (4 + CR)]\ bit/\mathrm{sec}\ (1)[3]$$

SF = Spreading Factor

BW = Modulation Bandwidth (Hz)

CR = code rate

This project will apply 41.7kHz bandwidth, nine spreading factors, and eight coding rates for all the LoRa devices.

This project uses the RSSI-based path loss model. The RSSI value can be used to estimate the distance between targets and nodes. Using this model, we can estimate the paths of the signals and use the data collected to calculate the position of humans in the detection area. The equation for the model is as follows:

$$RSSI = Z_0 - 10\ \tilde{a}\ log d^{2}\ (2)[2]$$

Z_0 = RSSI value at a constant distance

$\tilde{a}$ = Path Loss exponent

d = distance between the targets and nodes

The linear least-square (LLS) location estimation approach is often used due to its low processing complexity and ability to provide a closed-form solution. [2] Regarding LoRa indoor security systems, LLS has the potential to determine the linear model's parameters that relate the received signal measures of strength (RSS) to the separation between the gadgets. The LLS method can reduce the effect of Gaussian noise but not the effect of non-gaussian noise, which allows us to recode the difference in RSSI value due to a human blocking the signal. The image below shows the equation for LLS:

$$A\theta = b$$

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}, \quad \theta = \begin{bmatrix} x_t \\ y_t \\ x^2{}_t - y^2{}_t \end{bmatrix}$$

$$b = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} \quad [2]$$

After applying the Linear Least Square algorithm, the equation becomes

$$\theta = \tilde{A}b$$

$$\tilde{A} = (A^T A)^{-1} A^T \quad [2]$$

Fig 3.1

4. Methodology:

In this project, we will use 30dBi antennas with a height of 26 cm and tripods to raise the positions of the antennas. We use Arduino Uno and LoRa transmitters to broadcast and collect the RSSI values received at the target nodes. We use 18650b lithium batteries to supply power to the Arduinos and LoRa Chips, providing 3200mAh and 3.7 V to the hardware. This project will start by using 30dBi antennas with pure copper poles with a height of 26 cm, but we will also test whether a more robust antenna will help improve location accuracy.

The Arduino Uno program uses the principles of master and slave methods to communicate and collect RSSI values. They then compare the recorded RSSI values against the measured RSSI values to find the changing path, which is then used to estimate the position of the human.

This project also uses a JAVA program that facilitates the serial port communication with the master node. The program contains the serial writer and serial reader to communicate with the master node while the terminal receives user input to operate the different functions coded in the program. The program will also contain the functions to generate the GUI, the LLS equation, the RSSI comparison equation, the line equation, and the interception point equation. The functions are separated into different classes to allow them to provide the calculations separately during the program's start. The diagram below shows the JAVA program's UML diagram, which shows all the functions.

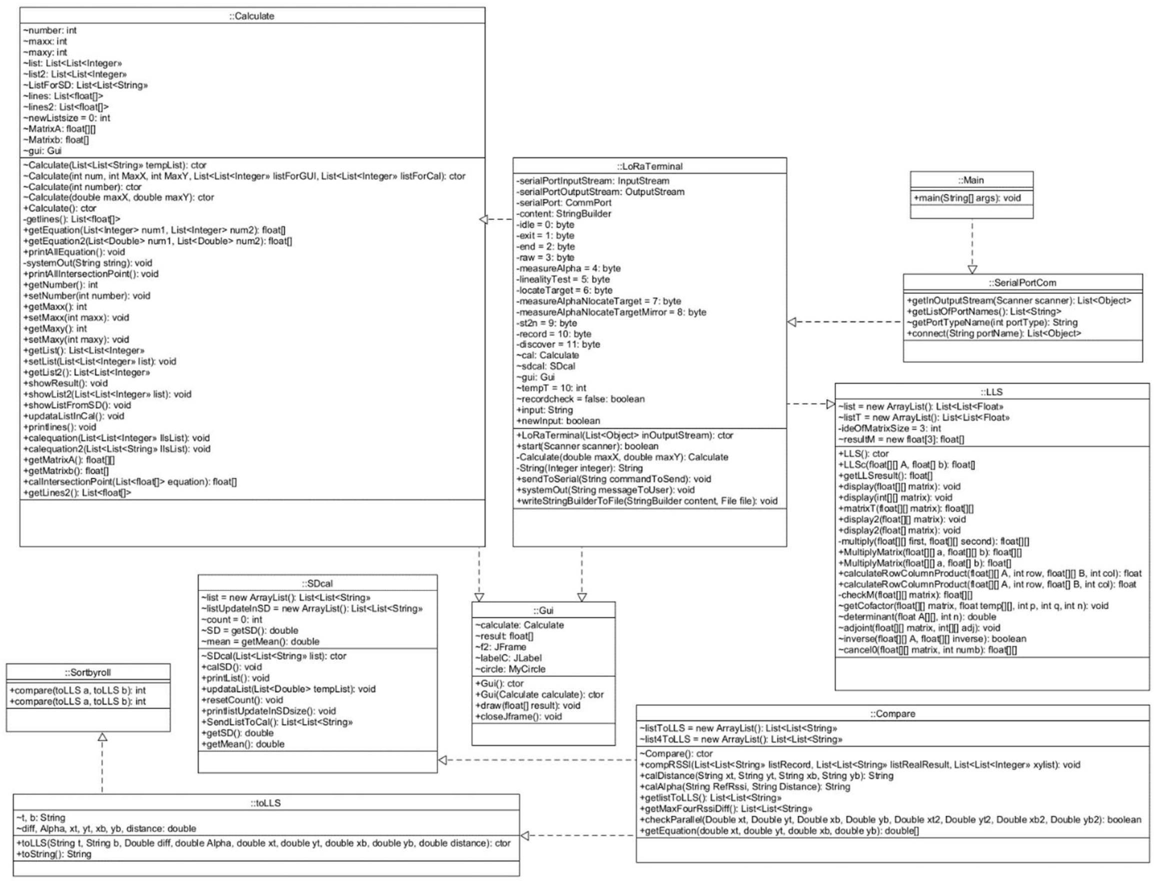


Fig 4.1

The program will take the recorded RSSI values without people to make the comparison against the measured RSSI values with people; when there are more than four records, it will select the four most considerable values of the comparison to calculate using the LLS method. After confirming the first path, the program then uses the slope of the path and checks whether the slopes of the other three paths are matched to avoid the four-path parallel problem, as the LLS method cannot manage them if the determinant is zero.

5. Findings

The first set of experiments was to find the change in RSSI value when humans block the signal. After running the experiments, the change in human blocking is between 5-10 dB.

Distance (meters)	RSSI w/o human	RSSI with human
1	-27	-37
2	-28	-35
3	-37	-44
4	-39	-49
5	-40	-50
6	-43	-50
7	-50	-49
8	-51	-50

Fig 5.1

As you can see, the RSSI values after 7 meters show that there is very little fluctuation between the RSSI with humans and RSSI without humans. From this, the effective range for each node can be determined; after finding this range, the next step was to find the optimum arrangement of nodes to ensure accurate location calculation. To find the optimum setup of the nodes, we assessed various configurations until we landed upon one that had the most lines of intersection while covering a large enough area. Some of the configurations we evaluated are shown in the following figures:

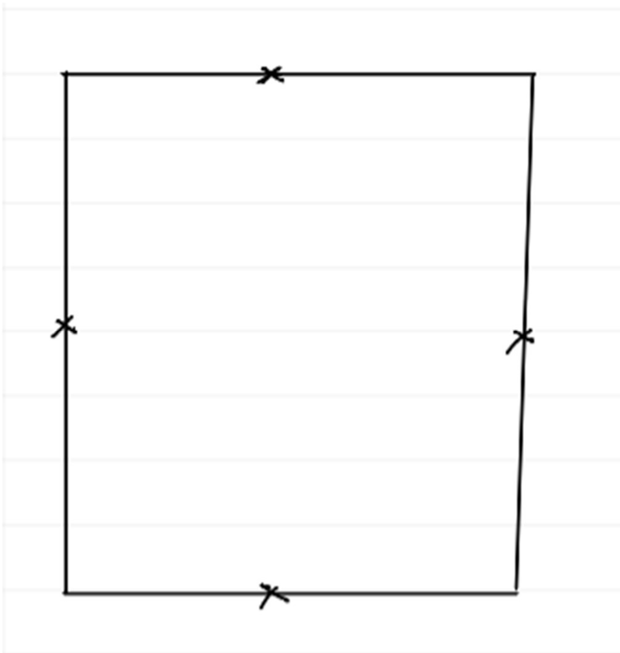


Fig 5.2

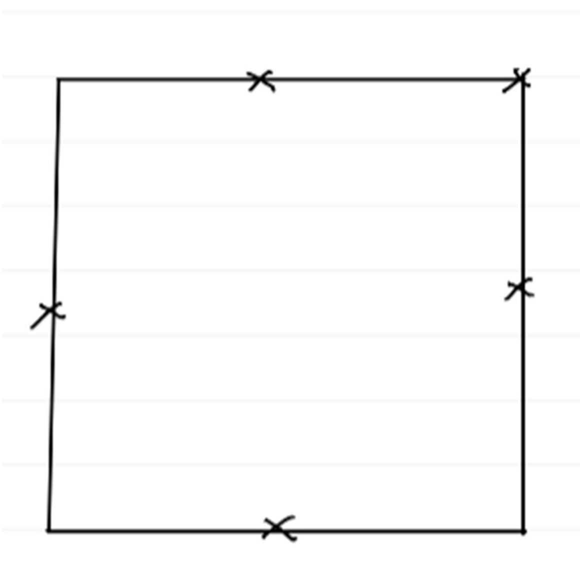
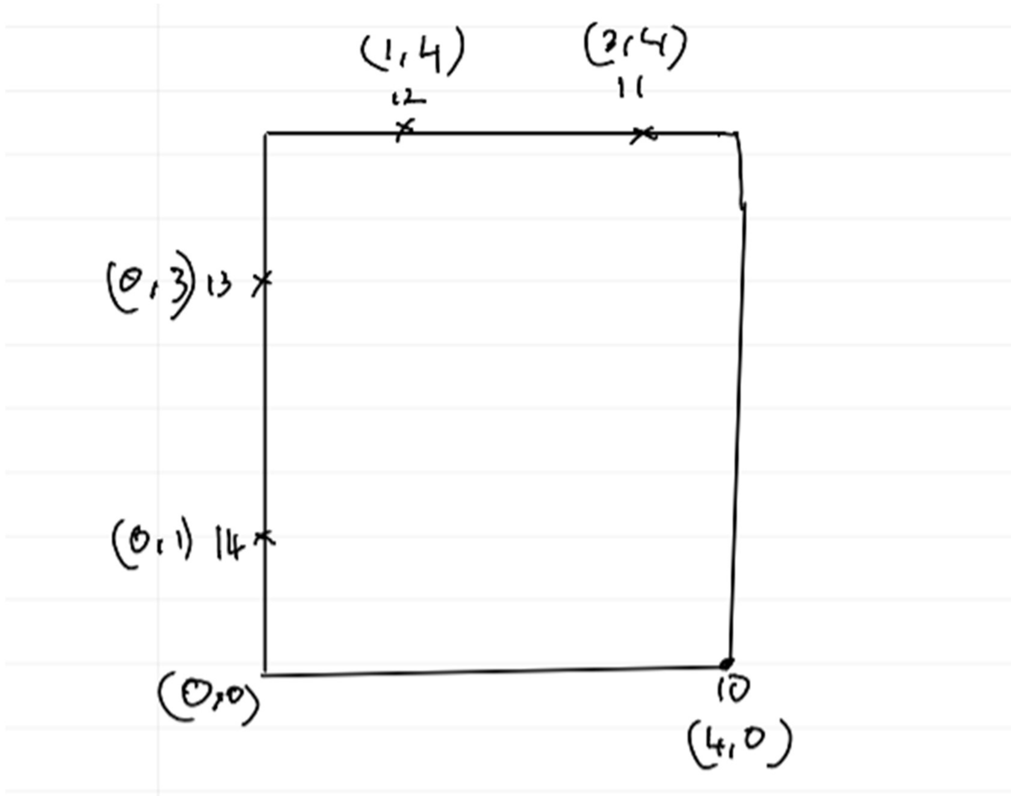


Fig 5.3

The issue with these setups was that they had too many parallel lines, meaning the calculations would not give a conclusive result for the position of the human blocking.

The best setup that we found was a fan shape with five nodes. The figure below shows a diagram of the setup.



.

Fig 5.4

The detection area was a 4m x 4m square with the nodes present at the following coordinates:

Node ID	X-Coordinate	Y-Coordinate
10	4	0
11	3	4
12	1	4
13	0	3
14	0	1

We ran three experiments with the human blocking at different positions to see how accurate the calculated positions were. In the first run, the setup was as shown in Figure 5.5.

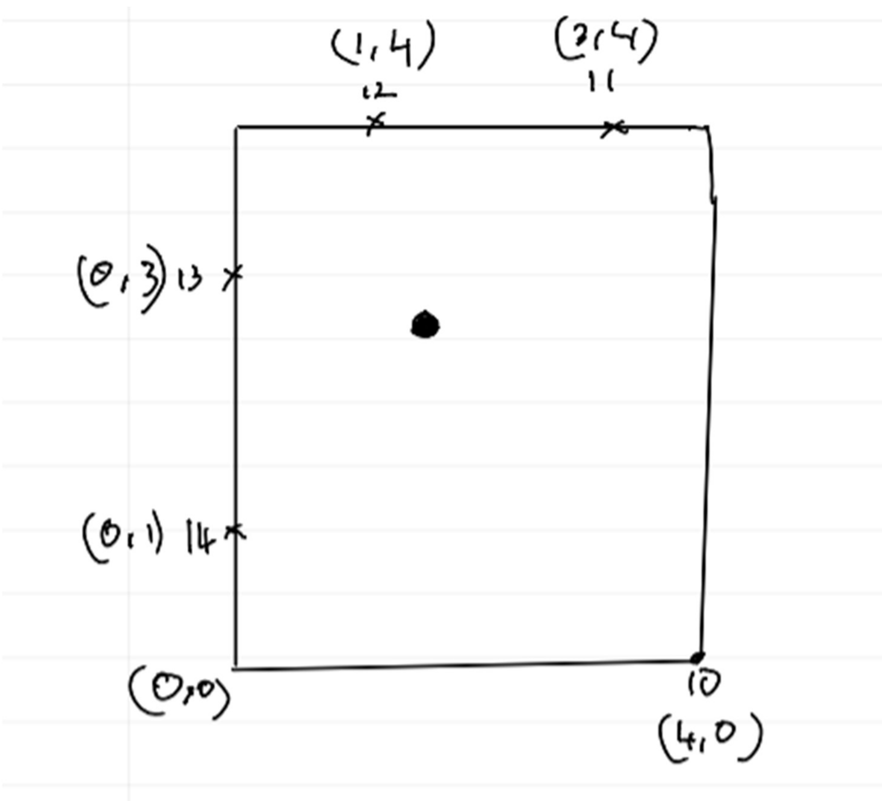


Fig 5.4

The RSSI values collected are shown in Figure 5.5.

Target	Base	RSSI w/o human	RSSI w human
	11	-38.2	-38
	12	-48	-51
	13	-33	-32
10	14	-32.2	-33
	10	-40	-37
	12	-30	-31
	13	-27.6	-27
11	14	-36.75	-52
	10	-47.6	-53
	11	-29.4	-30
	13	-24.6	-25
12	14	-33	-32
	10	-32.8	-32
	11	-27	-27
	12	-25.3	-26
13	14	-27.4	-29
	10	-33	-32
	11	-37	-46
	12	-32.8	-33
14	13	-27.4	-28

Fig 5.5

From these readings, we can see that the difference between RSSI values for node 11 and node 14 is more significant than 10, and the difference between the RSSI values for node 10 and 12 is greater than 5, meaning the line between these two nodes is used to calculate the position of human blocking.

The GUI generated by the program shows the calculated position, and the errors are shown in the figure below.

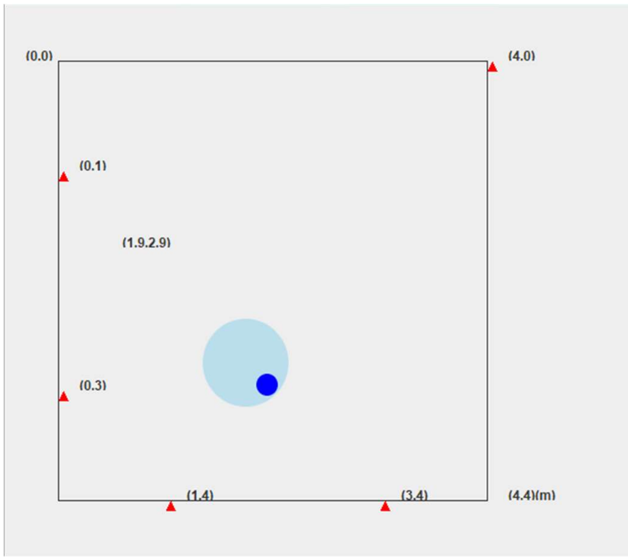


Fig 5.6

Actual X-coordinate	Actual Y-coordinate	Calculated X-coordinate	Calculated Y-coordinate	X -Error	Y-Error
1.8	2.6	1.85	2.87	1.25%	6.75%

The GUI result shows the calculated position of the person represented by the blue dot; the light blue circle shows the range of error, meaning the actual position can be anywhere within it.

The second run of the setup is shown in Figure 5.7.

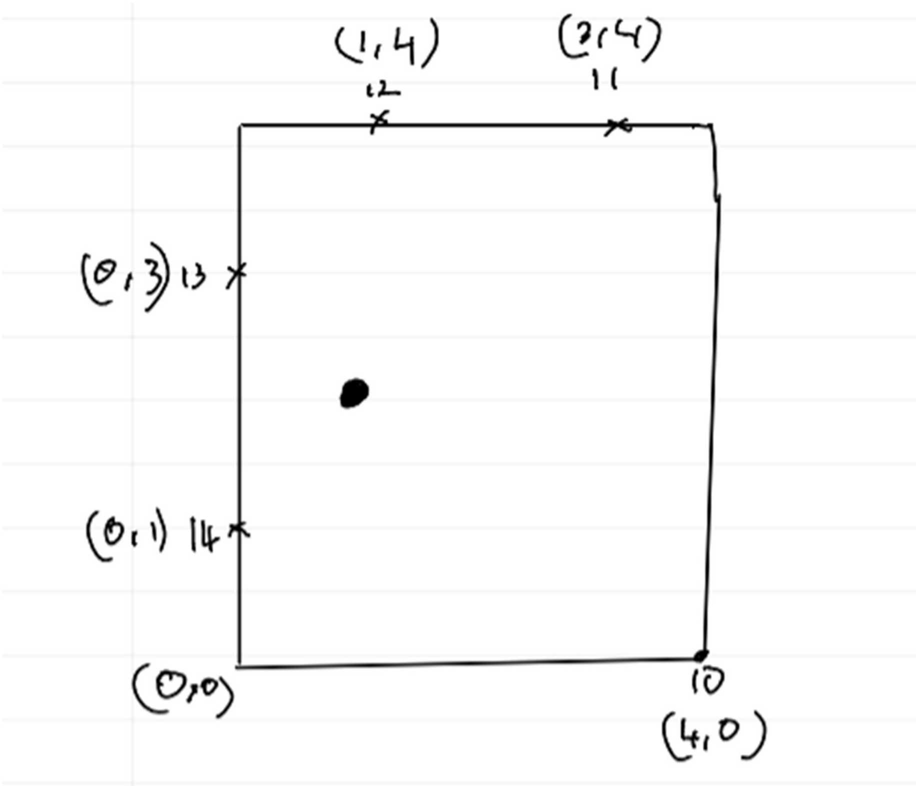


Fig 5.7

The RSSI values collected are shown in Figure 5.8.

Target	Base	RSSI w/o human	RSSI w human
	11	-38.2	-38
	12	-48	-48
	13	-33	-51
10	14	-32.2	-33
	10	-40	-37
	12	-30	-31
	13	-27.6	-27
11	14	-36.75	-52
	10	-47.6	-47
	11	-29.4	-30
	13	-24.6	-25
12	14	-33	-32
	10	-32.8	-32
	11	-27	-27
	12	-25.3	-26
13	14	-27.4	-29
	10	-33	-32
	11	-37	-46
	12	-32.8	-33
14	13	-27.4	-28

Fig 5.8

From these readings, we can see that the difference between RSSI values for node 11 and node 14 is more significant than 10, and the difference between the RSSI values for node 10 and 13 is more significant than 10, meaning the line between these two nodes is used to calculate the position of human blocking.

The GUI generated by the program is shown in Figure 5.9.

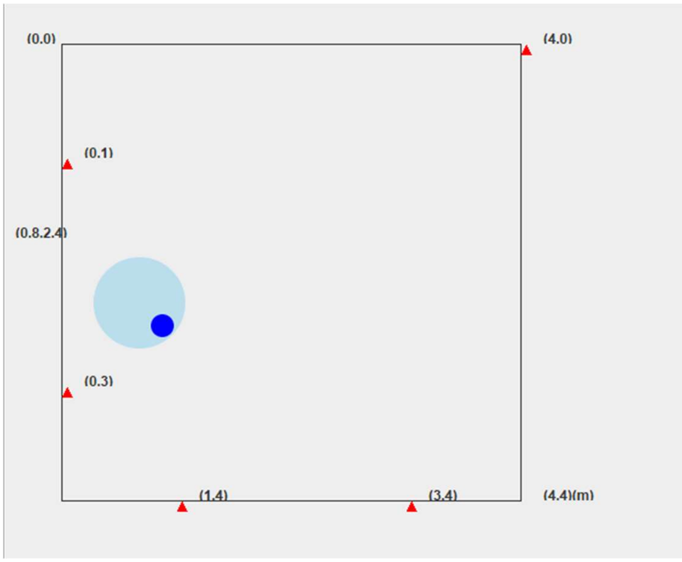


Fig 5.9

Actual X-coordinate	Actual Y-coordinate	Calculated X-coordinate	Calculated Y-coordinate	X -Error	Y-Error
0.6	2.0	0.78	2.3	4.5%	7.5%

As seen above, the error for these calculations was much higher than the previous one; the possible issue for this is the background noise due to other electronic devices in the vicinity.

The setup for the third run is shown in Figure 5.10.

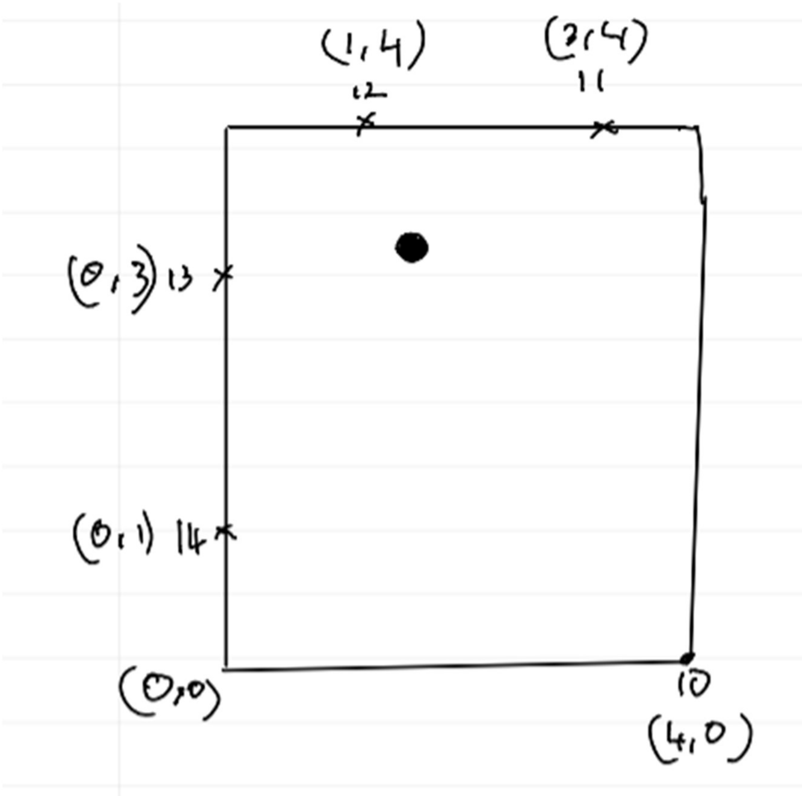


Fig 5.10

The RSSI values for this run are shown in Figure 5.11.

Target	Base	RSSI w/o human	RSSI w human
	11	-38.2	-38
	12	-48	-58
	13	-33	-32
10	14	-32.2	-33
	10	-40	-37
	12	-30	-31
	13	-27.6	-36
11	14	-36.75	-52
	10	-47.6	-57
	11	-29.4	-30
	13	-24.6	-25
12	14	-33	-32
	10	-32.8	-32
	11	-27	-36
	12	-25.3	-26
13	14	-27.4	-29
	10	-33	-32
	11	-37	-36
	12	-32.8	-33
14	13	-27.4	-28

Fig 5.11

From these readings, we can see that the difference between RSSI values for node 11 and node 13 is between 5db and 10 dB, and the difference between the RSSI values for node 10 and 12 is 10 meaning the line between these two nodes is used to calculate the position of human blocking.

The GUI generated by the program is shown in Figure 5.12.

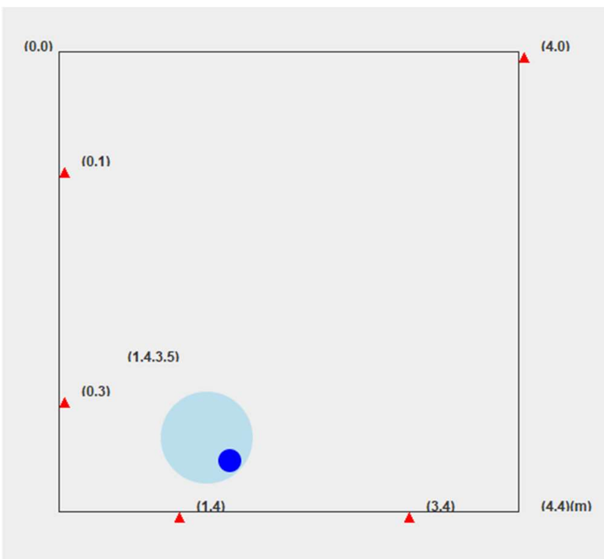


Fig 5.12

Actual X-coordinate	Actual Y-coordinate	Calculated X-coordinate	Calculated Y-coordinate	X -Error	Y-Error
1.3	3.3	1.39	3.46	2.25%	4%

6. Difficulties

During the project, we encountered a multitude of difficulties, some of which we were able to find solutions to, but some persisted till the very end. As this was a legacy project, we had taken over from someone else; this meant there was a learning curve to catch up to the progress made previously. In addition, the tools provided needed fine-tuning before experiments could be conducted.

While conducting experiments, we ran into the problem of multipath readings interfering with calculating the position of human blocking. To overcome this, we found a big enough room to construct a 4m x 4m square, with each side being a sufficient distance away from the walls and other obstructions. This helped reduce errors from multipath readings.

Another problem encountered was that some of the nodes kept timing out even if they were on and transmitting; initially, we assumed it was because the power cells were insufficiently charged; however, that was not the case. Unfortunately, we could not find out why this issue kept persisting. It might be a loose contact problem with the Arduino board, but we could not find proof.

There was also the issue when the RSSI values with human blocking did not differ at all from the RSSI without human blocking; we believe this may be because the signal may have been too strong; however, this was not a consistent problem, and it happened sporadically, so we were not able to pinpoint the exact problem causing this issue.

The presence of background noise also increased the inaccuracy of the location calculations leading to some calculated position being outside the detection area. There is no particular fix to this other than redoing the readings again.

The system only looks at the difference between RSSI with humans and RSSI without humans to decide whether humans are blocking or not. We attempted to modify the system and used the path loss model as another decisive element; however, the value alpha was not constant. The measured

RSSI without human blocking measured at a distance vastly differed from the expected RSSI calculated for the same distance.

7. Future Improvements

To further improve upon this project, one of the first methods is to increase the number of nodes in the detection area. This will allow for a larger detection area with intersecting lines and a more accurate location calculation. The figures below w\show some of the possible arrangements that can be used.

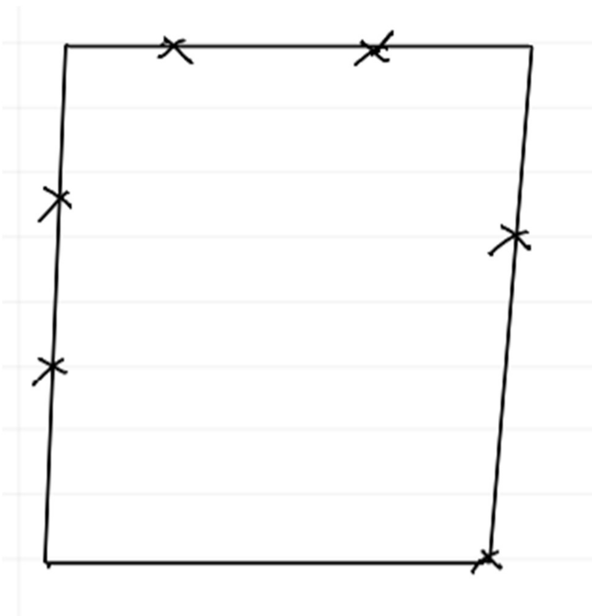


Fig 7.1 Arrangement with 6 nodes

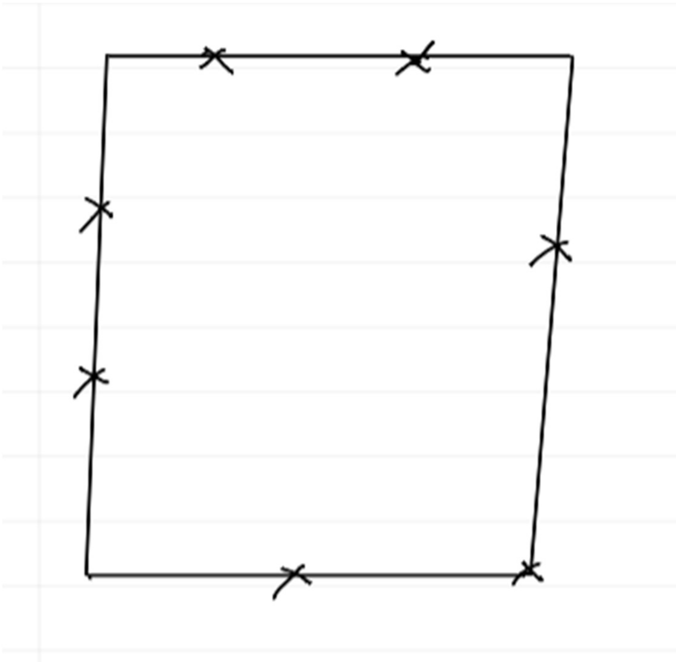


Fig 7.2 Arrangement with 7 nodes

We suggest upgrading the LoRA sensor chips on the node. This allows for more consistent readings and negates the difficulty I faced with the signal being too strong to detect a human blocking. It will also allow for testing in a much larger detection area.

Another improvement would be to modify the program to calculate the positions of multiple humans blocking. Although we could not implement this, the core principle would be to use at most two lines to calculate one person's position.

The system right only looks at the difference between RSSI values with human blocking and RSSI values without human blocking to determine the lines used to calculate the position of the obstruction. We suggest implementing the path loss model as a second decisive factor that helps improve the system's accuracy.

8. Conclusion

The project uses a system of LoRa sensor nodes to collect RSSI values between them, which are used to determine the location of the human blocking in the area. The experiments were conducted in various environments to note the differences between them to determine the optimum location type for this system; we also tested various arrangements of the nodes to find the best arrangement that gave the best accuracy in location calculations. However, this system could be better used in a real-life setting; further testing and improvements must be made before it can be used in a real setting.

References:

[1] Slats, L. (2021, August 9). A Brief History of LoRa®: Three Inventors Share Their Personal Story at The Things Conference. Semtech. https://blog.semtech.com/a-briefhistory-of-lora-three-inventors-share-their-personal-story-at-the-things-conference

[2] Lam, K. H., Cheung, C. C., & Lee, W. C. (2019). RSSI-Based LoRa Localization Systems for Large-Scale Indoor and Outdoor Environments. IEEE Transactions on Vehicular Technology, 68(12), 11778-11791. [8827665]. https://doi.org/10.1109/TVT.2019.2940272

[3] "LoRa Modulation Basics" (PDF). Semtech. Archived from the original (PDF) on 2019-07-18. Retrieved 2020-02-05. [4] Wong, S. P. (2022). Localization by using LoRa technology.