VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

CAPSTONE PROJECT

IMPLEMENTATION OF UWB-BASED INDOOR LOCALIZATION SOLUTION

Major: COMPUTER ENGINEERING

THESIS COMMITTEE:	OISP Computer Engineering 1
SUPERVISOR:	Dr. Pham Hoang Anh
REVIEWER:	Mr. Huynh Hoang Kha
	—o0o—
STUDENT 1:	Le Hoang Mai Phuong - 2053349
STUDENT 2:	Trinh Hoai Thanh - 2053427
STUDENT 3:	Ngo Chan Phong - 2053321

Ho Chi Minh City, 05/2024

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

KHOA: KH & KT Máy tính
BỘ MÔN: KTMT

NHIỆM VỤ LUẬN VĂN/ĐỒ ÁN TỐT NGHIỆP
Chú ý: Sinh viên phải dán tờ này vào trang nhất của bản thuyết trình

HỌ VÀ TÊN: NGÔ CHẤN PHONG MSSV: 2053321
HỌ VÀ TÊN: LÊ HOÀNG MAI PHƯƠNG MSSV: 2053349
HỌ VÀ TÊN: TRỊNH HOÀI THANH MSSV: 2053427
NGÀNH: KỸ THUẬT MÁY TÍNH LỚP:

1. Đầu đề luận văn/ đồ án tốt nghiệp:

Implementation of UWB-based Indoor Localization Solution

2. Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):

- Khảo sát và tìm hiểu các giải pháp định vị trong nhà và giải thuật liên quan
- Khảo sát và tìm hiểu công nghệ UWB
- Thực hiện một số thử nghiệm ban đầu trên phần cứng có hỗ trợ UWB
- Đề xuất, thiết kế và hiện mô hình
- Thực nghiệm để đánh giá kết quả thực hiện

3. Ngày giao nhiệm vụ: 08/01/2024
4. Ngày hoàn thành nhiệm vụ: 20/05/2024

5. Họ tên giảng viên hướng dẫn:	**Phần hướng dẫn:**
1) Phạm Hoàng Anh	100%
2)	

Nội dung và yêu cầu LVTN/ ĐATN đã được thông qua Bộ môn.

Ngày .... tháng .... năm ....

CHỦ NHIỆM BỘ MÔN	**GIẢNG VIÊN HƯỚNG DẪN CHÍNH**
(Ký và ghi rõ họ tên)	*(Ký và ghi rõ họ tên)*
PHẠM QUỐC CƯỜNG	PHẠM HOÀNG ANH

PHẦN DÀNH CHO KHOA, BỘ MÔN:
Người duyệt (chấm sơ bộ):
Đơn vị:
Ngày bảo vệ:
Điểm tổng kết:
Nơi lưu trữ LVTN/ĐATN:

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KH & KT MÁY TÍNH

Ngày 27 tháng 05 năm 2024

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU ĐÁNH GIÁ LUẬN VĂN/ ĐỒ ÁN TỐT NGHIỆP

(Dành cho người hướng dẫn)

1. Họ và tên SV: **Ngô Chấn Phong, Lê Hoàng Mai Phương, Trịnh Hoài Thanh**

   MSSV: **2053321, 2053349, 2053427** Ngành (chuyên ngành): **Kỹ thuật Máy tính**

2. Đề tài: **Implementation of UWB-based Indoor Localization Solution**

3. Họ tên người hướng dẫn: **Phạm Hoàng Anh**

4. Tổng quát về bản thuyết minh:

Số trang: Số chương:
Số bảng số liệu Số hình vẽ:
Số tài liệu tham khảo: Phần mềm tính toán:
Hiện vật (sản phẩm)

5. Những ưu điểm chính của LV/ ĐATN:
- Students have completed most of the requirements of a capstone project, including:
  - Investigate related works to indoor positioning solutions
  - Investigate the capability of UWB technology in indoor positioning solutions
  - Propose a system model that applies UWB to indoor localization applications
  - Conduct experiments to evaluate the proposed system
- The experimental results have shown the effectiveness and beter accuracy of the UWB technology in indoor localization compared with other technologies such as WiFi and Bluetooth.
- Students have demonstrated the ability to self-study new knowledge and apply it into the implementation of this capstone project appropriately.

6. Những thiếu sót chính của LV/ĐATN:
- The current implementation is only at a testing level and not as complete as expected.

7. Đề nghị: Được bảo vệ ☒ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐

8. Các câu hỏi SV phải trả lời trước Hội đồng:

9. Đánh giá chung (bằng chữ: Xuất sắc, Giỏi, Khá, TB): Xuất sắc Điểm: **8.9**/10

Ký tên (ghi rõ họ tên)

Phạm Hoàng Anh

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KH & KT MÁY TÍNH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Ngày 31 tháng 05 năm 2024

PHIẾU ĐÁNH GIÁ LUẬN VĂN/ ĐỒ ÁN TỐT NGHIỆP

(Dành cho người ~~hướng dẫn~~/phản biện)

1. Sinh viên thực hiện:

Ngô Chấn Phong	2053321	Ngành (chuyên ngành): Kỹ thuật Máy tính
Lê Hoàng Mai Phương	2053349	Ngành (chuyên ngành): Kỹ thuật Máy tính
Trịnh Hoài Thanh	2053427	Ngành (chuyên ngành): Kỹ thuật Máy tính

2. Đề tài: Implementation of UWB-based Indoor Localization Solution (Hiện thực giải pháp định vị trong nhà sử dụng UWB)

3. Họ tên người ~~hướng dẫn~~/phản biện: Huỳnh Hoàng Kha

4. Tổng quát về bản thuyết minh:

Số trang: Số chương:
Số bảng số liệu Số hình vẽ:
Số tài liệu tham khảo: Phần mềm tính toán:
Hiện vật (sản phẩm)

5. Những ưu điểm chính của LV/ ĐATN:

- Students studied and perceived background knowledge and applications of the UWB technology, together with techniques and algorithms applicable in indoor localization.
- Selection of distance measuring techniques and positioning algorithms were conducted reasonably, based on both theory and reality.
- Students also tested and successfully deployed an indoor positioning system adopting the UWB technology. Practical experiments exhibited that the product had an acceptable accuracy for a wide range of applications.

6. Những thiếu sót chính của LV/ĐATN:

- The project results were merely at the extent of hardware testing.
- The system model demonstrated in the project may not meet most application requirements in practice as it supported only one tag.

7. Đề nghị: Được bảo vệ ☒ Bổ sung thêm để bảo vệ □ Không được bảo vệ □

8. Các câu hỏi SV phải trả lời trước Hội đồng:

- To make the product of this project come in handy for realistic applications, what must be conducted?

9. Đánh giá chung (bằng chữ: Xuất sắc, Giỏi, Khá, TB): Điểm : **8.5** /10

Ký tên (ghi rõ họ tên)

Huỳnh Hoàng Kha

Commitment

We would like to assure that this Capstone Project is a serious and honest scientific research work of the group under the guidance of Dr. Pham Hoang Anh. All content presented in this specialized project, except for the parts that have been clearly annotated and cited in the reference materials section, is done by the team ourself.

We take full responsibility if there is any fraud in the content of the group's Capstone Project.

Acknowledgement

First of all, we would like to express our gratefulness to all Ho Chi Minh University of Technology lecturers for giving us meaningful years of study, and especially our main direct instructor, Dr. Pham Hoang Anh. The person who provided knowledge, direction, and guidance documents as well as followed us during the process of implementing the project. He not only provided us with knowledge but also with spirit and working space so that we would have more motivation and ability to focus on researching and writing our thesis in the most effective and optimal way.

Besides, the knowledge that we have is indispensable thanks to the dedicated lecturers in the Department of Computer Science and Engineering. In addition, mental and physical motivation comes from family, relatives and friends to help us have enough strength and health to successfully complete the specialized project. Although we have completed our specialized project as well as possible but errors are certainly unavoidable. Therefore, we look forward to receiving opinions and suggestions from teachers to improve, move towards development and be a useful reference for the next stage of the thesis.

Finally, we would like to send our best wishes to all educators for enduring good health, ongoing success, and the continuous guidance of future generations of students.

Best regards.

Summary

Indoor localization technology has become a focal point of innovation. This dynamic field is dedicated to providing accurate location information for objects, devices, or individuals within buildings, warehouses, and other indoor environments. As our reliance on smart technologies grows, so does the demand for seamless navigation and context-aware applications indoors.

Ultra-Wideband (UWB) indoor localization technology represents a cutting-edge approach to achieving spatial awareness in limited spaces. This technology's exceptional accuracy, low power consumption, and resistance to signal interference make it a game-changer for various applications, from smart homes to industrial settings.

In the field of indoor localization techniques, there are many algorithms and approaches, each presenting unique advantages and challenges. As we work on the project, it is important to undertake thorough research to determine which methodologies align most effectively with UWB technology and meet the requirements of our specific objectives. Subsequently, develop an application that ensures a user-friendly experience, offering intuitive displays of location data and facilitating user interaction.

In the first stage, Computer Engineering Project, our objectives are to conduct research, experiments, and implementation to identify suitable solutions for indoor localization using UWB.

Following this, in the Capstone Project, we transition from the research phase to implementing the findings into an application, culminating in the development of a finalized indoor localization system. Then, conclude our project by conducting tests and evaluating the system.

Table of contents

Tables

List of figures

List of Acronyms

AOA	Angle of Arrival
AoD	Angle of Departure
APs	Access Points
APIs	Application Programming Interfaces
BLE	Bluetooth Low Energy
CSS	Cascading Style Sheets
DF	Direction Finding
DOM	Document Object Model
EKF	Extended Kalman Filter
FTP	File Transfer Protocol
GPS	Global Positioning System
HF	High Frequency
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	Identifier
IDE	Integrated Development Environment
IFTTT	If This, Then That
IoT	Internet of Things
LF	Low Frequency
LOS	Line of Sight
MAC	Media Access Control
MLAT	Multi-lateration
MQTT	Message Queuing Telemetry Transport
NLOS	Non-Line-of-Sight
ORM	Object-Relational Mapping
PC	Personal Computer
RFID	Radio Frequency Identification

RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTLS	Real-Time Location System
SMTP	Simple Mail Transfer Protocol
TDoA	Time Difference of Arrival
ToF	Time of Flight
TOA	Time of Arrival
TWR	Two Way Ranging
UART	Universal Asynchronous Receiver/-Transmitter
UHF	Ultra-High Frequency
UI	User Interface
UKF	Unscented Kalman Filter
USB	Universal Serial Bus
UWB	Ultra-Wideband
WPS	Wi-Fi Positioning System
WSNs	Wireless Sensor Networks

CHAPTER 1

INTRODUCTION

1.1 Reasons for choosing the Topic

1.1.1 Problems

The exploration and utilization of Internet of Things (IoT) applications along with indoor localization are subjects being researched and developed in the current era of technological advancement. The application of indoor localization technology facilitates seamless navigation, asset tracking, and context-aware services in various domains such as smart buildings, healthcare, and retail environments.

Indoor localization faces several challenges that impact its accuracy and reliability. Some of the major problems include:

- **Multipath Propagation:** Indoor environments often have reflective surfaces such as walls and furniture, leading to multipath propagation. Signals can take multiple paths before reaching a receiver, causing interference and inaccuracies in distance measurements.
- **Dynamic Environments:** Indoor spaces are dynamic, with people and objects constantly moving. Tracking and localizing entities accurately in such environments require robust algorithms that can adapt to changes in real-time.
- **Signal Attenuation:** Walls, floors, and obstacles in indoor spaces can attenuate wireless signals, reducing their strength. Signal attenuation poses a challenge to accurate distance estimation, especially in situations where obstacles obstruct the direct line of sight.

- **Non-Line-of-Sight (NLOS) Conditions:** The presence of obstacles in indoor spaces creates non-line-of-sight conditions, making it challenging to establish direct communication paths between devices. NLOS conditions complicate the accuracy of localization algorithms.
- **Power Consumption:** Many indoor localization devices, especially those in resource-constrained environments, face challenges related to power consumption. Balancing the need for accurate localization with energy efficiency is a critical consideration.
- **Costs:** The challenge of costs in indoor localization arises from the significant investment needed for deploying infrastructure elements, developing advanced technologies, and addressing scalability issues, which could interfere widespread adoption.

1.1.2 UWB Indoor Localization Technology

Ultra-Wideband (UWB) technology has emerged as a solution for indoor localization, offering a range of advantages in accuracy and versatility. UWB utilizes a broad spectrum of frequencies, enabling precise time-of-flight (ToF) measurements for calculating distances between devices. This capability results in high-precision localization, making UWB particularly suitable for applications such as asset tracking, navigation, and context-aware services within indoor environments. Its resistance to multipath interference, fine spatial resolution, and low power consumption further enhance its effectiveness. This introduction explores the fundamental features that make UWB a compelling choice for indoor localization, highlighting its potential to address the challenges posed by dynamic indoor spaces and demanding applications.

UWB technology has the capability to address existing challenges in localization methods:

- **High Accuracy:** UWB provides precise distance measurements, enabling highly accurate localization. This accuracy is crucial for applications such as asset tracking, navigation, and location-based services within indoor environments. UWB signals have a wide bandwidth, allowing for fine spatial resolution. This feature is beneficial for distinguishing between closely spaced objects or devices within confined indoor spaces.

- **High Data Rate Capability:** UWB supports high data rates, allowing for efficient communication and data exchange between devices. This feature is advantageous for applications requiring real-time updates or the transmission of large amounts of data.
- **Immunity to Multipath Interference:** UWB is known for its resistance to multipath interference, a common challenge in indoor environments. This resilience ensures more reliable and accurate distance measurements even in environments with reflective surfaces.
- **Good Penetration Through Materials:** UWB signals exhibit good penetration capabilities through materials like walls and obstacles, maintaining signal strength and accuracy even in non-line-of-sight conditions.
- **Low Power Consumption:** UWB technology can operate with low power consumption, making it suitable for battery-powered devices and extending the lifespan of sensors or tags deployed for indoor localization.
- **Cost-effective:** UWB's precise distance measurements and resistance to multipath interference can reduce the need for extensive infrastructure deployment, contributing to cost savings in system implementation and maintenance. Additionally, the technology's versatility and compatibility with various applications make it a cost-efficient choice for achieving accurate indoor localization.

1.2 Introduction of the Hardware

In this project, we used Qorvo's MDEK1001 Ultra-wideband (UWB) development kit. The kit includes 12 DWM1001-DEV development boards in plastic enclosures. Each can be configured as an anchor, tag or bridge node.

Kit contents:

- 12 DWM1001-DEV boards configurable as anchors, tags or bridge node. (A DWM1001-DEV configured as bridge node and associated with a Raspberry Pi 3 model B will constitute a gateway)
- Plastic enclosures for all development boards
- USB cable for flashing and debugging (can also provide power via USB power supply or battery pack)

- Quick start guide, adhesive pads, right-angled USB connectors, coloured stickers

Figure 1.1: DWM1001-DEV board with the cover

System Performance:

- Maximum tag location rate: 10 Hz
- X-Y location accuracy: typically <10 cm
- Point to Point range: up to 60 m in Line-of-Sight conditions
- RTLS scheme range: 25 to 30 meters between anchors (4+) and tags
- Adaptive location rate using motion sensor activity enables longer battery life & higher tag density

1.3 Project Objectives

The major purpose of this project is to develop a UWB-based indoor localization system capable of providing accurate and real-time positioning information. To accomplish this objective, we must complete the following tasks:

- *First*, research about the background knowledge related to the topic.
- *Second*, implement and experiment different methods and algorithms to achieve the expected requirements.

[0]Qorvo. *MDEK1001 Ultra-Wideband (UWB) Transceiver Development Kit.* `https://www.qorvo.com/products/p/MDEK1001`

- *Third*, develop the software to handle UWB data, implement the chosen localization algorithm, and visualize the position estimates.
- *Fourth*, test and validate the system accuracy along with performance under different conditions and scenarios.

1.4 Specific Requirements

The specific requirements for this UWB localization application include:

- **Accuracy:** The system is expected to achieve meter-level accuracy, with an error margin of less than 0.3 meters, even in non-line-of-sight (NLOS) environments.
- **Coverage Area:** The simplest setup of the system could cover an indoor space of up to 54 square meters while maintaining an error margin below 0.3 meters.
- **Display latency:** The tag data will be displayed on the application within 6 seconds after the tag moves.
- **User-Friendly Application:** Develop an easy-to-use app that effectively visualizes localization data in an understandable manner.

1.5 Scope of the Project

About the timeline, this thesis is divided into two phases:

- **Phase 1**, Computer Engineering Project, to address the first and second objectives. Duration in 15 weeks.
- **Phase 2**, Capstone Project, to address the third and fourth objectives. Duration in 15 weeks.

About the goal, During the first phase, our primary goal is to identify optimal methods for UWB indoor localization through research and experimentation. Then, we aim to propose a system architecture that we plan to work on and develop a plan for future steps. In the second phase, our main goal is to develop an application to display our UWB system and evaluate its performance. During this phase, we will also finalize our detailed system and integrate it into a comprehensive application for launch.

About the expected result, our objective is the fulfillment of all our requirements, which will be validated through planned experiments and observation of their results. Following this validation process, any necessary adjustments will be made to our system to ensure compliance with the established requirements.

1.6 Outline of the Thesis

The thesis is organized with a structure divided into six chapters as follows:

Chapter 1: clarify our choice of topic and the reasoning behind opting for UWB technology. It also outlines the targets and requirements of our project, alongside our planned approach to accomplishing these goals.

Chapter 2: presents all the background knowledge that we need and related to this system. Includes localization technologies, measurement techniques, and localization algorithms.

Chapter 3: details the methods and algorithms we've selected after thorough research. It explains the reason behind our choices and outlines how we intend to integrate them into our system.

Chapter 4: the final system architecture along with how we set up our system based on that architecture. Then, shows the results of launching our application.

Chapter 5: shows our experiment scenarios and the results of those experiments. Experiment scenarios are based on our system requirements.

Chapter 6: conclusion about what we did and what we learned throughout our project. It shows the evaluation of our application and suggests directions for future improvements.

CHAPTER 2

BACKGROUND KNOWLEDGE

2.1 Indoor localization technology

2.1.1 Wi-Fi Positioning system

The Wi-Fi positioning system (WPS) has been increasingly uses in a variety of aspects recently. While GPS navigation cannot work effectively because of signal blocking, the abundance of pathways, and other factors, Wi-fi can be done in both internal and external side of places where people want to have navigation in.

Wi-Fi access points that dispatch specific data are used to define coordinates in the wireless indoor positioning system. The multilateration approach allows the system to determine the precise location of the user's device using the RSSI and MAC-address. As a result, The quantity of access points, the layout of the building, and the type of smartphone all affect how accurate Wi-Fi positioning is. In conlusion, the Wi-Fi location system makes use of the complete Wi-Fi network, which includes phones and tablets in addition to Wi-Fi routers. The system can easily specify the location of any Wi-Fi-powered devices and manage their mobility inside the building by using Wi-Fi access points.

Although Wi-Fi positioning has a wide variety of advantages such as no necessity in additional equipment and infrastructure maintenance or the co-operation with other positioning technologies for improving accuracy and a more flexible deployment, it is still unable to prevent the low accuracy of Wi-Fi positioning with accuracy at the higher end of 15 meters, it may not be as suitable for precise location applications. Besides, when security of the system is concerned, Wi-Fi positioning has a low security. As a result, Wi-Fi positioning has witnessed a large number of hack, which is a warning issue for those who are going to use it.

2.1.2 Bluetooth Positioning system

Bluetooth positioning system has been recently called Bluetooth Low Energy (BLE). The growth of the BLE enables the numberous number of devices which are used for indoor places to position. 2019 has witnessed the release of Bluetooth 5.1 which is poised to bring enhanced abilities leading to more precise location detection via direction finding (DF), which can deliver centimeter level accuracy.

BLE will be applied through beacons, which are small, versatile, and can be determined by BLE using devices while they are consuming a very small amount of energy. Beacons can be put in some specific place like walls or mobile device to provide the indoor positioning applications the position where they are.

Comparing to Wi-Fi positioning system, BLE has a more effective positioning accuracy when it can deliver location accuracy below 5 meters. Besides, BLE requires significantly less power, which is necessary for resource consumption. On the contrary, BLE applications will need beacons in order to be able to position while Wi-Fi can communicate over longer ranges and higher data rates, both areas where BLE is much more limited.

2.1.3 Radio Frequency Identification (RFID)

Nowadays when the need of improving more and more convenience in our life has appeared, a lot of wireless technologies are created such as Wifi and BLE which have been mentioned above, but there is one technology called RFID which have been used for a very long time since 1970s. Thanks to the utilization of electromagnetic and electrostatic that RFID apply to, it can connect to the radio frequency portion of the electromagnetic spectrum to seperately determine an object, animal or person. RFID is the system that is provided of 3 components: antenna, reader and transponder. The reader which has one or more antennas is the main part when it emanate radio waves and transponder will send the signals back to the reader. Transponder or tags, employ radio waves to read identity communication and information from nearby devices. RFID system is divied into 3 core patterns: There are three main types of RFID systems: low frequency (LF), high frequency (HF) and ultra-high frequency (UHF). Low-frequency RFID system bounds from 30 KHz to 500 KHz, though the traditional frequency is 125 KHz. LF RFID has short transmission bound, from a few centimeters to smaller than 2 meters. High-frequency RFID system vary from 3 MHz to 30 MHz, with the traditional HF frequency being 13.56 MHz. The standard variation is from a few centimeters to several 5 meters. UHF RFID systems vary from 300 MHz to 960 MHz, with the typical frequency of 433 MHz and can be read from over 8 meters.

RFID can be varied in many applications, such as inventory control, vehicle tracking, and healthcare. Thanks to the productiveness when it can determine single objects without direct line of sight. Besides that, RFID technology can enhance security in businesses. It can guarantee areas, determine and pursue valuable goods and prevent bad people to stole it. On the contrary, RFID Technology has some disadvantages. One of the main disadvantages of RFID tags is their lack of robust security. On the other hand, RFID systems can be more expensive to implement when comparing to barcode system.

2.1.4 Ultra-Wideband Positioning system

2005 witnessed an establishment for Ultra-Wideband technology by FFC (Federal Communications Commisson). Similar to Wi-fi and BLE positioning system, Ultra-Wideband positioning system is able to transmit data between radio waves when it is based on radio frequency to work precisely. Although BLE is radio frequency positioning like Wi-fi and BLE, it is able to work with high frequencies and transmit large amounts of data over short distances.

UWB operating principle is the use of short pulse of radio waves and use techniques that cause a spreading of the radio energy (over a wide frequency band) with a very low power spectral density. These pulses help the receiver to triangulate the position and movement of an object through what is known as “ranging”. When ranging, one UWB device which is called as a tag delivers radio pulses to other device named as anchors. The distance between these devices can be calculated by measuring the time it takes for these pulses to travel from the transmitting device to the receiving device - known as Time of Flight (ToF). Besides, other algorithms can be applied like Angle of Arrival Measurements (AOA) or Time of Arrival Measurements (TOA). In conclusion, its accuracy is very high for about 10 to 30 centimeters.

In the present age, when people all want a indoor positioning technology that is cost-effective, low power consumption and high accuracy for a wide variety of purposes like goods positioning or vacuum cleaners positioing. UWB has much more upgrades than other indoor positioning technologies when comparing such as accuracy from 10 to 30 centimeters, frequency from 3.1 to 10.6 GHz and data rate up to 27 Mbps. UWB transmissions can penetrate walls and obstacles and provide high accuracy location determination when other technology as meet walls and obstacles the range may be reduced, which affects the effectiveness. Moreover, from its name "Ultra-wideband", it can be seen that UWB provides a frequency diversity to fight against interference which can be harmful to indoor positioning. Lastly, UWB device does have a significantly low price, and it uses a small amount of power that enablse power efficiency for better battery life of devices.

The image below show the comparison of UWB technology performance to Bluetooth, WiFi, RFID, and GPS for localization purposes:

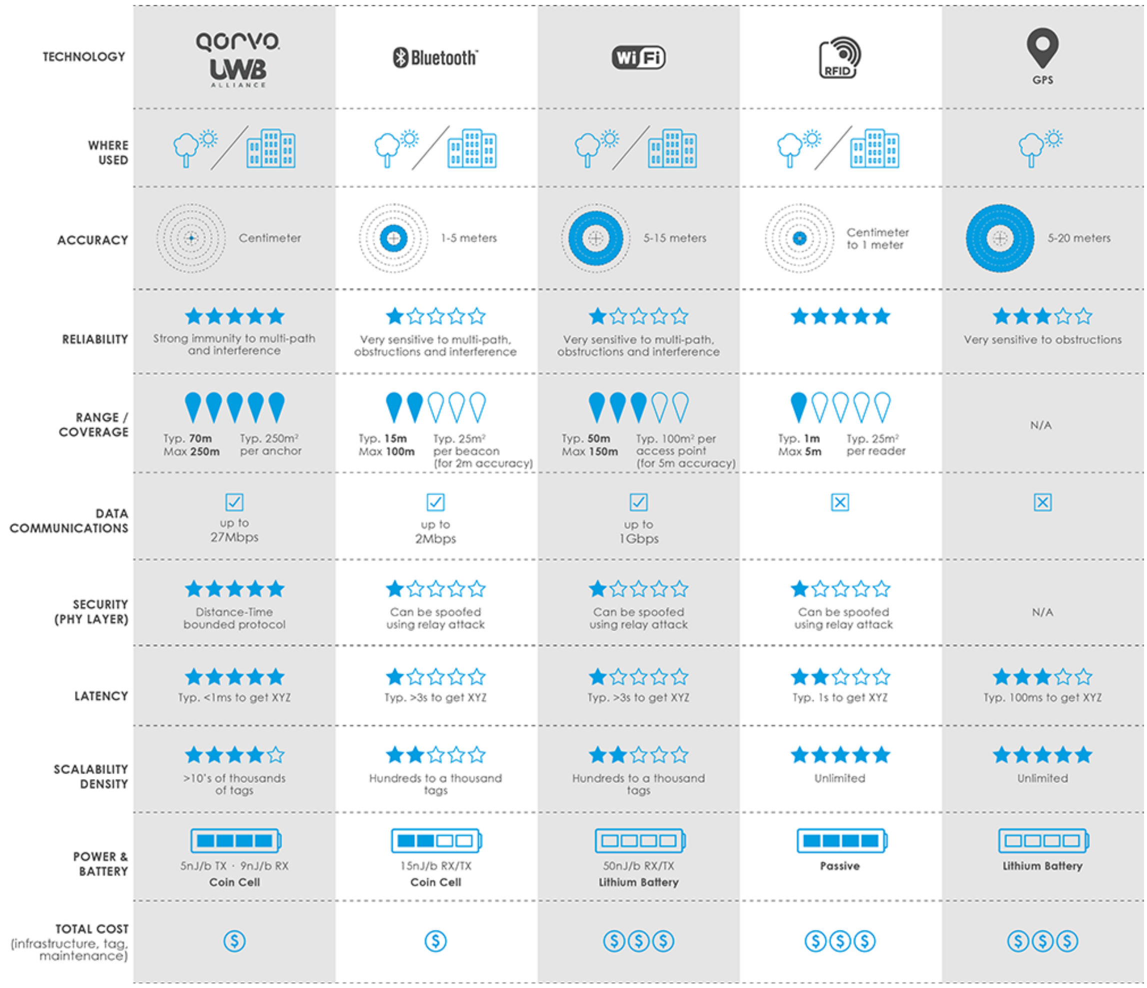

Figure 2.1: Technologies comparison

2.2 Localization measurement techniques

Measurement techniques for wireless sensor networks (WSNs) are typically categorized into three main classes: Angle-based methods, Distance-based methods, and Fingerprinting-based methods. These three categories represent diverse approaches to solving the challenges of wireless positioning, each tailored to specific application needs and environmental conditions.

2.2.1 Angle-based methods

Angle-based methods in indoor localization involve determining the direction or angle from which signals arrive at a receiver, providing information about the position of objects or devices within an indoor environment. Two common angle-based methods are Angle of Arrival (AoA) and Angle of

[0] Qorvo. *UWB technology*. `https://www.qorvo.com/innovation/ultra-wideband/technology`

Departure (AoD).

2.2.1.1 Angle of Arrival measurements

Angle of Arrival (AoA) measurement techniques, refer to a technique used in wireless communication systems to determine the direction from which a signal arrives at a receiving antenna or array of antennas. These methods calculate the angle at which a signal arrives from an anchor node to an unknown sensor node. Subsequently, the region where the unknown sensor is situated forms a line at a specific angle from the anchor node. To execute AoA measurement techniques, a minimum of two anchor nodes is necessary for position calculation. However, a small measurement error can result in a considerable localization error. The accuracy is contingent on the antenna's directionality, and measurements are further complicated by shadowing and multipath effects in the environment. Multipath components can lead to signals being perceived from different directions, causing substantial inaccuracies. Therefore, AoA techniques are of limited practicality in localization unless applied with large antenna arrays, making them less energy-efficient for WSNs with small sensor nodes. Disadvantages of this method include complex implementation, requiring arrays of antennas and complicated signal processing.

2.2.1.2 Angle of Departure measurements

Angle of Departure (AoD) measures the direction from which signals depart from a transmitting antenna. This method is particularly relevant in scenarios where understanding the directional characteristics of the transmitted signal is important. In AoD, the transmitting device uses multiple antennas or antenna arrays to emit signals in different directions. The objective is to understand and measure the angles at which these signals depart from the transmitting device.

However, AoD implementations often requires complicated antenna arrays and beamforming techniques, adding complexity to the hardware. As well as other angle-based methods, AoD can be affected by changes in the environment, such as obstacles or reflections.

2.2.2 Distance-based methods

Distance-based methods in the context of indoor localization involve measuring the physical separation between two points, providing information about the distance or range between them. These methods play a crucial role in determining the position of objects or devices within an indoor environment. Three common distance-based methods are Time of Flight (ToF), Time Difference of Arrival (TDoA), and Two Way Ranging (TWR).

2.2.2.1 Time of Flight measurements

Time of Flight (ToF) is a distance-based measurement method commonly used in various applications, including indoor localization, ranging systems, and 3D imaging. ToF measures the time it takes for a signal to travel from a transmitter to a target and back to a receiver. This time measurement is typically done with high precision.

The distance between the transmitter and the target is calculated using the formula:

$$\text{Distance} = \frac{\text{Time of Flight} \times \text{Speed of Signal}}{2} \tag{2.1}$$

The division by 2 accounts for the round-trip travel of the signal.

By deploying multiple transmitters or receivers, the position of a device can be determined with high accuracy. Therefore, ToF is commonly used in indoor positioning and localization systems. ToF often requires synchronization between the transmitter and receiver to accurately measure the time of flight. It is commonly used in ultra-wideband (UWB) systems and can provide high-precision distance measurements.

Nevertheless, TOF method faces the challenge of synchronization issue. Achieving precise synchronization between the transmitter and receiver could be challenging, especially in environments with interference.

2.2.2.2 Time Difference of Arrival measurements

Time Difference of Arrival (TDoA) involves measuring the difference in arrival times of signals at multiple receivers or sensors to estimate the location of a transmitting device. The time difference is a key parameter used for localization. TDoA systems typically have a network of known reference points or anchors with precisely known locations. The signal source, or mobile device, is located based on the time differences observed at these reference points.

TDoA systems require a network of at least three receivers or sensors with known locations. These receivers are often strategically placed in the environment, forming a triangulation setup. The transmitting device emits a signal, and the time it takes for the signal to reach each receiver is recorded. The time differences between the arrivals at different receivers are then used to calculate the position of the signal source. TDoA systems use the time differences to create hyperbolic curves or surfaces in the time-space domain. The intersection points of these curves or surfaces represent possible locations of the signal source.

TDoA is particularly useful in scenarios where the precise time difference information can be used to determine the direction or position of the signal source. This system can achieve high precision in localization, especially in environments where the geometry of the reference points is well-defined.

Disadvantages of this method include clock synchronization problems since the TDoA system requires precise synchronization of clocks between receivers to accurately measure time differences. Besides, TDoA may be sensitive to multipath effects, where signals reflect off surfaces and reach the receivers through multiple paths.

2.2.2.3 Two Way Ranging measurements

Two Way Ranging (TWR) method works by measuring the time it takes for a signal to travel from a mobile device to an anchor node and back. This round-trip time is crucial for calculating the distance between the mobile device and the anchor. For accurate measurements, TWR systems require synchronization between the clocks of the mobile device and the anchor nodes. This synchronization ensures precise time measurements.

TWR systems consist of a network of anchor nodes with known locations. These nodes act as fixed reference points, and the mobile device interacts with them to determine its position. Then, the mobile device sends a signal to an anchor node, and the anchor node responds upon receiving the signal. The round-trip time of the signal is recorded. To determine the location of the mobile device, TWR is often performed with multiple anchor nodes. The distances to multiple anchor nodes are used to triangulate the position of the mobile device. The distance between the mobile device and the anchor node is calculated by multiplying the round-trip time by the speed of the signal. The formula is similar to ToF method:

$$\text{Distance} = \frac{\text{Round-Trip Time} \times \text{Speed of Signal}}{2} \tag{2.2}$$

TWR is commonly used in indoor localization systems, where the distances between a mobile device and multiple anchor nodes are used to determine the device's position with high accuracy. Moreover, the system could adapt to dynamic environments, where the positions of the mobile device and anchor nodes may change over time.

However, the TWR method also suffers from the clock synchronization problem, since achieving accurate synchronization between the clocks of the mobile device and anchor nodes is essential for precise distance measurements. Also, TWR systems may require complex hardware setups and communication protocols, especially in scenarios with a large number of anchor nodes.

2.2.3 Fingerprinting-based methods

Fingerprinting-based methods are a category of localization techniques that rely on creating a database or map of signal characteristics at known locations in an environment. These methods compare the measured signal characteristics of an unknown location with the database to determine the location. One of the common fingerprinting-based methods is Radio Signal Strength (RSS) Fingerprinting.

2.2.4 Radio Signal Strength fingerprinting measurements

Received Signal Strength (RSS) fingerprinting is a localization technique that relies on measuring and analyzing the strength of signals received from wireless transmitters, such as Wi-Fi access points or UWB (Ultra-Wideband) anchors. Different locations in the environment exhibit unique patterns of signal strength due to obstacles, reflections, and other environmental factors. These patterns are used to create distinctive fingerprints for each location.

During the calibration or training phase, a mobile device or a set of sensors is moved to predefined locations within the target area. At each location, the device measures the RSS from the available transmitters and records this information. Then, the collected RSS measurements at each location are used to construct a fingerprint database. Each entry in the database represents the characteristic RSS pattern associated with a specific location.

In the online phase, when the device needs to be localized, it measures the RSS from the available transmitters at its current location. The measured RSS values are compared with the fingerprints stored in the database. The system looks for the best match or similarity between the measured RSS pattern and the patterns in the database. The location of the device is estimated based on the closest match or matches in the fingerprint database. Techniques such as nearest neighbor algorithms or machine learning classifiers may be employed for this purpose.

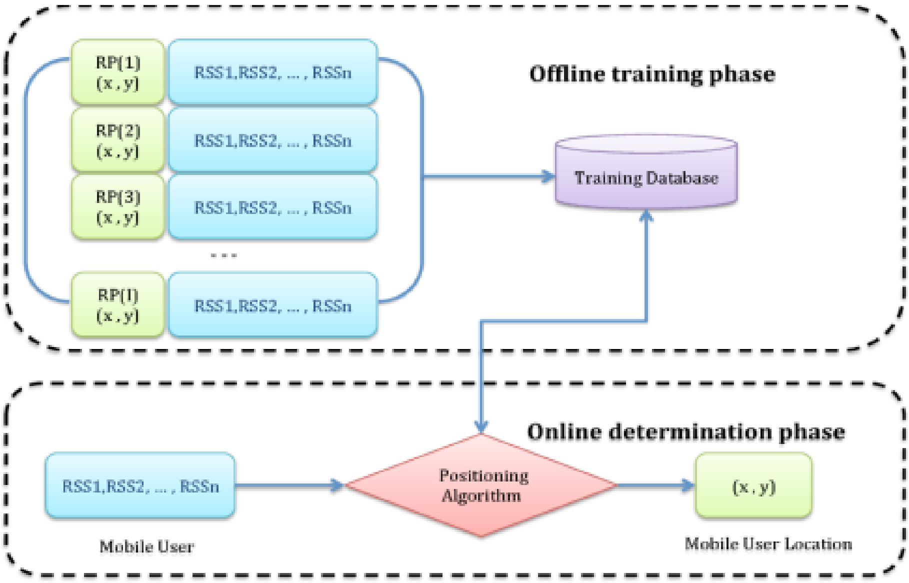

Figure 2.2: RSS Fingerprinting measurements

RSS fingerprinting is widely used in indoor positioning applications, such as guiding users within shopping malls, airports, or large buildings. Moreover, RSS fingerprinting can work with various types of wireless signals, including Wi-Fi, Bluetooth, or UWB. It is adaptable to different indoor environments, and the fingerprint database can be updated or expanded as needed. Also, RSS fingerprinting does not require specialized hardware infrastructure, making it a cost-effective solution for indoor localization.

However, this method could be affected by obstacles, interference, and other environmental factors, leading to variability in measurements. It also needs regular calibration and maintenance are necessary to update the fingerprint database, especially in dynamic environments.

2.3 Localization Algorithms

2.3.1 Trilateration

Ultra-Wideband (UWB) trilateration method for indoor localization is a technology that makes it possible to precisely locate and track items or persons in inside environments. High precision localization and distance measurement are made possible by UWB technology, which makes use of radio waves with an extremely broad bandwidth.

Trilateration is a geometric method that involves measuring the distances between an item and at least three known reference points, or anchors, that are known to be in certain locations. These anchors in indoor localization are usually UWB transceivers.

The trilateration algorithm calculates the coordinates (x, y, and sometimes z in three-dimensional spaces) of an object or device based on distance measurements to three or more known reference points (anchors).

Let (x, y, z) be the coordinates of the unknown point (the device to be localized).

Let (x_i, y_i, z_i) be the coordinates of the i-th anchor.

Let (d_i) be the measured distance from the unknown point to the i-th anchor.

The trilateration algorithm can be expressed using the following equations:

For 2D trilateration (x and y coordinates):

$$\begin{aligned}
(x - x_1)^2 + (y - y_1)^2 &= d_1^2 \\
(x - x_2)^2 + (y - y_2)^2 &= d_2^2 \\
(x - x_3)^2 + (y - y_3)^2 &= d_3^2
\end{aligned} \tag{2.3}$$

[0]https://www.researchgate.net/publication/344380057

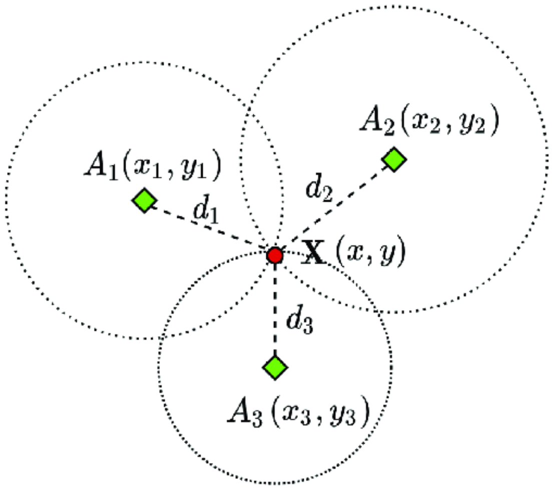

Figure 2.3: Example of 2D Trilateration Method

For 3D trilateration (x, y, and z coordinates):

$$
\begin{aligned}
(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= d_1^2 \\
(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= d_2^2 \\
(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= d_3^2
\end{aligned}
\tag{2.4}
$$

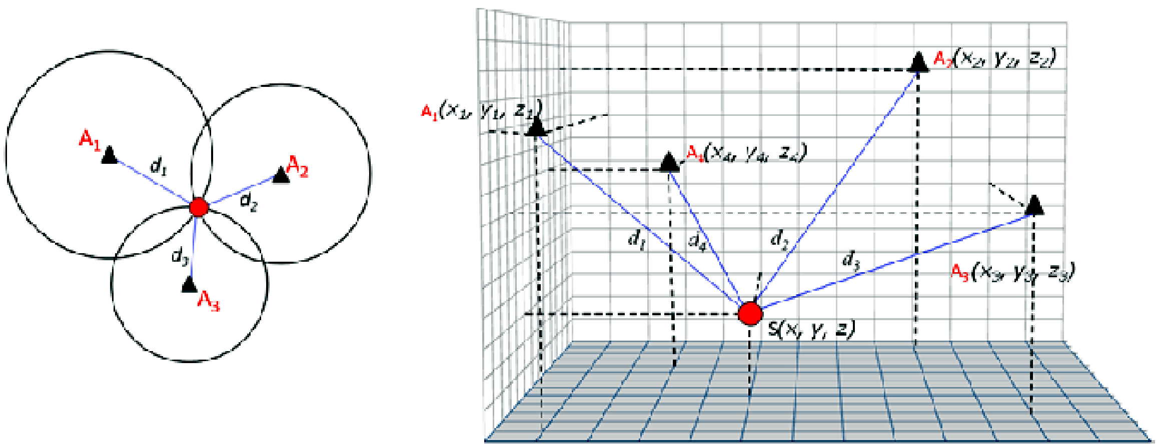

Figure 2.4: Example of 3D Trilateration Method

2.3.2 Triangulation

Similar to trilateration, which likewise uses signal or transmission time delay to determine distance, is the triangulation method. To compute the angle and distance measurements to the object to be determined, however, this approach simply requires a minimum of two reference sites.

[0]https://tinyurl.com/4thec24f

The following formula, where H is the intersection point of the straight line between the two reference locations and the object's perpendicular projection line, calculates the distance between the object and the straight line:

$$d = \frac{H \sin\alpha \sin\beta}{\sin(\alpha + \beta)} \tag{2.5}$$

The primary determinant of both trilateration and triangulation techniques is the signal that the access points transmit. In actuality, environmental factors may readily disrupt radio signals, resulting in large mistakes, and the provided formulae are typically limited to the best-case scenarios when it comes to precise distance estimation. The fingerprint signal approach is another widely used, very accurate location technique nowadays.

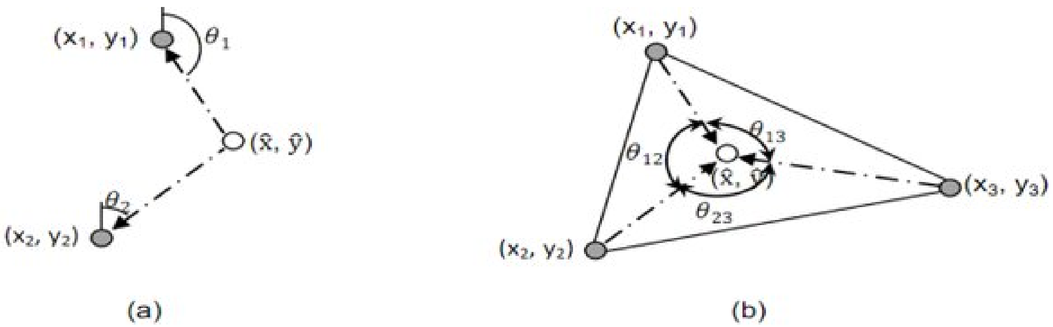

Figure 2.5: Triangulation algorithm

[0]https://www.researchgate.net/figure/Triangulation-localization-method_fig5_316960120

CHAPTER 3

PROPOSED SOLUTIONS

Based on our research, including background knowledge and experiments, we propose a solution for developing a UWB indoor localization system. This system consists of several UWB modules with fixed position as anchors and one UWB module functioning as the tag or target of the localization process.

We begin by explaining the calibration of the anchors to ensure their precise placement and functionality. Next, we detail the method for measuring the distances between each anchor and the tag. Finally, we discuss our selected algorithm for determining the target's location.

Each stage of the process includes an explanation of our methodological choices, supported by the results we achieved through our implementation.

3.1 Setup Anchors by Calibrating

Calibrating UWB (Ultra-Wideband) anchor devices is a critical step in the process of setting up a localization system. Calibration involves determining the precise location of each anchor device in the coordinate system that will be used for positioning calculations. The level of calibration accuracy directly impacts the accuracy of the UWB localization system.

As this is an indoor localization system, relying on GPS signals is neither practical nor sufficiently accurate, so it is necessary to implement a local coordinate system. This involves defining a coordinate system specific to the environment where the UWB system will operate. The origin (0,0,0) and orientation of this local coordinate system are determined, and each UWB anchor device is assigned coordinates within this system. Then, install each UWB anchor device in its designated location. Next, assign a unique identifier (ID) to each UWB anchor device. This ID will be used to distinguish between anchors during the trilateration process. Because we used equipment provided by Qorvo, all the modules already possess an ID. Finally, update the configuration settings of the UWB localization system with the calibrated coordinates and IDs of the anchor devices.

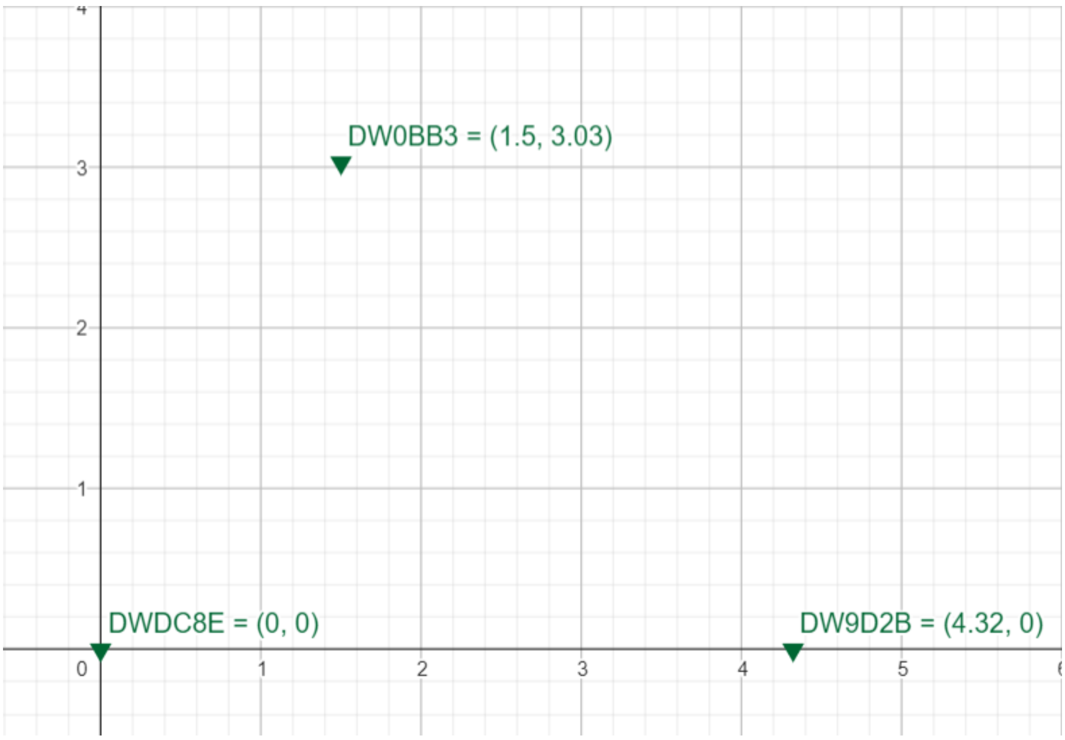

Figure 3.1: Set up static Anchors

3.2 Measure Distances using Two Way Ranging

According to Qorvo, Ultra-wideband (UWB) is a radio technology based on the IEEE 802.15.4a and 802.15.4z standards that can enable the very accurate measure of the Time of Flight (ToF) of the radio signal, leading to centimeter accuracy distance/location measurement. Therefore, we integrate the ToF method into our devices to determine the distance between tags and anchors. Subsequently, we compare these outcomes with real-life distances to assess the accuracy of the device.

First, set up modules as initiator and responder in a Symmetric Double-Sided Two-Way Ranging (SS TWR) distance measurement exchange.

The initiator sends a "poll"frame (recording the transmission (TX) time-stamp of the poll), after which it waits for a "response"message from the responder to complete the exchange. The response message contains the remote responder's time-stamps of poll reception (RX), and response TX. Using these time-stamps along with the local time-stamps of the poll TX and response RX, the system calculates the ToF, then calculates the distance between devices.

3.3 Implement Trilateration Algorithm

The choice of employing the trilateration algorithm for UWB indoor localization stems from its inherent suitability for the unique challenges posed by indoor environments. Trilateration excels in scenarios where precise location determination is crucial, especially within confined spaces where signals may encounter obstructions and reflections. Unlike other algorithms that might struggle with multipath interference or signal attenuation in complex indoor settings, trilateration leverages the ToF measurements from multiple UWB anchors to accurately pinpoint the device's location. This algorithm's robustness in mitigating the effects of signal distortion and its ability to provide accurate three-dimensional coordinates make it a preferred choice for UWB indoor localization applications, ensuring reliable and precise positioning in dynamic indoor environments.

To implement the algorithm, we first compile a dataset comprising coordinates and corresponding distance measurements obtained through the above two steps. Subsequently, we employ the Python programming language and NumPy library to implement the algorithm. Through this implementation, our objective is to deduce the accurate coordinates of the tag based on the recorded data, allowing us to precisely determine its position within the designated space. The trilateration graph presented below illustrates the coordinates obtained through the applied algorithm. Upon examination of the graph, it is apparent that the implementation has produced positive results, indicating accuracy in the outcomes.

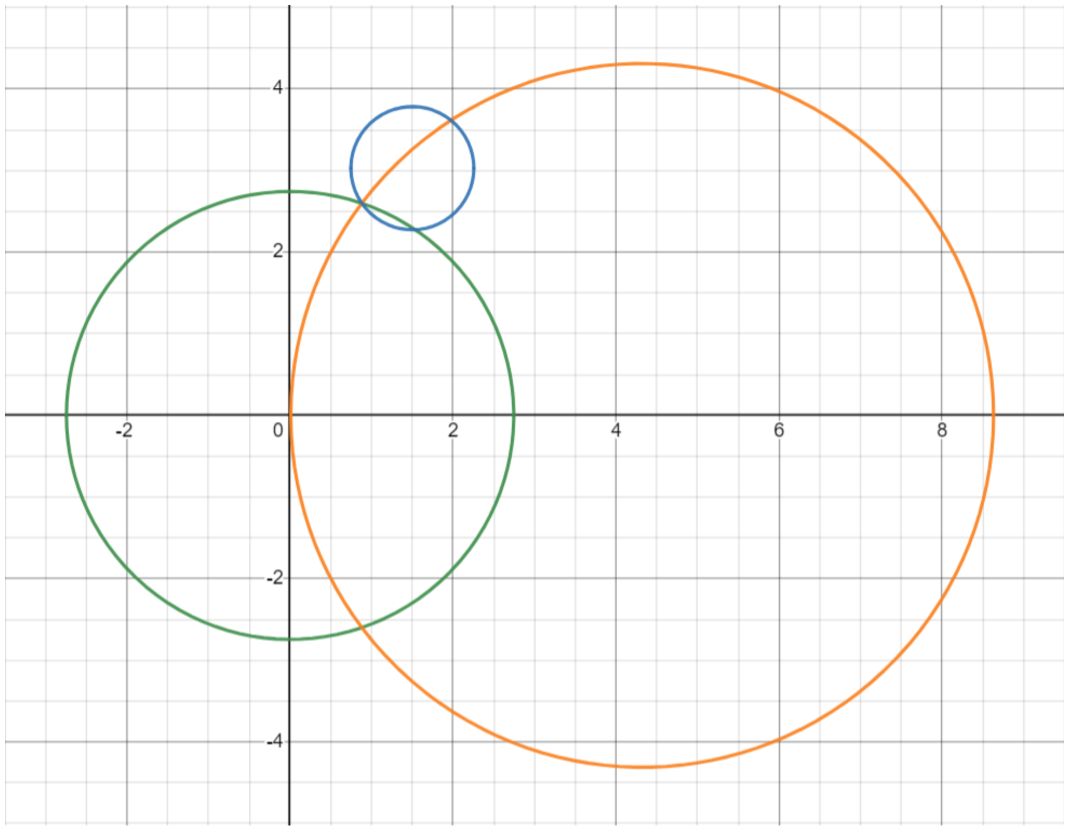

Figure 3.2: Trilateration graph

Subsequently, we create a graph that integrates the coordinates of anchors and tags, allowing us to visually represent their spatial relationships. Through this graphical representation, we can observe and verify the distance results aligned with the implemented algorithm.

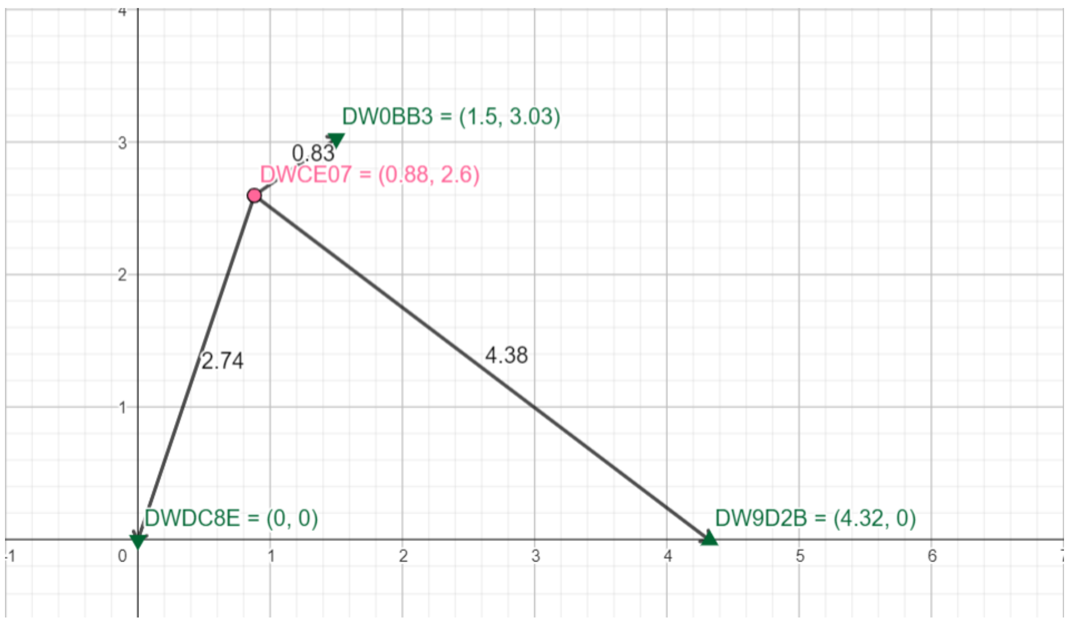

Figure 3.3: Trilateration point graph

CHAPTER 4

SYSTEM ARCHITECTURE AND IMPLEMENTATION

4.1 System Architecture

After selecting the methods to calculate the position of our target, the next step is to develop a system that integrates our localization solution into an application. To achieve this, we have designed a system architecture consisting of four main blocks:

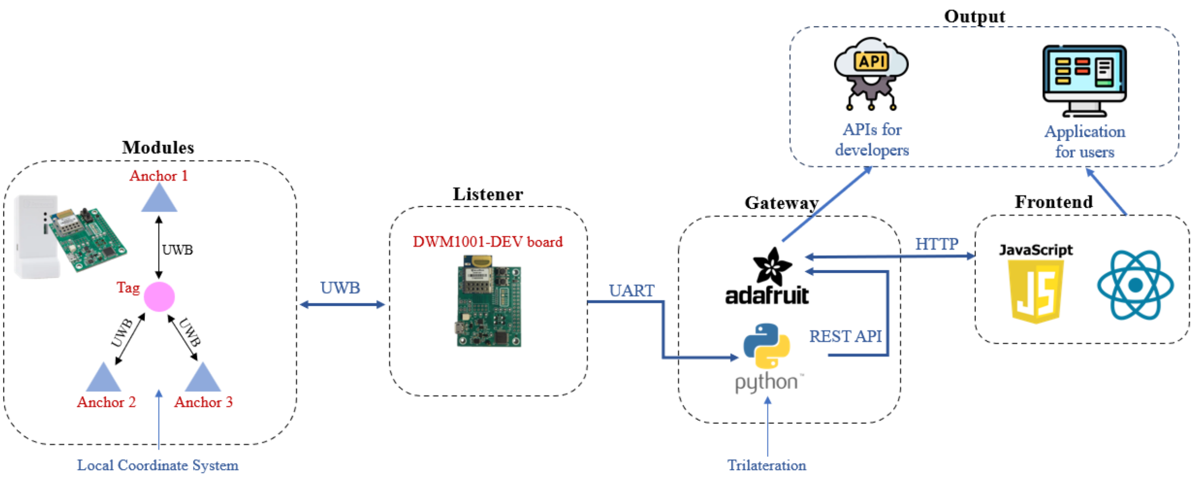

Figure 4.1: System architecture

The system consists of five main basic components:

- **Modules:** The system uses devices in Qorvo's MDEK1001 kit. There are two types of devices involved: "Anchor" and "Tag". The "Anchor" devices are stationary and serve as reference points, while the "Tag" device is mobile whose position needs to be determined.
- **Listener:** One DWM1001-DEV board functions as a bridge within the

UWB system. This device acts as a listener, supplying data to our system and eliminating the need to directly connect the tag to our PC.

- **Gateway:** Connected the listener to our PC via UART, used Python to read and filter the necessary data, implemented trilateration to position our target, and sent the data to Adafruit.io using Adafruit's REST API.
- **Front-End:** Used HTTP method to take data from the Adafruit database into our FrontEnd. Then, Develop UI using React framework.
- **Output:** Our two final outputs are the software to visualize our UWB system and the APIs of our UWB system for other developers.

4.2 Implement the System

In this section, we will explain into the details of how we implement each block of our system architecture, covering both software and hardware components. We will also explain why we chose this particular implementation approach for each block.

4.2.1 Modules Block

4.2.1.1 Required Components

Hardware:

- 4 DWM1001-DEV boards
- Lithium batteries or Power banks

Software:

- DRTLS Android app by Qorvo
- Tera Term as SSH clients to communicate with the modules

4.2.1.2 Implementation

In this part of our system, we require several UWB modules to function as anchors and a tag. For our configuration, we have chosen three modules to serve as anchors and one module to act as the tag. We've installed batteries or power banks on these boards to provide power and mounted them to walls, as illustrated in the image below.

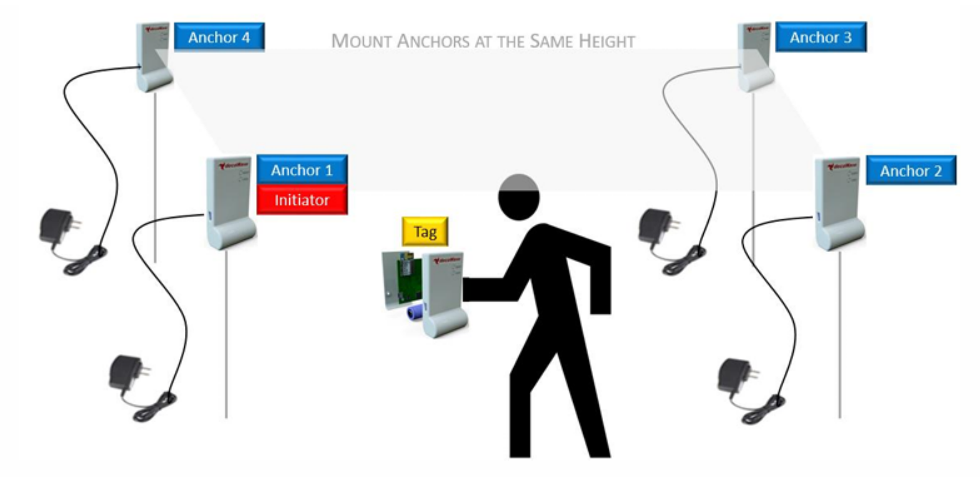

Figure 4.2: Positioning of Anchors and Tags

Configuration of data for both anchors and tag can be done in two ways: through Qorvo's Android app or via your PC.

System Configuration by using the DRTLS Android manager

Create a network and set up three anchors and one tag:

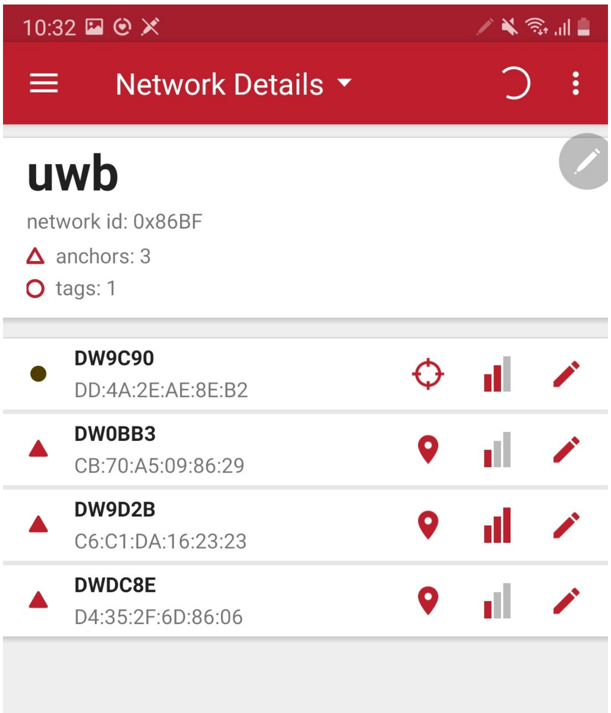

Figure 4.3: Network

Configure for an anchor, set the UWB as "active"and click on "initiator"block to show the coordinates of anchor. In there, we also could able

[0] https://www.qorvo.com/products/p/MDEK1001#documents

to change the anchor coordinates

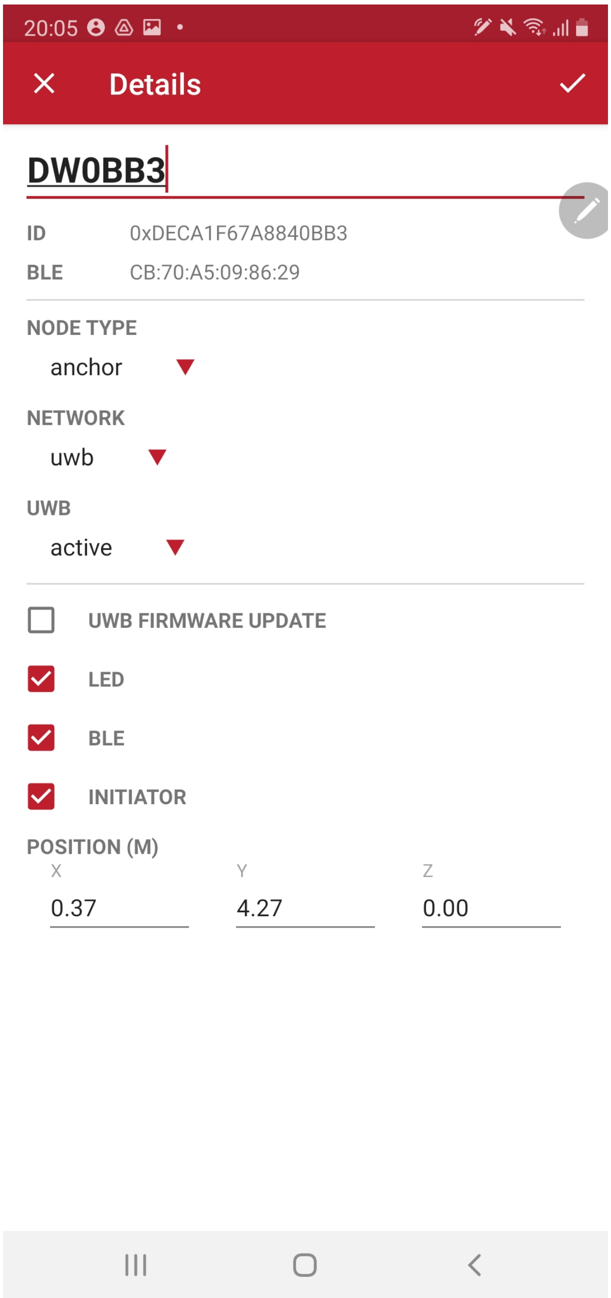

Figure 4.4: Configure for anchor

Configure for a tag, set the UWB as "active"and click on "responsive mode"along with "location engine"blocks

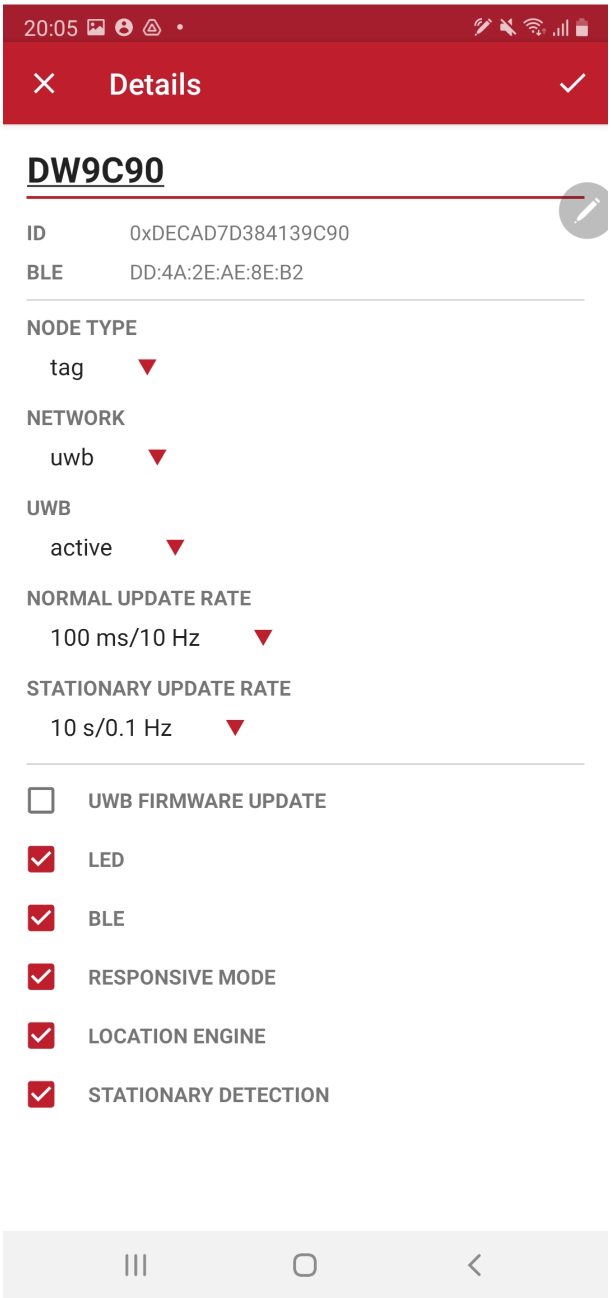

Figure 4.5: Configure for tag

Then, measure the anchors coordinate:

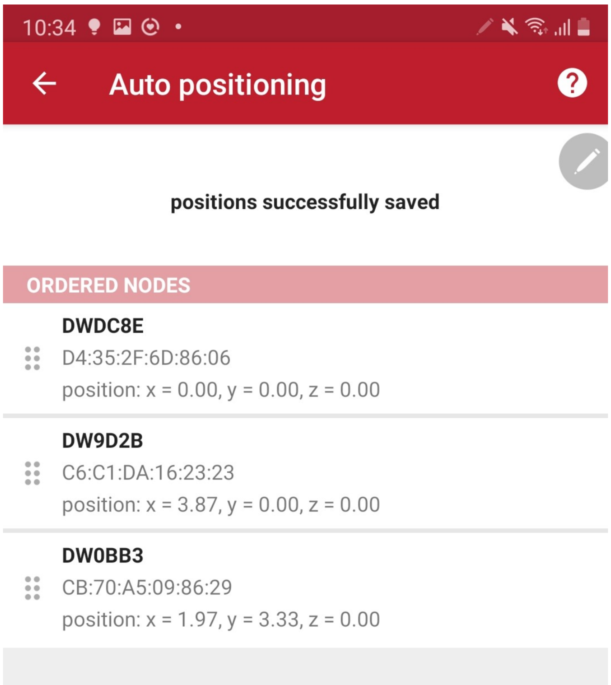

Figure 4.6: Measure anchors

Proceed to the grid view to see the map:

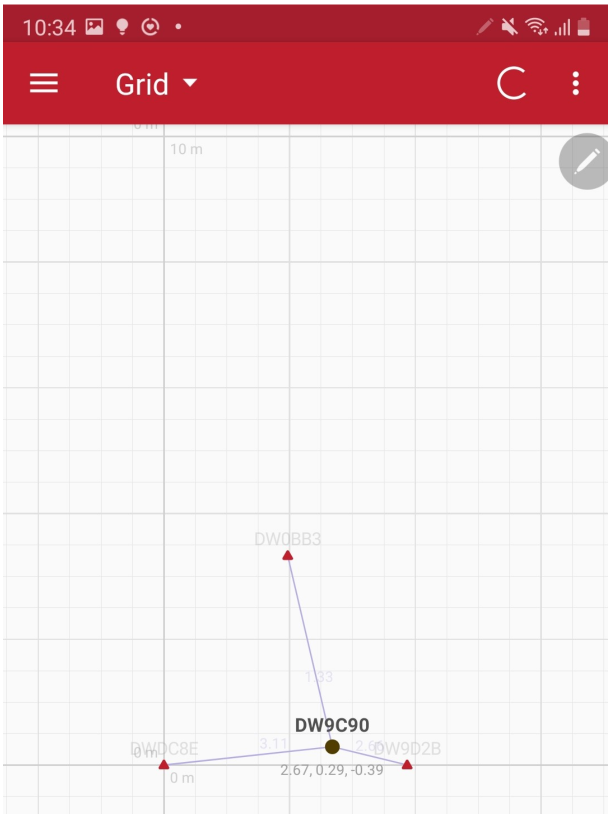

Figure 4.7: View map

System Configuration by using the UART shell mode

Using Tera Term, you can read and configure nodes directly over UART. Connect the DWM1001-DEV to the PC via USB and launch TeraTerm. Select connect by "Serial"and choose the COM port connected to the device:

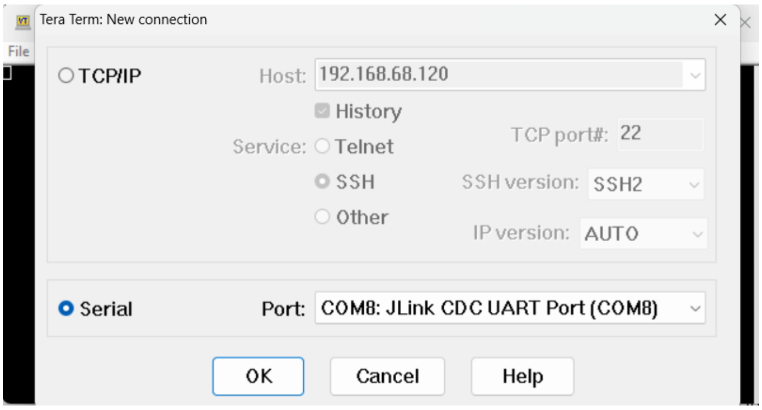

Figure 4.8: Connect by UART

Click on the "Setup"tab and select "Serial Port", then change the speed to *115200*:

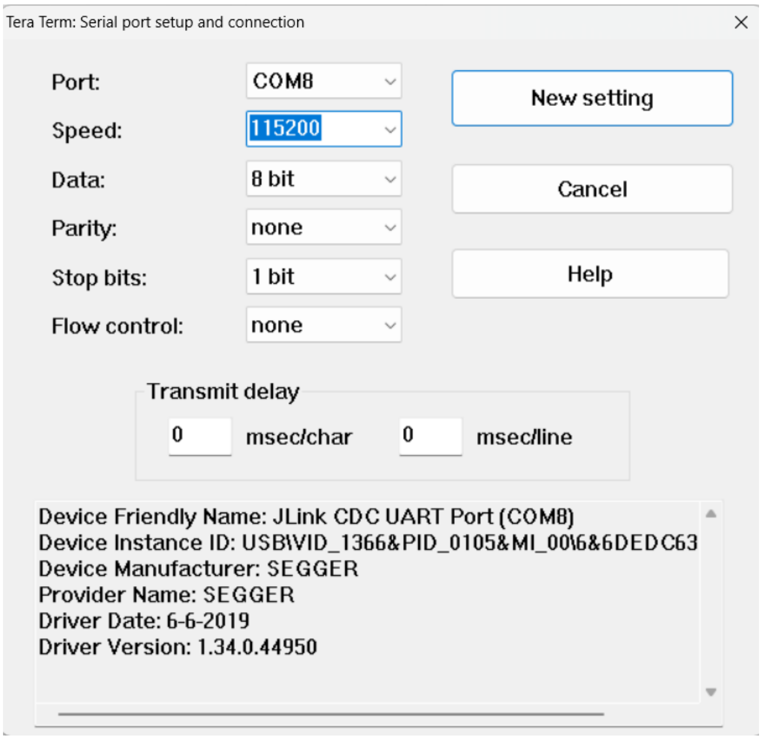

Figure 4.9: Setup Serial port

In the shell, press the "Enter"key twice to start the DWM1001 UART shell mode. Then configure the anchor using the following commands:

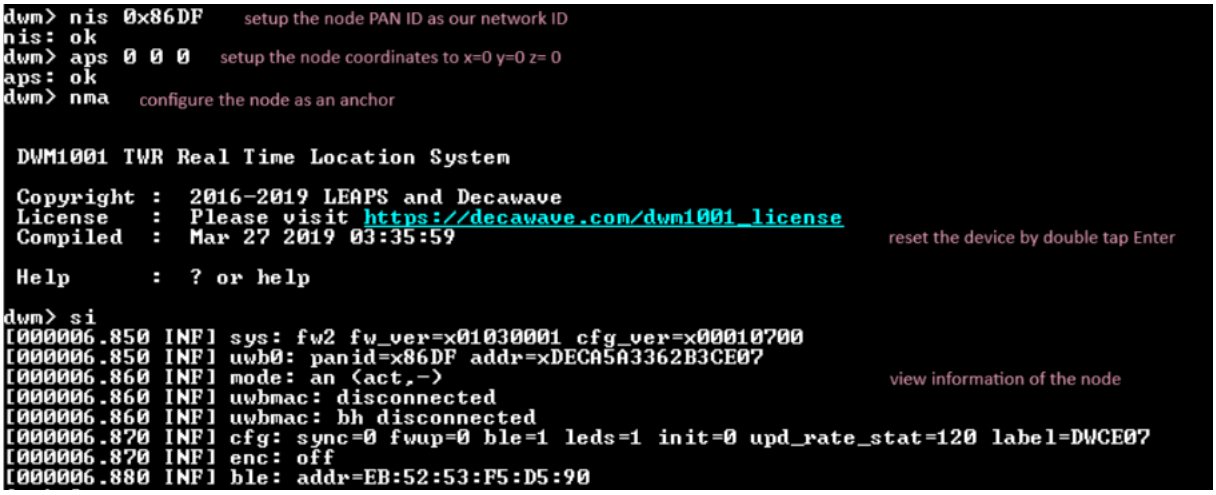

Figure 4.10: Configure the anchor

4.2.2 Listener Block

4.2.2.1 Required Components

Hardware:

- 1 DWM1001-DEV board
- Lithium battery or Power bank

Software:

- DRTLS Android app by Qorvo
- Tera Term as SSH clients to communicate with the modules

4.2.2.2 Implementation

This component acts as a bridge node connecting our UWB modules to the database. This device listens to all the data transmitted by the system without altering it. We opted to connect the modules directly to our PC via UART rather than using Bluetooth or WiFi. This choice enhances the robustness of the signal transmission, ensuring more reliable and stable communication within our system.

To configure the module as a listener, there are two methods you can employ:

Using the DRTLS Android app: Set the module as a tag, but switch the UWB mode to "passive."This adjustment allows the module to operate as a listener without actively participating in location tracking.

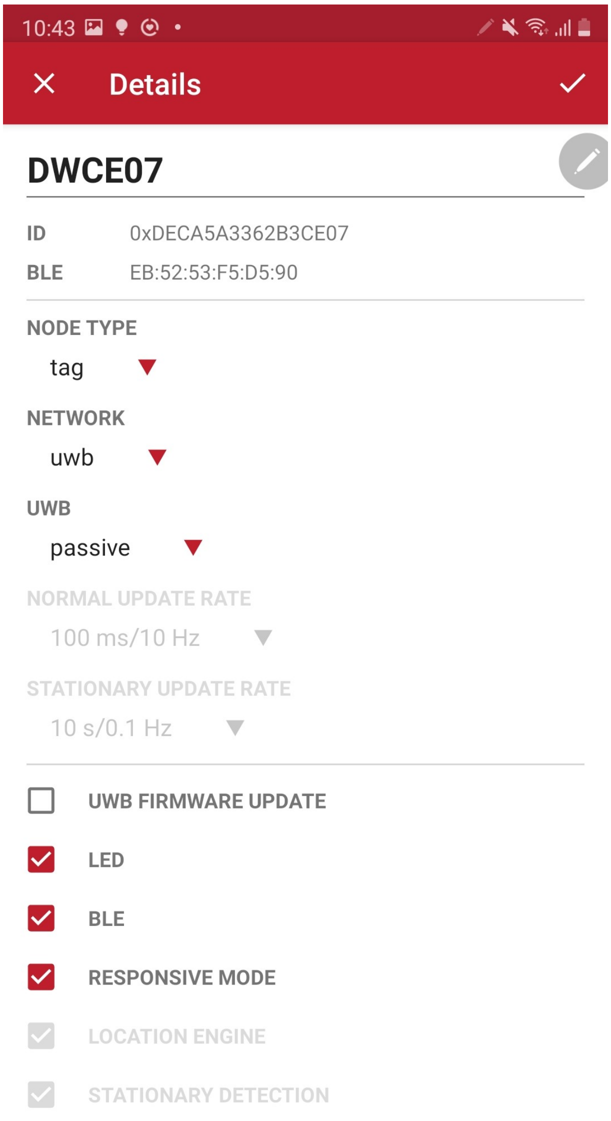

Figure 4.11: Set a tag with UWB as passive to become a listener node

Using the UART shell mode: Connect the module to a PC and use Tera Term, setup as previous step. Then, set the module to function as a bridge by entering the command "nmb."

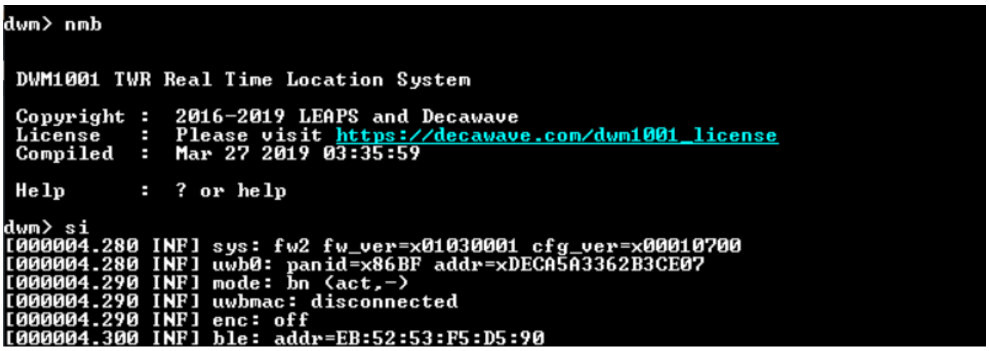

Figure 4.12: The mode of module is set to bridge

4.2.3 Gateway Block

4.2.3.1 Required Components

Software:

- **Python:** a modern object-oriented programming language that provides powerful typing and extensive built-in data structures. Python's support for packages and modules promotes code reuse and modularity. Its comprehensive standard library and freely available interpreter make it accessible on all major platforms. Key features of Python for backend programming include:
  - Scalability: Python's support for multiprocessing and threading, along with frameworks like Django, makes it suitable for building scalable backend systems.
  - Integration Capabilities: Python excels at integrating with other systems and services through its extensive libraries and modules, enabling easy interaction with APIs, web services, messaging queues, and various protocols like HTTP, FTP, and SMTP.
  - Database Access: Python supports various databases, including MySQL, PostgreSQL, SQLite, and NoSQL databases like MongoDB. Its database APIs, such as SQLAlchemy and Django ORM, simplify database interactions.

  In this project, Python is used to receive data from devices and send it to Adafruit.io via UART and MQTT. The libraries needed are: adafruit-io, keras, tensorflow, numpy, opencv-python, pyserial

- **Adafruit.io:** offered by Adafruit, is a cloud service tailored for Internet of Things (IoT) projects, providing data storage, real-time monitoring, and device control functionalities. It serves as a platform for storing and accessing IoT device data via the internet. While resembling a database with its storage capabilities, Adafruit.io is more accurately categorized as an IoT platform or cloud service. Key features of Adafruit.io include:
  - Data Logging: Log and store data from IoT devices for analysis.
  - Real-time Data Monitoring: Monitor device data in real-time with

live updates and visualizations.

– Device Control: Remotely control IoT devices by sending commands from the platform.

– Integration: Supports integration with other IoT platforms like IFTTT for automation.

– APIs: Provides MQTT and RESTful APIs for seamless data communication.

– Security: Emphasizes security with secure communication protocols and authentication mechanisms.

Adafruit.io serves as both a database for storing device data and a means to send data to web applications via HTTP (using Axios).

4.2.3.2 Implementation

First, we create a new dashboard in Adafruit.IO, named "UWB". Then, we create feeds corresponding to the number of modules used. In this case, we use 4 modules, so we create 4 feeds: 3 for anchors and 1 for the tag.

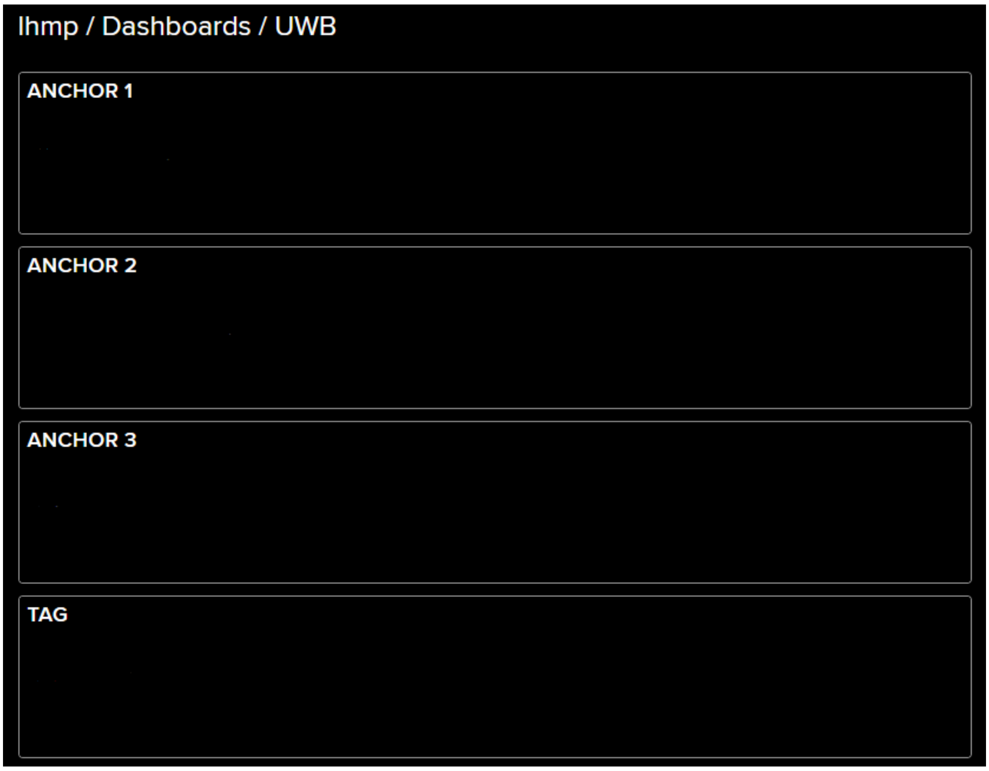

Figure 4.13: UWB dashboard

Next, we write Python scripts to perform the following tasks:

- Connect to Feeds: Use the Adafruit.IO username and key to connect to each feed.
- Connect to Serial: Select the COM port to which our modules are connected.
- Read Data from Serial: Continuously read data from the serial connection.
- Process Data: Split the data into strings to extract the necessary information, such as module ID, coordinates, and the distance to the tag (for anchors).
- Publish Data: If the extracted data is in the correct format, publish it to the corresponding feed.

When the data is published successfully, we can view it on our feed as shown in the image below:

Figure 4.14: The Data on Database

4.2.4 Front-End Block

4.2.4.1 Required Components

Technology:

- **Javascript:** a versatile and widely-used programming language specifically designed for creating interactive and dynamic web applications. JavaScript enables seamless integration with HTML and CSS, allowing for the development of rich user interfaces. Its extensive ecosystem, including libraries and frameworks like React, facilitates efficient and scalable app development. Additionally, JavaScript's ability to run on both the client and server sides (via Node.js) enhances performance and enables full-stack development, making it an ideal choice for modern web applications.

- **React:** we chose the React framework to develop our web application due to its efficiency and flexibility in building user interfaces. React's component-based architecture allows for reusable, modular code, which simplifies the development and maintenance of complex UIs. Its virtual DOM enhances performance by minimizing direct manipulation of the actual DOM, leading to faster rendering and a more responsive user experience. React's strong community support, extensive ecosystem of tools and libraries, and seamless integration with other technologies make it an ideal choice for developing scalable and maintainable web applications.

4.2.4.2 Implementation

We retrieve data from Adafruit.io into our application using the Axios library, with the API links for each feed. Next, we implement Trilateration algorithm to process this data and determine the location of the tag. Then, we used the React framework to build a user-friendly UI, featuring three main components:

- Location Mapping: This feature takes module data, calculate the tag and visualize the result in a grid map.
- Add Floor Plan: Users can integrate the floor plan of their indoor space into the map for enhanced visualization.
- Manage Network: Access and modify network components within our application.

The flow of each feature is described in the section below.

***Location Mapping:**

Go to "My Network,"where you will find a scan button for detecting UWB modules within the network:

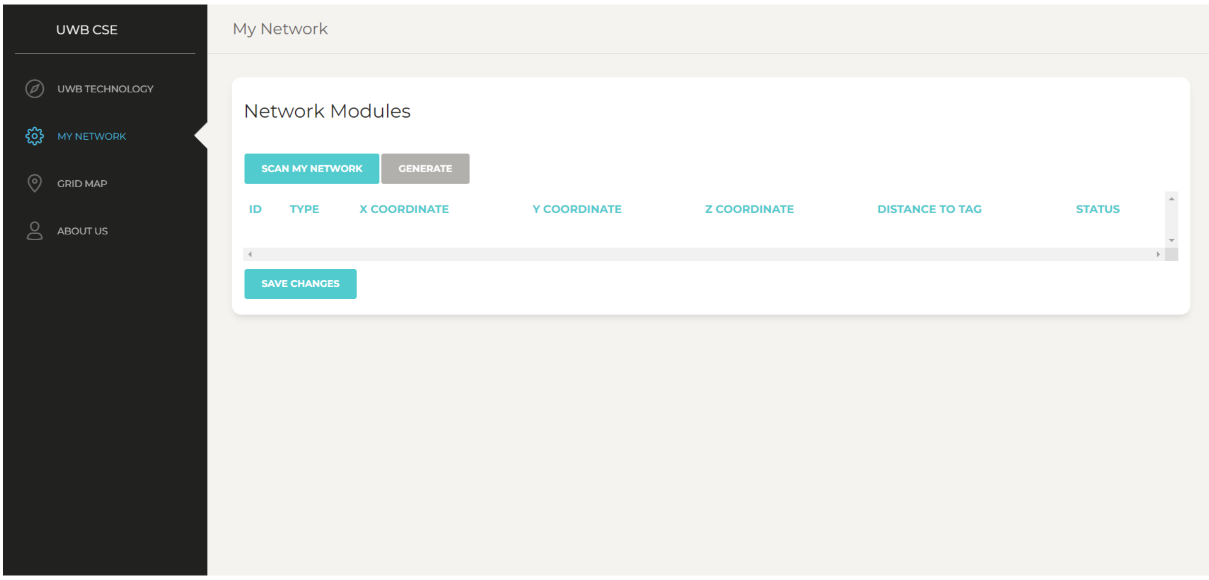

Figure 4.15: My Network view

After scanning, the table will display the modules that are active in the network:

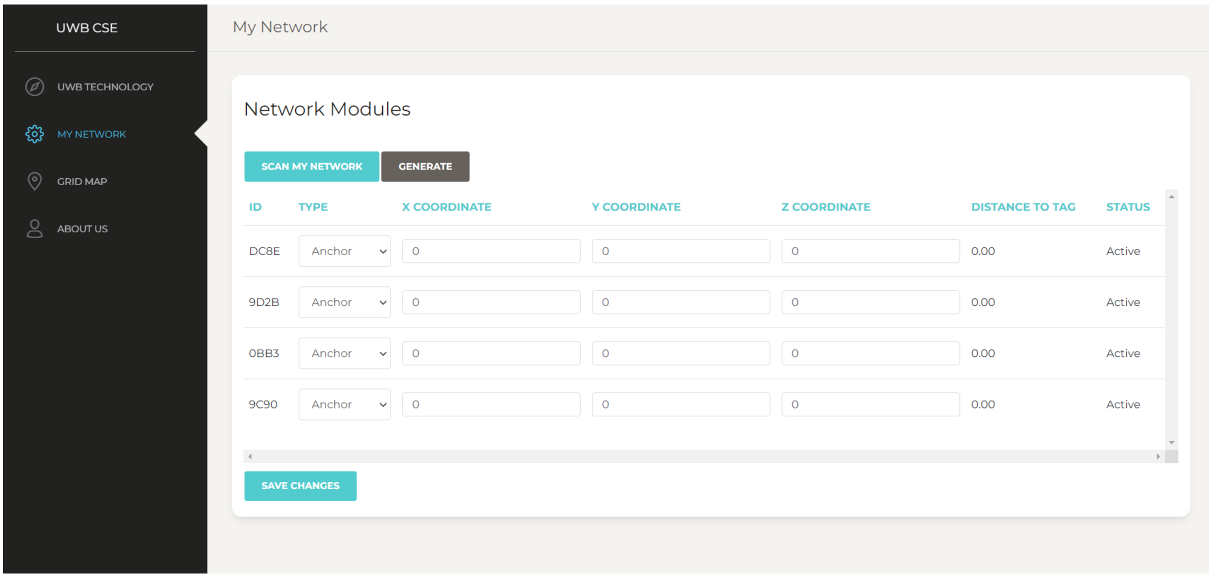

Figure 4.16: Scan results

Choose the type of modules — either tags or anchors. Then, click the generate button to generate the coordinates and the distances from the anchors to the tag:

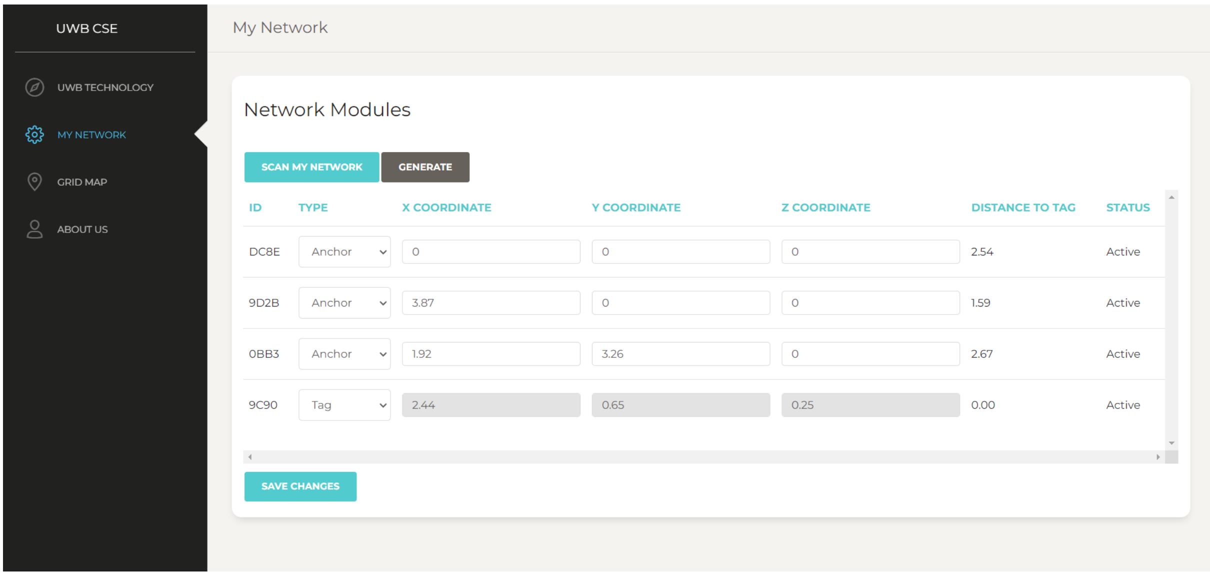

Figure 4.17: Generate results

Finally, click the save button to save the network:

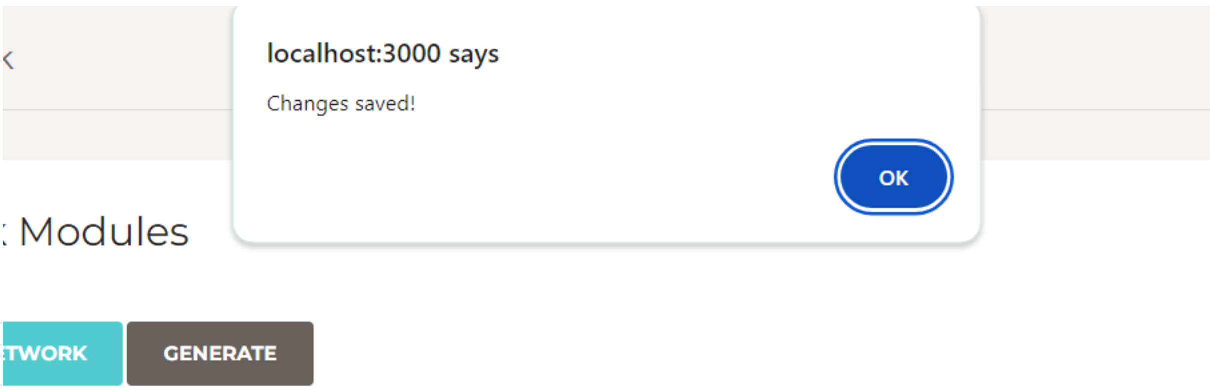

Figure 4.18: Successfully saved

Finally, go to the Grid Map view to see the map generated from the network's data:

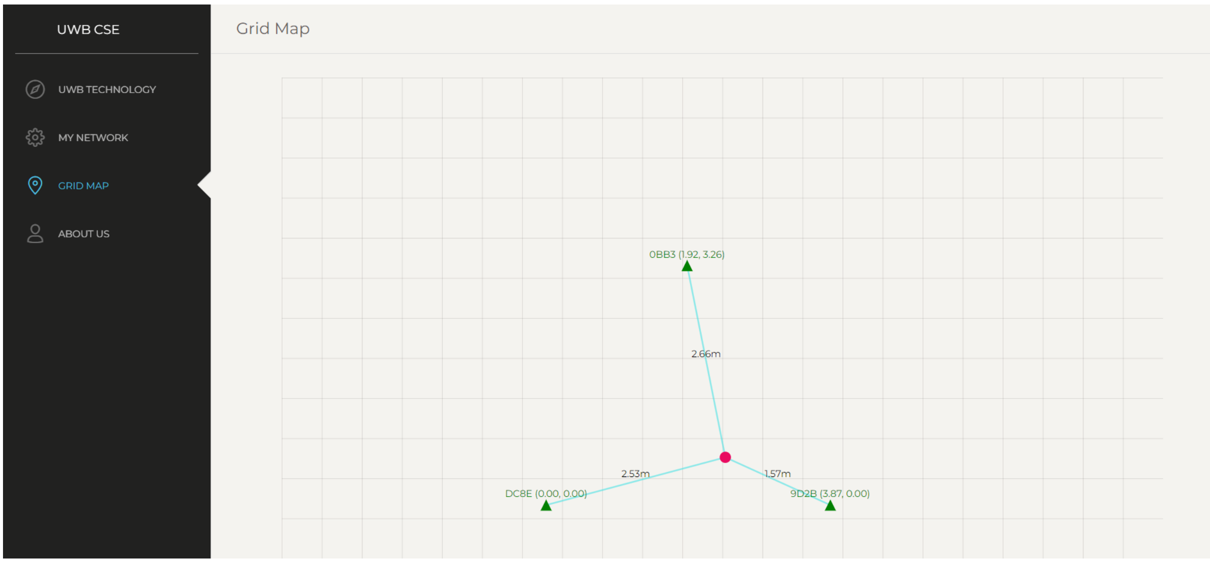

Figure 4.19: View the Map

*Add Floor plan:

In the Grid Map view, there is a button to add a floor plan:

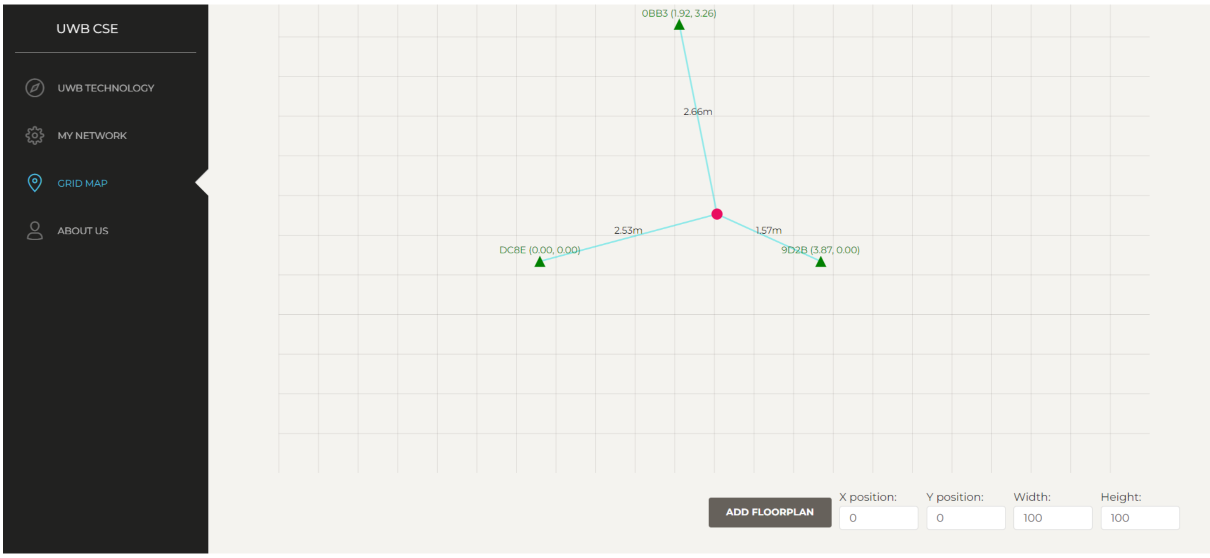

Figure 4.20: Add floor plan

Upload the floor plan, then edit the XY coordinates and the size to match your environment:

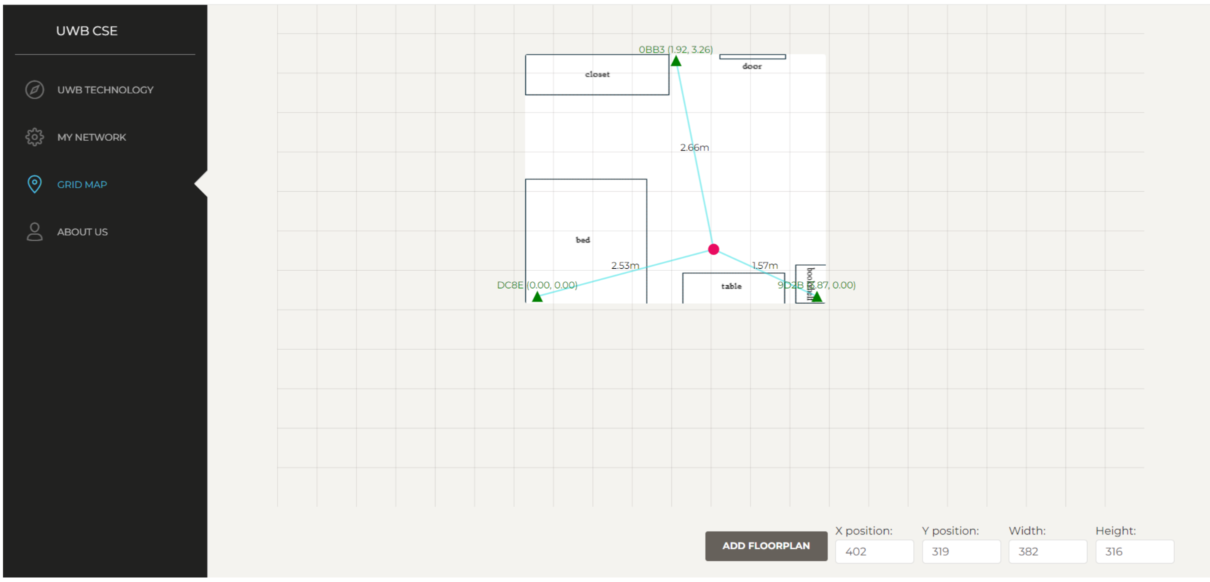

Figure 4.21: Add floor plan successfully

*Manage Network:

You can manage your network by choosing the type of modules and edit their coordinates.

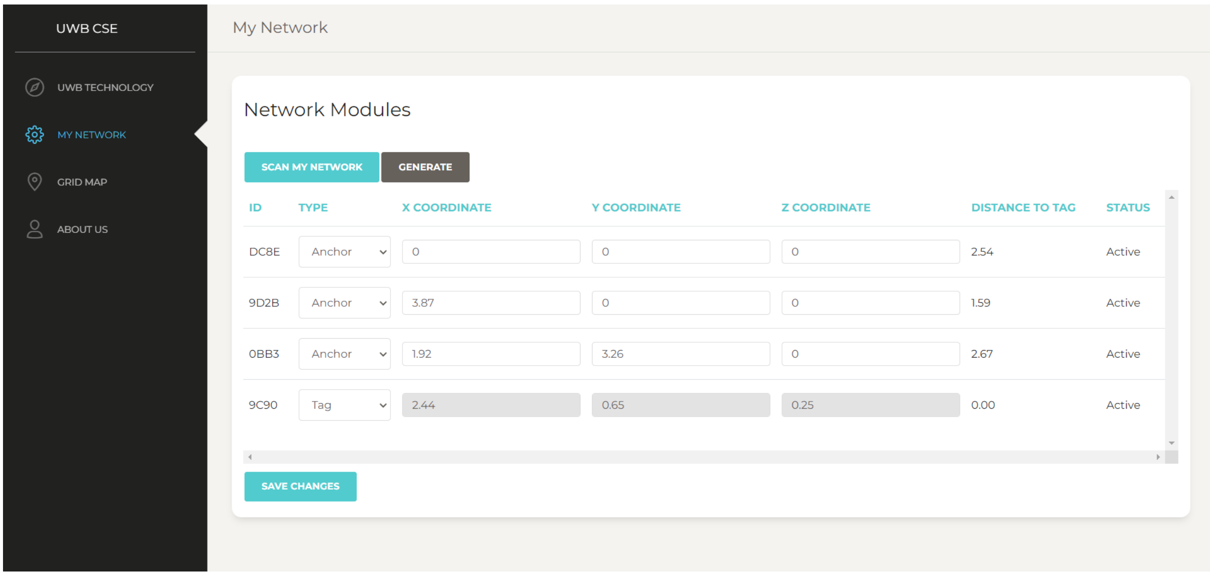

Figure 4.22: Data before editing

If you choose anchors, you can edit their coordinates. If you choose tags, the coordinate is locked. After changing the coordinates of the anchors, the system will automatically recalculate the distances to the tags:

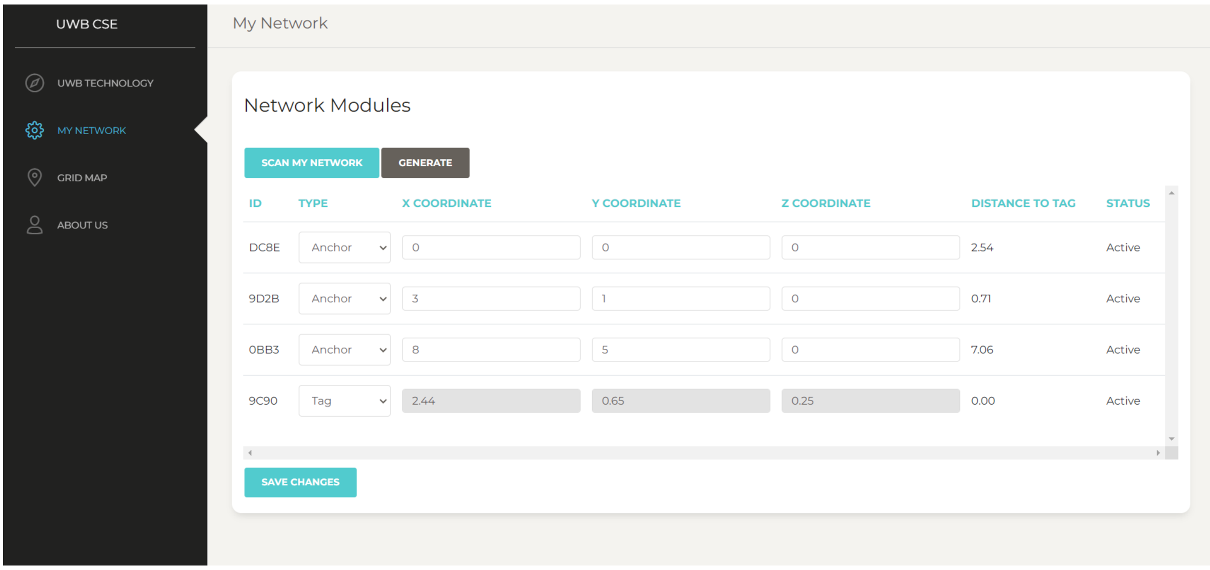

Figure 4.23: Data after editing

Then, click the save button to save your changes:

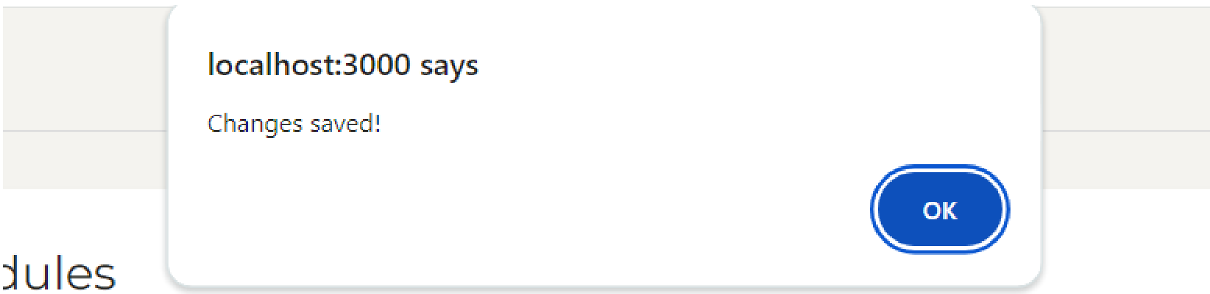

Figure 4.24: Successfully saved

4.3 Launch the System

Set up devices:

Mount the anchors on the wall. Connect the listener to the PC. Open Tera Term to check if the PC is connected to the listener node:

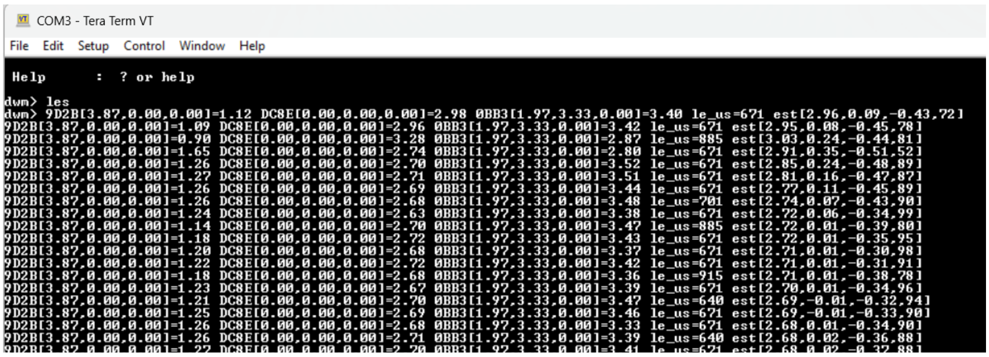

Figure 4.25: Listener node is connected

Launch the Application:

Access the application folder within your IDE, in this case, we used Visual Studio Code. Next, open a terminal and execute **npm i** to install all packages needed. Finally, run **npm start** to launch the application.

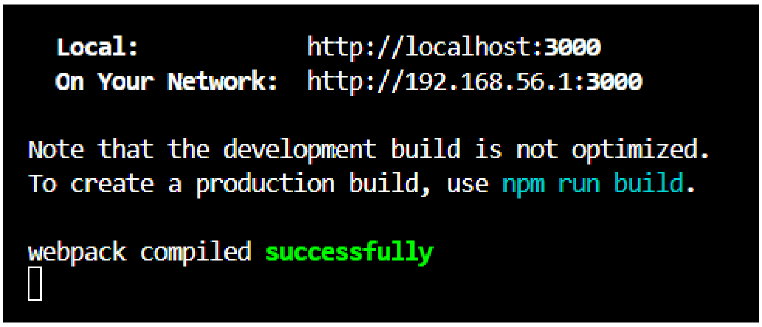

Figure 4.26: Successfully launched

View the Application in your localhost:

Open your browser and navigate to the link **http://localhost:3000/** to view the app.

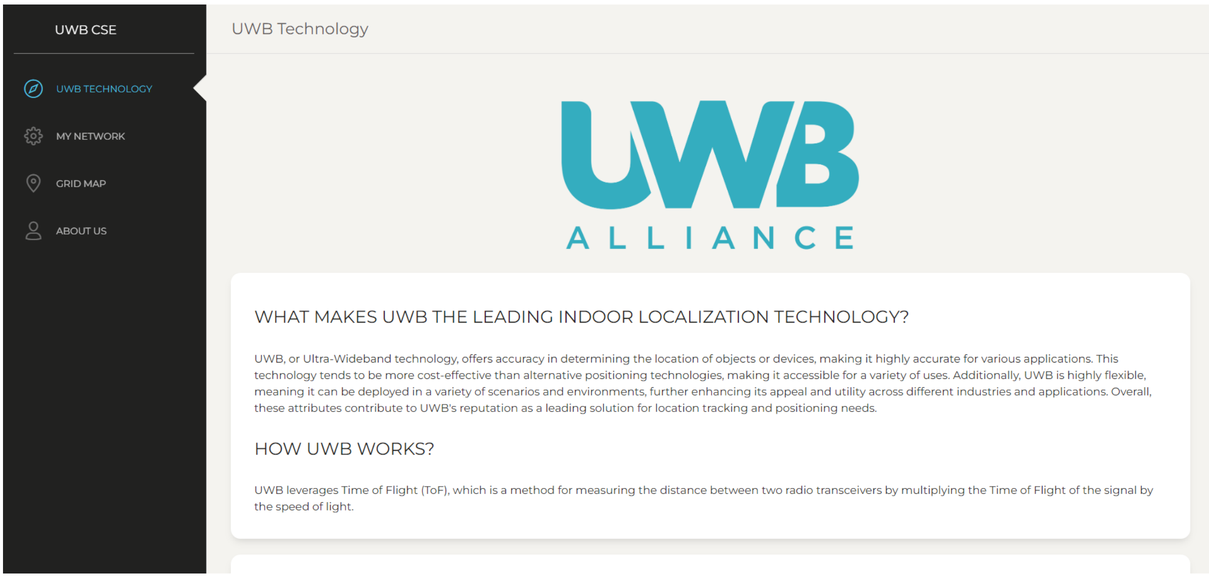

Figure 4.27: UWB Technology view

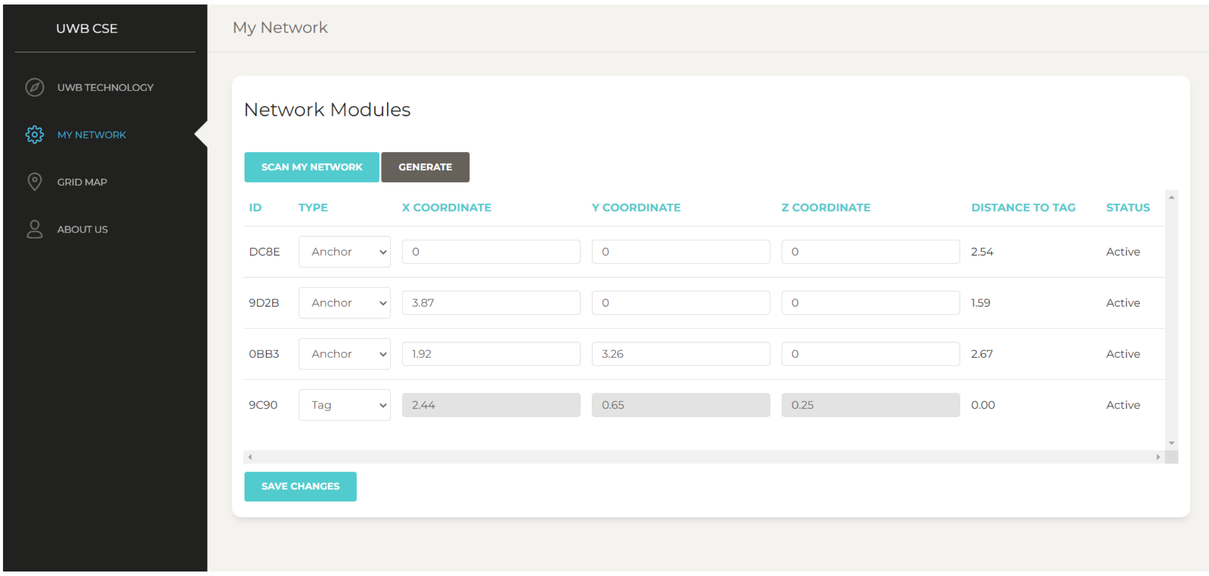

Figure 4.28: My Network view

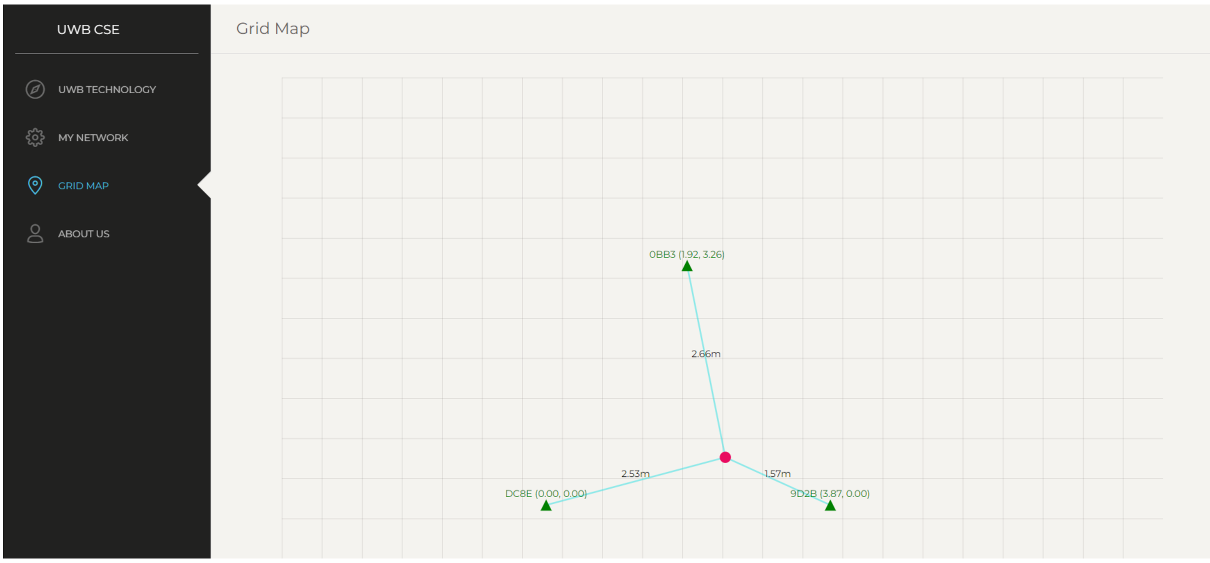

Figure 4.29: Grid Map view

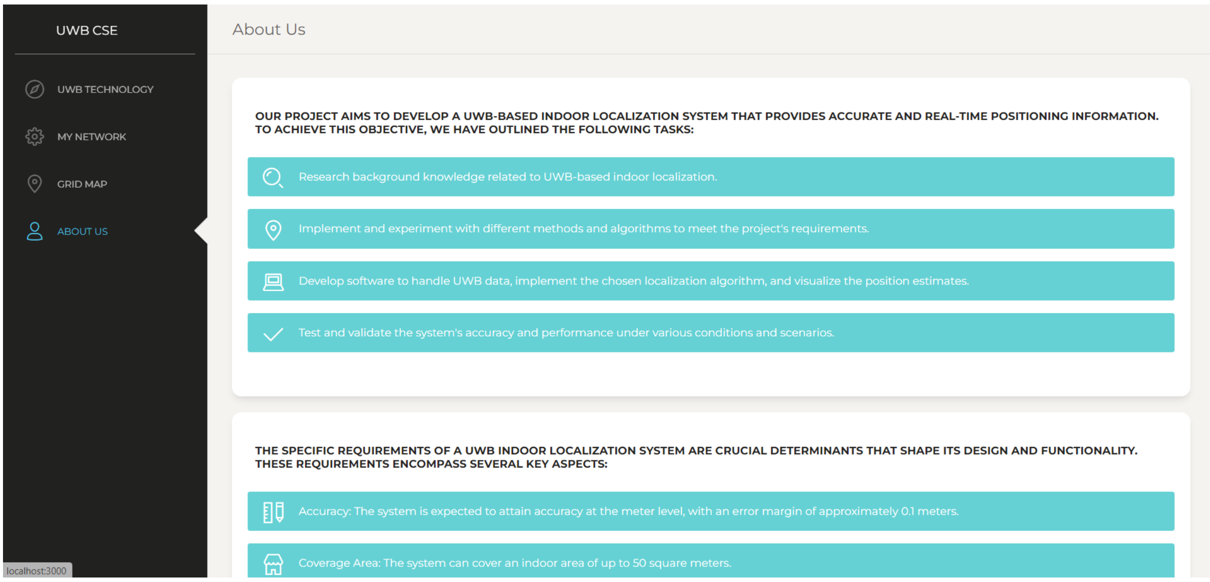

Figure 4.30: About Us view

CHAPTER 5

EXPERIMENTS AND EVALUATE THE SYSTEM

5.1 Experiments

We used the project requirements outlined in our introduction chapter to create the experimental scenarios for our system. After conducting experiments, we analyze the results to evaluate our system's performance and determine if it meets the predefined requirements.

5.1.1 Accuracy Requirement

5.1.1.1 Explanation of Experimental Scenarios

This experiment tests the precision requirement: The system is expected to achieve meter-level accuracy, with an error margin of less than 0.3 meters, even in non-line-of-sight (NLOS) environments.

Therefore, the setup of this test is in the same room of size 4.3 x 4.5 meters. We calculated 10 fixed points on a coordinate grid, and for each point, we capture the results of the system 10 times. This results in a total of 100 measurements for the entire test. We perform 2 test, one with LOS environment and one with NLOS environment.

The method to evaluate this requirement will go through 3 steps:

1. Calculate Euclidean Distance for Each Error:

$$\text{Error}_i = \sqrt{(x_{\text{i}} - x_0)^2 + (y_{\text{i}} - y_0)^2} \tag{5.1}$$

This calculates the error for each measurement, where x_0, y_0 are the actual coordinates of the fixed point, and x_i, y_i are the estimated coordinates by the system.

2. Find the Mean Error:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N}\text{Error}_i \tag{5.2}$$

Here, $\bar{x}$ is the mean of the errors over N measurements.

3. Calculate Standard Deviation:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\text{Error}_i - \bar{x})^2} \tag{5.3}$$

Use the errors calculated from the Euclidean distances to compute the standard deviation, which indicates the precision of the localization system. A smaller standard deviation suggests the system reliably estimates positions close to their actual values, indicating high precision.

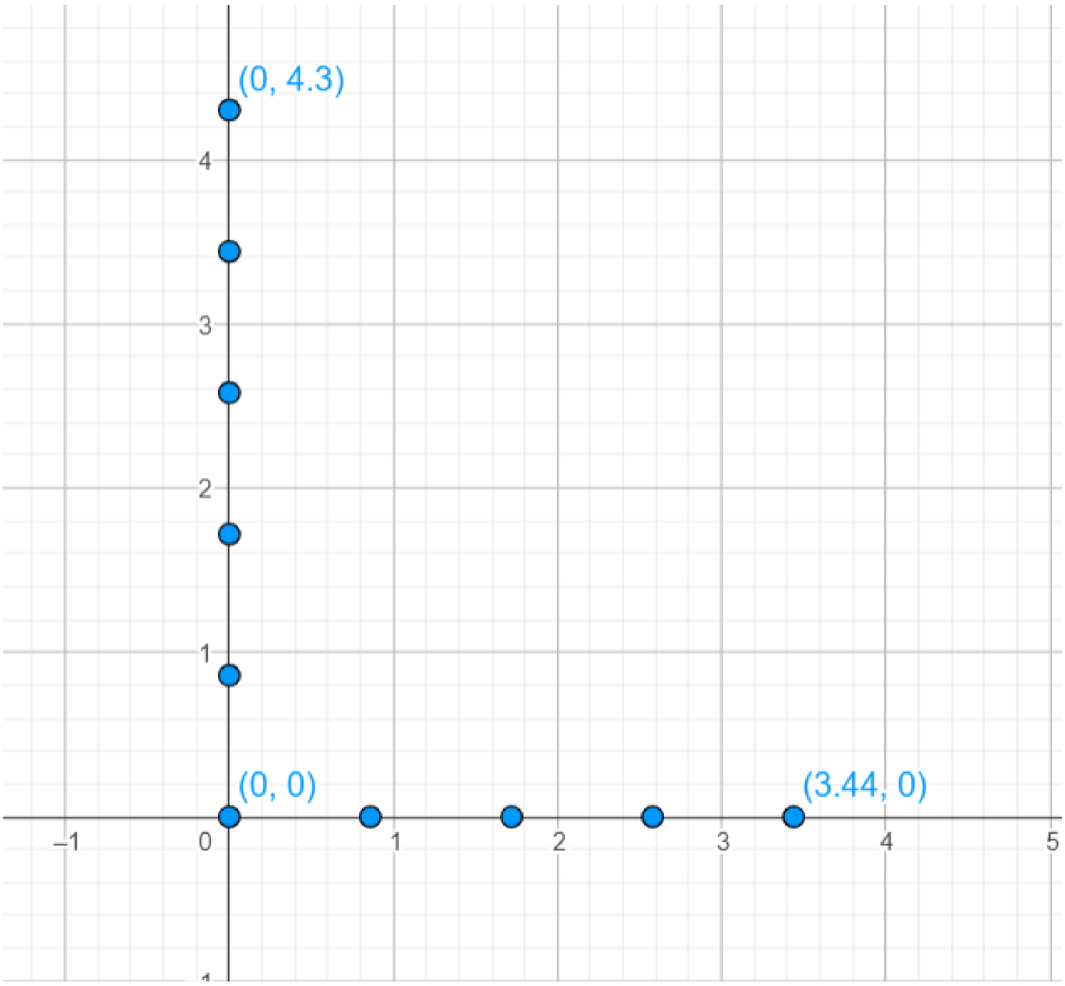

Figure 5.1: Coordinate testing map

5.1.1.2 Experimental results

*Experiment for Precision Requirement in LOS environment:

	LOS
Testing points (points)	100
Min Error (m)	0.0335
Max Error (m)	0.5093
Mean Error (m)	0.2565
Standard Deviation (m)	0.2371

Table 5.1: Experiment results for Precision Requirement in LOS environment

Explanation of the Experiment Result Table:

- Testing Points: The experiment involves 10 fixed points, with each point tested 10 times by the system, resulting in a total of **100** measurements for each test. Unit: points.
- Min Error: This is the smallest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The minimum error recorded is **0.0335** meters, indicating that some results were very close to the actual positions. Unit: meters (m).
- Max Error: This is the largest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The maximum error recorded is **0.5094** meters. Unit: meters (m).
- Mean Error: This is the average of all errors after 100 measurements. The mean error in this test is **0.2565** meters, which is below the target error margin of 0.3 meters, meeting our system's accuracy requirement. Unit: meters (m).
- Standard Deviation: The standard deviation in this test is **0.2371** meters. This value is significant as it indicates the consistency of the measurements; a lower standard deviation relative to the mean error suggests that most errors cluster tightly around the mean. Unit: meters (m).

***Experiment for Precision Requirement in NLOS environment:**

	NLOS
Testing points (points)	100
Min Error (m)	0.0451
Max Error (m)	0.5802
Mean Error (m)	0.2843
Standard Deviation (m)	0.2744

Table 5.2: Experiment results for Precision Requirement in NLOS environment

Explanation of the Experiment Result Table:

- Testing Points: The experiment involves 10 fixed points, with each point tested 10 times by the system, resulting in a total of **100** measurements for each test. Unit: points.
- Min Error: This is the smallest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The minimum error recorded is **0.0451** meters, indicating that some results were very close to the actual positions. Unit: meters (m).
- Max Error: This is the largest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The maximum error recorded is **0.5802** meters. Unit: meters (m).
- Mean Error: This is the average of all errors after 100 measurements. The mean error in this test is **0.2843** meters, which is below the target error margin of 0.3 meters, meeting our system's accuracy requirement. Unit: meters (m).
- Standard Deviation: The standard deviation in this test is **0.2744** meters. This value is significant as it indicates the consistency of the measurements; a lower standard deviation relative to the mean error suggests that most errors cluster tightly around the mean. Unit: meters (m).

5.1.2 Coverage Area Requirement

5.1.2.1 Explanation of Experimental Scenarios

This experiment tests the Coverage Area requirement: The simplest setup of the system could cover an indoor space of up to 54 square meters while maintaining an error margin below 0.3 meters.

This requirement encompasses two main aspects: the capability of the simplest system setup and its coverage area, which should extend up to 54 square meters with a precision indicated by a standard deviation of errors less than 0.3 meters.

To validate this requirement, the experiment is conducted in two rooms of different sizes and with varying numbers of anchors. Specifically, tests are carried out in:

- A 4.3 x 4.5 meters room (approximately 19.35 square meters) with both 3 and 4 anchors:
  - Fixed Points: There are 10 fixed points within this room.
  - Measurements per Point: Each fixed point is used to record results from 17 system-measured positions.
  - Total Measurements: This results in 170 test points (10 fixed points x 17 measurements each).

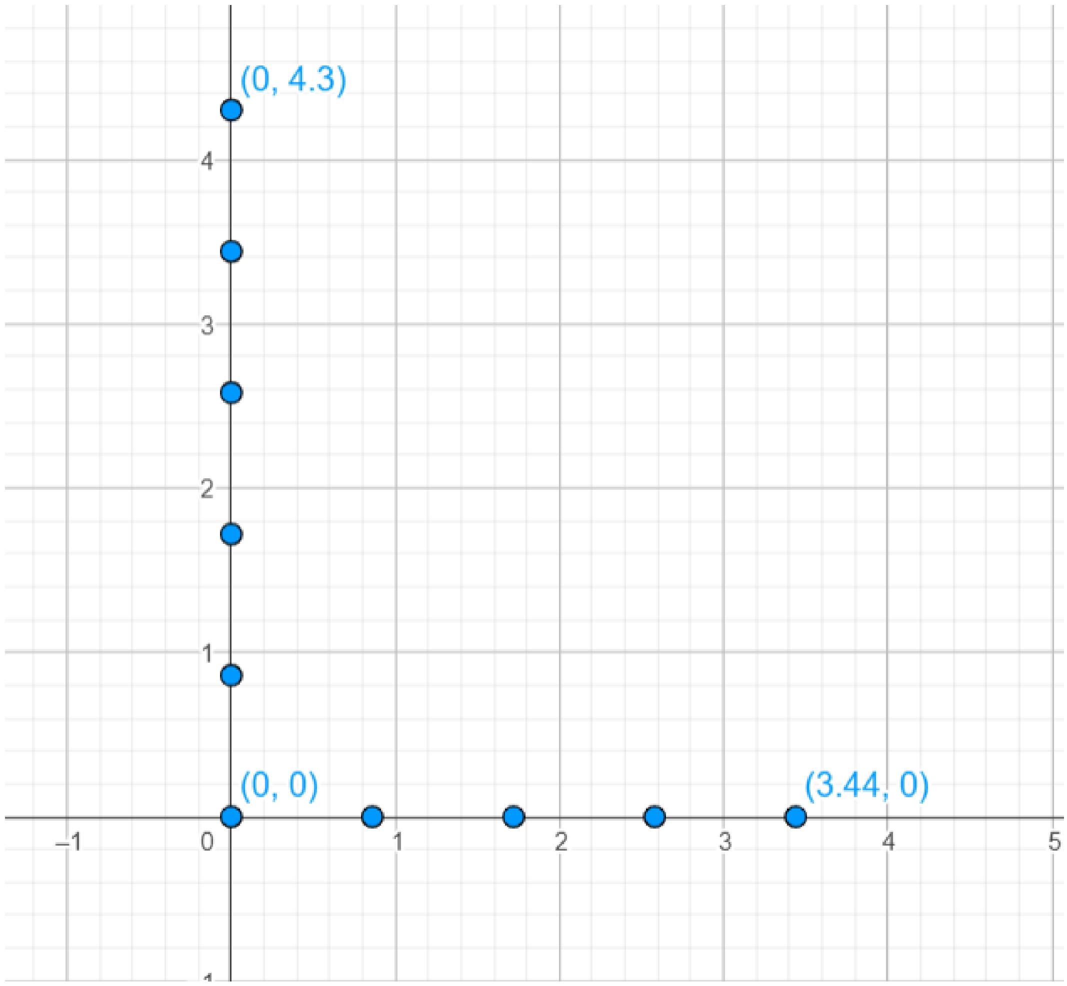

Figure 5.2: Coordinate testing map for 4.3 x 4.5 room

- A 6 x 9 meters room (approximately 54 square meters) with both 3 and 4 anchors:
  - Fixed Points: There are 17 fixed points within this room.
  - Measurements per Point: Each fixed point is used to record results from 10 system-measured positions.
  - Total Measurements: This also results in 170 test points (17 fixed points x 10 measurements each).

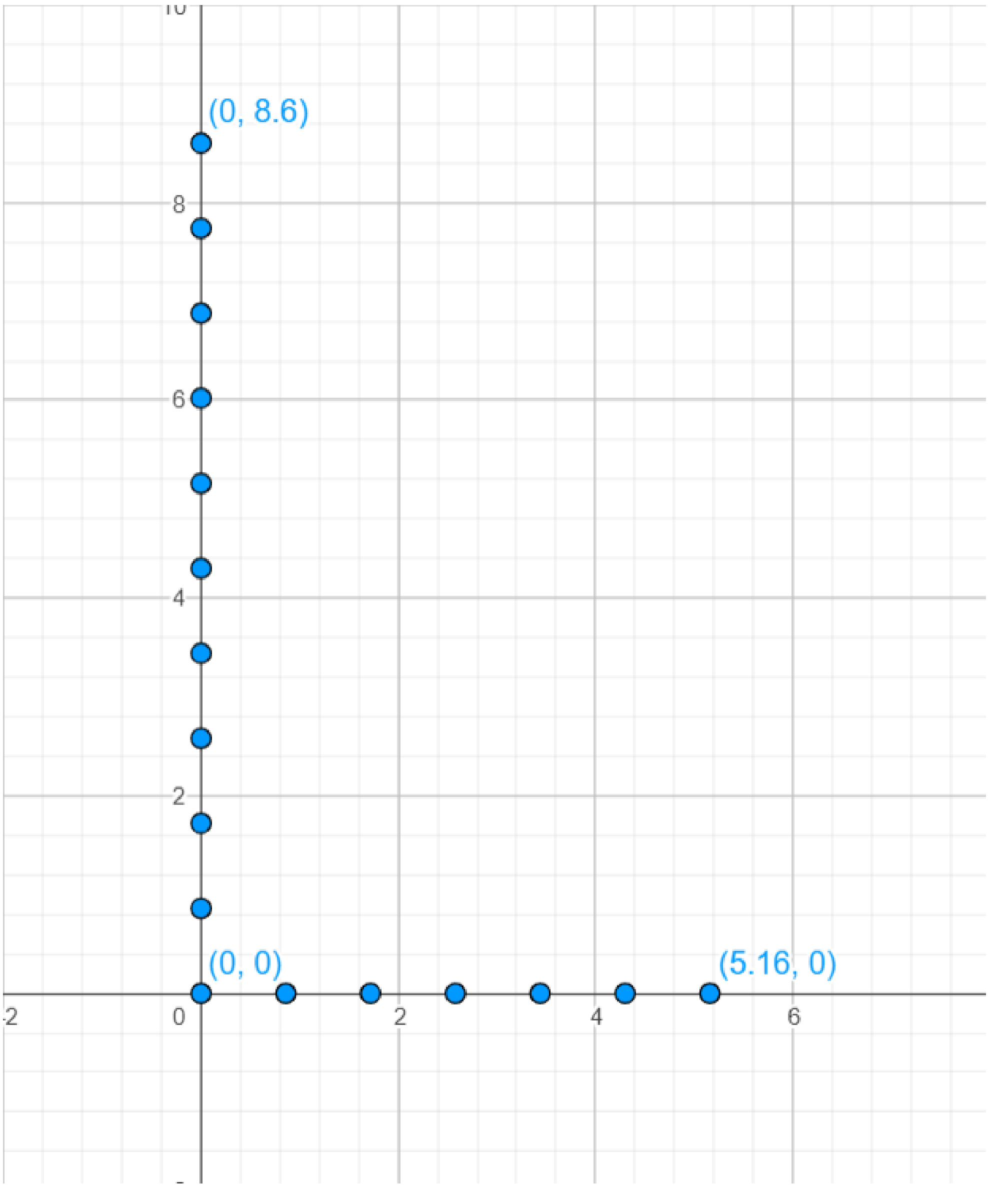

Figure 5.3: Coordinate testing map for 6.0 x 9.0 room

The evaluation methods are similar with the above experiments, which are calculate Euclidean Distance, then compute Mean Error, and finally determine Standard Deviation. Those methodologies allow us to determine if the system can reliably cover larger areas while still maintaining high accuracy and precision as required.

5.1.2.2 Experimental results

***Experiment for Coverage Area Requirement in 4.3 x 4.5 meters room:**

4.3 x 4.5 meters room	3 anchors setup	4 anchors setup
Testing points (points)	170	170
Min Error (m)	0.0324	0.0317
Max Error (m)	0.5054	0.5108
Mean Error (m)	0.2618	0.2336
Standard Deviation (m)	0.2402	0.2195

Table 5.3: Experiment results for Coverage Area Requirement in 4.3 x 4.5 meters room

Explanation of the Experiment Result Table:

- Testing Points: The experiment involves 10 fixed points, with each point tested 17 times by the system, resulting in a total of **170** measurements for each test. Unit: points.
- Min Error: This is the smallest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The minimum error recorded are **0.0324** meters and **0.0317**, indicating that some results were very close to the actual positions. Unit: meters (m).
- Max Error: This is the largest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The maximum error recorded are **0.5054** meters and **0.5108** meters. In this experiment, the Max error of a system with 4 anchors even larger than a system with only 3 anchors. Unit: meters (m).
- Mean Error: This is the average of all errors after 140 measurements. The mean error in this test are **0.2618** meters and **0.2336** meters, which is below the target error margin of 0.3 meters, meeting our system's accuracy requirement. Unit: meters (m).
- Standard Deviation: The standard deviation in this test are **0.2402** meters and **0.2195** meters. This value is significant as it indicates the consistency of the measurements; a lower standard deviation relative to the mean error suggests that most errors cluster tightly around the mean. Unit: meters (m).

In comparing the results from the setups with 3 and 4 anchors in a 4.3 x 4.5 meters room, it is evident that the setup with 4 anchors performs marginally better. However, the difference between the two setups is not substantial, with the standard deviation differing by only 0.0207 meters, which indicates a very minor improvement in precision.

Besides, the 4 anchors setup yielding a larger maximum error, this could potentially be attributed to the increased number of devices affecting signal robustness. However, it's important to note that both the mean error and the standard deviation are lower in the 4 anchors system, indicating better overall performance. The presence of a larger maximum error in this setup could also be due to random variations inherent in testing a complex system like this.

***Experiment for Coverage Area Requirement in 6.0 x 9.0 meters room:**

6.0 x 9.0 meters room	**3 anchors setup**	**4 anchors setup**
Testing points (points)	170	170
Min Error (m)	0.0318	0.0295
Max Error (m)	0.6037	0.5760
Mean Error (m)	0.2844	0.2613
Standard Deviation (m)	0.2748	0.2496

Table 5.4: Experiment results for Coverage Area Requirement in 6.0 x 9.0 meters room

Explanation of the Experiment Result Table:

- Testing Points: The experiment involves 17 fixed points, with each point tested 10 times by the system, resulting in a total of **170** measurements for each test. Unit: points.
- Min Error: This is the smallest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The minimum error recorded are **0.0318** meters and **0.0295**, indicating that some results were very close to the actual positions. Unit: meters (m).
- Max Error: This is the largest Euclidean distance error observed between a system-measured point and its corresponding fixed point. The maximum error recorded are **0.6037** meters and **0.5760** meters. Unit: meters (m).
- Mean Error: This is the average of all errors after 140 measurements. The mean error in this test are **0.2844** meters and **0.2613** meters, which is below the target error margin of 0.3 meters, meeting our system's accuracy requirement. Unit: meters (m).

- Standard Deviation: The standard deviation in this test are **0.2748** meters and **0.2496** meters. This value is significant as it indicates the consistency of the measurements; a lower standard deviation relative to the mean error suggests that most errors cluster tightly around the mean. Unit: meters (m).

Here's a revised and clarified version of your analysis, comparing experiment results between different room sizes and anchor setups:

The experiments conducted in the 6.0 x 9.0 meter room demonstrate that the setup with four anchors also yields better results than the setup with three anchors. However, the difference in performance is quite modest, with only a 0.0252 meter difference in standard deviation.

Comparing results across the two different room sizes, it's observed that errors in the larger room (6.0 x 9.0 meters) tend to be higher than in the smaller room (4.3 x 4.5 meters). Specifically, the standard deviations in the larger room are 0.2748 and 0.2496 meters for the 3 and 4 anchors setups, respectively. This compares to 0.2402 and 0.2195 meters in the smaller room. Taking the average standard deviations for each room, we find 0.2622 meters in the 6.0 x 9.0 room and 0.22985 meters in the 4.3 x 4.5 room. The difference between these averages is 0.03235 meters, indicating slightly higher variability in the larger space.

Despite the larger errors in the 6.0 x 9.0 meter room, the system with the minimal setup of 3 anchors still achieves high accuracy localization, with the standard deviation remaining well below the 0.3 meters threshold, even in a space as large as 54 m^2. This suggests that even with a simplest setup (3 anchors), the system is capable of maintaining effective performance within the specified accuracy requirements. The results are meeting our system requirement.

5.1.3 Display Latency Requirement

5.1.3.1 Explanation of Experimental Scenarios

Testing the Display Latency Requirement: The requirement specifies that tag data must be displayed on the application within 6 seconds after the tag moves.

The decision to set the maximum display latency at 6 seconds in the application is based on the sum of individual delays and an allowance for processing overhead in the system. Specifically:

- Publish Data Delay: The function that publishes data to Adafruit introduces a delay of approximately 2 seconds. This accounts for the time taken to transmit the data from the source to the Adafruit IoT platform.
- Fetch Data Delay: The function that fetches data into the application incurs a delay of approximately 1 second. This covers the time required to retrieve the data from Adafruit and make it available in the application.
- System Processing Time: To accommodate any additional processing time within the system itself, we consider the worst-case scenario by doubling the total delay from data publishing and fetching. Thus, 2 seconds (publish) + 1 second (fetch) = 3 seconds, and doubling this gives an additional 3 seconds for system processing.

We setup the experiment like below:

- Location: The test is conducted in a 4.3 x 4.5 meter room.
- Configuration: The setup includes 3 anchors and 1 tag.
- Procedure: The real-time movement of the tag is tracked, and the corresponding display time on the application is recorded when the tag's new position is shown.
- Repetitions: This process is repeated 100 times to ensure robust data collection.

The methodology to calculate the experiment results including:

1. Record Timing: For each of the 100 tests, record the time when the real tag moves and the time when the application displays the tag's new position.

2. Calculate Time Delays: For each instance, calculate the delay between the tag's actual movement and its display on the application.

3. Compute Mean Delay: Calculate the mean of these delays to see if, on average, the display is within 6 seconds after the tag moves.

4. Calculate Standard Deviation: Determine the standard deviation

of the delays to assess the consistency of the data.

Based on the results, verify if the mean time delay is less than 6 seconds to determine if the system meets the Display Latency Requirement.

5.1.3.2 Experimental results

Number of tests (tests)	100
Min Time Delay (s)	2.2
Max Time Delay (s)	8.1
Mean Time Delay (s)	4.9
Standard Deviation (s)	3.2

Table 5.5: Experiment results for Display Latency Requirement

Explanation of the Experiment Result Table:

- Number of tests: This process is repeated **100** times. Unit: tests.
- Min Error: This is the smallest Time Delay calculated between the tag's actual movement and its display on the application. The smallest Time Delay of this system is **2.2** seconds. Unit: seconds (s).
- Max Error: This is the largest Time Delay calculated between the tag's actual movement and its display on the application. The smallest Time Delay of this system is **6.3** seconds. Unit: seconds (s).
- Mean Error: This is the the mean of these delays, which is **4.7** seconds. This number meet our requirement of a system with time delay is less than 6 seconds. Unit: seconds (s)
- Standard Deviation: The standard deviation of the delays to assess the consistency of the data. The SD calculated from this project is **2.2**. Unit: seconds (s).

The experimental data indicate that the mean display latency is 4.9 seconds, which is under the maximum allowed threshold of 6 seconds. This confirms that the system meets the designated display latency requirement.

Additionally, the standard deviation of the display latency is 2.2 seconds. This measure indicates that there is considerable variability in the latency results, with some delays deviating significantly from the mean. While the mean latency is within the acceptable range, the relatively high standard deviation suggests that the timing of data display can vary widely, which might impact user experience or system reliability in real-world applications.

5.2 Evaluate the System

To evaluate the system, we analyze our experimental results alongside the interface of our app to determine if they meet the requirements of our project.

About our first requirement, the system is expected to achieve meter-level accuracy, with an error margin of less than 0.3 meters, even in non-line-of-sight (NLOS) environments. Based on our experiments, the mean errors are 0.2565 meters for LOS conditions and 0.2843 meters for NLOS conditions, both of which are below the maximum allowable error of 0.3 meters. Additionally, the difference in mean distance errors between NLOS and LOS is only 0.0278 meters, which is quite minimal. Therefore, we conclude that our system consistently achieves distance accuracy with an error under 0.3 meters, even in NLOS environments. With a standard deviation of only 0.2744 meters, the errors are relatively close to each other, further indicating a high level of precision.

About our second requirement, the simplest setup of the system could cover an indoor space of up to 54 square meters while maintaining an error margin below 0.3 meters. Based on our experiments, the comparison between systems with four and three anchors shows that four anchors offer a marginally higher accuracy: by 0.0277 meters in a 4.3x4.3m room and 0.0231 meters in a 6.0x9.0m room. Although the four-anchor system provides slightly better results, the improvement is minimal. Considering cost efficiency, the simplest system configuration, which includes three anchors, is sufficient for UWB localization. Regarding coverage space, the mean errors in a 54 square meter room are only slightly larger than those in a 19.35 square meter room—by a mere 0.0226 meters, which is a negligible difference. We conclude that the simplest system setup can effectively support good performance, with mean errors still below 0.3 meters, even in spaces as large as 54 square meters.

About our third requirement, the tag data must be displayed on the application within 6 seconds after the tag moves. Based on experimental data, the average time delay is only 4.9 seconds, which is within the acceptable range. However, the standard deviation is relatively large, and some instances recorded delays of more than 8 seconds. This variability can be attributed to our use of the free version of Adafruit.io, which imposes limits on data transfer rates per minute and may occasionally block data flow when these limits are reached, leading to longer display times on the application. Despite these challenges, the mean delay remains within an acceptable margin, indicating that the requirement is met overall. This suggests that while the average performance is satisfactory, occasional spikes in delay might affect the consistency of the system experience.

About our final requirement, develop an easy-to-use app that effectively visualizes localization data in an understandable manner. We have successfully created an application designed to visualize an UWB system through a simple three-step process: scan modules, generate coordinates, and view results on a grid map. Additionally, users have the option to overlay a floor plan on the grid map to enhance the demonstration of the localization data, making the visualization more contextual and relevant to specific environments. Our application also includes information about UWB technology and the methodologies used to calculate positions, which helps users gain a better understanding of this technology.

Furthermore, there are several ways in which this system can be enhanced to provide even better performance and functionality. Increasing the number of anchors could significantly improve the accuracy of our UWB system. By doubling the number of modules, we would not only enhance the precision within a single room but could also extend the system to cover multiple rooms. Linking these expanded systems through a gateway would enable a more comprehensive and seamless tracking network across larger areas.

Additionally, investing in a premium subscription to Adafruit.io would provide higher data transmission rates, reducing the likelihood of data bottlenecks and improving the overall responsiveness of our application. Enhanced data flow would lead to faster updates and more timely display of localization data on the application, improving the user experience by providing real-time tracking updates with minimal delay.

By implementing these improvements, we could greatly increase the system's robustness and utility, making it more versatile and effective in a variety of settings. This would not only meet the current needs but also adapt to future expansions and challenges, ensuring that the system remains at the cutting edge of UWB technology.

CHAPTER 6

CONCLUSION

6.1 Evaluate the Process

Over the course of the 15 weeks for of Computer Engineering Project and 15 weeks for the Capstone Project, the team has achieved the established objectives, including:

- *Clarifying background knowledge surrounding UWB-based indoor localization technology:*
  - About indoor localization technology, we have compiled and conducted research on various Radio Frequency (RF) based navigation technologies. Such as Wi-Fi, Bluetooth, RFID, and UWB technology.
  - About localization measurement techniques, our research involved categorizing methods into angle-based, distance-based, and fingerprinting based methods. Subsequently, we conducted evaluations to discern the advantages and disadvantages in each method.
  - About localization algorithms, our study delves into trilateration and triangulation algorithms. We explore the formulas and methodologies for their implementation.
- *Implement and experiment different methods and algorithms to achieve the expected requirements:*
  - Calibrating anchor devices by implement a local coordinate system, document and visualize the outcomes.
  - Measuring distances between anchors and tags by setting up a Sym-

metric Double-Sided Two-Way Ranging distance measurement exchange.

  - Comparing the documented results with real-life outcomes and identify solutions to enhance the system's performance.
  - Applying the trilateration algorithm to determine the tag's position and illustrating the outcomes through graphical representations.

- *Develop the software to handle UWB data, implement the chosen localization algorithm, and visualize the position estimates:*
  - Set up the hardware devices, which include 3 anchors, 1 tag, and 1 listener.
  - Push the collected data to a server for processing.
  - Develop a web application to visualize the position estimates and provide an intuitive user interface for interacting with the system.
- *Test and validate the system accuracy along with performance under different conditions and scenarios:*
  - Build experimental scenarios based on the requirements of the project.
  - Conduct the experiments and gather the results.
  - Evaluate the system based on experimental results, and then propose ways for the system to improve.

In conclusion, the system and application performed as expected. The main issue we encountered during implementation was related to data transmission limits. We also faced some hardware problems in phase 1, but we were able to rectify these issues in phase 2. This project has provided us with substantial knowledge about indoor localization systems. Along with data processing and database usage for applications. Additionally, we have learned a great deal about how to construct testing scenarios for a system based on its objectives and requirements. Overall, we have learned a lot throughout the two semesters working on this project.

6.2 Development Plan

For the future of this project, there are several ways to enhance the implementation of a UWB indoor localization system, including:

- Expand the Network of Anchors and Build a Gateway: Adding more anchors and constructing a simple gateway, such as a Raspberry Pi connected to a DWM1001-DEV module, could enable communication across rooms. This expansion would facilitate indoor localization on a larger scale.
- Integrate Additional Localization Techniques: Incorporating other localization methods, such as WiFi or Bluetooth, alongside UWB could create a hybrid system with enhanced performance. Combining these technologies would leverage the strengths of each to achieve more accurate and reliable localization.
- Enhance Visualization to Include 3D Results: Improving the system to display localization results in 3D would provide a more comprehensive view of the environment, enhancing user interaction and the overall utility of the application.
- Develop Application Features for Specific Users: Tailoring the application to meet the needs of specific user groups could significantly extend its utility. For example, creating customized localization systems for factories, hospitals, or schools could address unique challenges and requirements found in these environments.

REFERENCES

[1] Anup Kumar Paul, Takuro Sato (October 27, 2017). *Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges.*

[2] Qorvo. *Ultra-wideband Technology.* `https://www.qorvo.com/innovation/ultra-wideband/technology`

[3] Simon, Nikolas A (Febuary 16, 2016). *Trilateration Calculation Program for Location Tracking System* .

[4] Huthaifa Obeidat, Wafa Shuaieb, Omar Obeidat, Raed Abd-Alhameed (Febuary 19, 2021) *A Review of Indoor Localization Techniques and Wireless Technologies.*

[5] Sujata Mohanty, Dr.Aruna Tripathy, Dr.Bikramaditya Da (May, 2023). *UWB Based Indoor Localization Techniques.*

[6] Shimiao Li (July 30, 2017).*A review of feature detection and match algorithms for localization and mapping.*

[7] Ke Shi, Zhenjie Ma, Rentong Zhang, Wenbiao Hu, and Hongsheng Chen (January, 2015). *Support Vector Regression Based Indoor Location in IEEE 802.11 Environments.*

[8] Yuekui Zhang, Yunxia Chu, Yunfang Fu, Zhixiang Li, Yufei Song (November 1, 2021). *UWB Positioning Analysis and Algorithm Research.*

[9] Chen Yi, An Ze Da, Chen Hui, Chen Shan, Zhou Xuan (2021). *A UWB Location Algorithm Based on Adaptive Kalman Filter.*

[10] Ivan A. Mantilla-Gaviria, Mauro Leonardi, Gaspare Galati, Juan V. Balbastre-Tejedor (January, 2014). *Localization Algorithms for Multilateration (MLAT) Systems in Airport Surface Surveillance.*

[11] Adham Sabra, Wai-keung Fung (September,2020). *A Fuzzy Cooperative Localisation Framework for Underwater Robotic Swarms.*

[12] Chia-Yen Shih, P. Marrón (July 18, 2010). *COLA: Complexity-Reduced Trilateration Approach for 3D Localization in Wireless Sensor Networks.*

[13] Fatiha Mekelleche, Haffaf Hafid (April, 2017). *Classification and Comparison of Range-Based Localization Techniques in Wireless Sensor Networks.*