



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

SMARTPHONE AS A LOCATOR IN A LOCALIZATION SYSTEM

SMARTPHONE JAKO LOKÁTOR V LOKALIZAČNÍM SYSTÉMU

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Adam Bičíš

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Radovan Juráň

BRNO 2025

Bachelor's Thesis

Bachelor's study program **Telecommunication and Information Systems**

Department of Telecommunications

Student: Adam Bičíš

ID: 253176

**Year of
study:** 3

Academic year: 2024/25

TITLE OF THESIS:

Smartphone as a locator in a localization system

INSTRUCTION:

The goal of this thesis is to analyze the possibilities of integrating smartphones into the Sewio RTLS location platform using UWB technology. The objectives are to study the Sewio RTLS UWB location platform, the available UWB-enabled smartphones and analyze what radio modules they use. Next, for the given platforms, evaluate their API and integration options. If any platform allows it, design and implement a communication and integration model into Sewio's RTLS localization platform. Evaluate the integration options and the results achieved. Necessary hardware will be available.

The expected result is an analysis of available platform options (Android, iOS), then a mobile application will be created to evaluate the use of UWB on a given smartphone. Subsequently, the feasibility of smartphone integration into Sewio RTLS will be evaluated and, if possible, such integration will be implemented in the form of a demo scenario.

RECOMMENDED LITERATURE:

According to the instructions of the supervisor and consultant.

**Date of project
specification:** 10.2.2025

**Deadline for
submission:** 3.6.2025

Supervisor: Ing. Radovan Juráň

Consultant: Jakub Lešínský

prof. Ing. Jiří Mišurec, CSc.
Chair of study program board

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

This thesis explores the potential for integrating Ultra-Wideband (UWB)-enabled smartphones with the SEWIO RTLS positioning platform. The theoretical part presents an analysis of available smartphones supporting UWB technology, focusing on the radio modules they use and the integration capabilities of their respective platforms (Android, iOS). The practical part involves the development of a prototype application that demonstrates the use of UWB technology on supported devices and serves as a foundation for future integration into the SEWIO RTLS system. Due to technical limitations, direct connection with the SEWIO RTLS system could not be achieved; however, this work provides a theoretical basis for future research in this area.

KEYWORDS

Ultra-Wideband, UWB, SEWIO RTLS, mobile localization, Android, iOS, Bluetooth Low Energy, BLE, wireless communication, Kotlin, real-time location systems, RTLS

ABSTRAKT

Tato práce se zabývá možností integrace mobilních telefonů s technologií Ultra-Wideband (UWB) do lokalizační platformy SEWIO RTLS. Teoretická část analyzuje dostupné smartphony podporující technologii UWB, zejména z hlediska použitých rádiových modulů a možností integrace jejich platforem (Android, iOS). V praktické části je vyvíjen prototyp aplikace, která demonstruje využití technologie UWB na podporovaných zařízeních a představuje základ pro budoucí integraci do systému SEWIO RTLS. Z důvodu technických omezení nebylo možné realizovat přímé propojení se systémem SEWIO RTLS, avšak práce poskytuje teoretické základy pro budoucí výzkum v této oblasti.

KLÍČOVÁ SLOVA

Ultra-Wideband, UWB, SEWIO RTLS, mobilní lokalizace, Android, iOS, Bluetooth Low Energy, BLE, bezdrátová komunikace, Kotlin, systémy pro určování polohy v reálném čase, RTLS

ROZŠÍŘENÝ ABSTRAKT

Cílem této práce bylo prozkoumat možnosti integrace Ultra-Wideband (UWB) technologie v chytrých telefonech s lokalizační platformou SEWIO RTLS (Real-Time Locating System). Hlavním záměrem bylo vyhodnotit technické předpoklady pro využití UWB v mobilních zařízeních a navrhnout prototypovou aplikaci demonstrující lokalizaci založenou na této technologii.

Práce byla rozdělena na teoretickou a praktickou část. V teoretické části byly analyzovány dostupné smartphony s podporou UWB, se zaměřením na použité rádiové moduly a možnosti integrace v rámci mobilních platform (Android, iOS). Dále byla provedena komparace přístupů operačních systémů k UWB rozhraním, včetně dostupnosti API pro vývojáře.

V praktické části byl vyvinut prototyp aplikace pro Android, která ověřuje podporu UWB, navazuje spojení s UWB tagem pomocí Bluetooth Low Energy a realizuje dvousměrné měření vzdálenosti (Two-Way Ranging). Aplikace také obsahuje 3D vizualizační modul, který zobrazuje polohu tagu na základě naměřených parametrů (vzdálenost, azimut, elevace).

Přestože přímá integrace se systémem SEWIO RTLS nebyla z důvodu technických omezení realizována, práce poskytuje teoretický základ pro další výzkum v této oblasti. Prototypová aplikace prokázala, že UWB v mobilech umožňuje přesné měření polohy, avšak integrace s průmyslovými RTLS systémy vyžaduje další vývoj hardwarových i softwarových rozhraní.

Tato práce demonstruje rostoucí význam UWB technologie v kontextu lokalizačních systémů a mobilních zařízení. Přestože současná omezení brání plné kompatibilitě s existující infrastrukturou (jako je SEWIO RTLS), vyvinutý prototyp poskytuje cenné poznatky pro budoucí výzkum. Práce přispívá jak teoretickou analýzou UWB integrace v mobilech, tak praktickým návrhem aplikace, která může sloužit jako výchozí bod pro další vývoj.

BIČIŠTĚ, Adam. *Smartphone as a locator in a localization system*. Bachelor's Thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2025. Advised by Ing. Radovan Juráň

Author's Declaration

Author: Adam Bičíšřě
Author's ID: 253176
Paper type: Bachelor's Thesis
Academic year: 2024/25
Topic: Smartphone as a locator in a localization system

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno

.....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my thesis supervisor, Ing. Radovan Juráň, for his professional guidance, valuable consultations, patience, and insightful ideas throughout the development of this thesis. I would also like to thank Jakub Lešinský from Sewio Networks s.r.o. for providing the necessary hardware and sharing his expertise on the subject.

Contents

| | |
|--|-----------|
| Introduction | 12 |
| 1 Location | 13 |
| 1.1 Outdoor positioning systems | 13 |
| 1.2 Indoor positioning systems | 13 |
| 1.3 Real-time locating systems | 14 |
| 1.4 Sewio RTLS | 14 |
| 1.4.1 Hardware | 14 |
| 1.4.2 Software | 15 |
| 2 UWB technology | 18 |
| 2.1 Physical principles | 19 |
| 2.1.1 PHY Standards (IEEE 802.15.4a/b/z) | 19 |
| 2.1.2 Power Consumption | 19 |
| 2.1.3 Interference | 19 |
| 2.1.4 Higher-Level Standards | 20 |
| 2.2 Ranging and locationing | 20 |
| 2.2.1 Ranging methods | 21 |
| 2.2.2 Locationing methods | 22 |
| 3 UWB in smartphones | 25 |
| 3.1 Smartphone UWB Chip Manufacturers | 26 |
| 3.2 Android UWB module and API | 27 |
| 3.3 iOS UWB module and API | 28 |
| 4 Existing applications | 29 |
| 4.1 Qorvo Connect | 29 |
| 4.2 Qorvo Nearby Interaction | 30 |
| 5 Implementation | 32 |
| 5.1 Android application | 32 |
| 5.1.1 Programming | 35 |
| 5.2 Development kit | 41 |
| 5.2.1 Configuration | 42 |
| 5.2.2 Firmware | 42 |
| 5.3 Sewio RTLS integration | 45 |
| 5.3.1 Future Directions | 45 |

| | |
|---|-----------|
| Conclusion | 47 |
| Bibliography | 48 |
| Symbols and abbreviations | 52 |
| A Content of the electronic attachment | 53 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Sewio's UWB Tags | 15 |
| 1.2 | Sewio's UWB Anchors | 15 |
| 1.3 | Demo of RTLS Manager | 16 |
| 1.4 | Demo of Sensmap | 16 |
| 1.5 | Demo of API DOC | 17 |
| 1.6 | Demo of Sage Analytics | 17 |
| 2.1 | Spectral Density for Ultrawide and Narrowband | 18 |
| 2.2 | Time of Flight diagram | 22 |
| 2.3 | Single-sided two way ranging & double-sided two way ranging | 23 |
| 2.4 | Location determination with TDoA | 23 |
| 2.5 | Reverse TDoA | 24 |
| 2.6 | PDoA distance | 24 |
| 3.1 | UWB-Enabled device shipments prediction | 25 |
| 3.2 | Scheme of UWB connectivity between smartphones | 27 |
| 4.1 | Demo of Qorvo Connect app | 30 |
| 4.2 | Demo of Qorvo Nearby Interaction app | 31 |
| 5.1 | Main screen: Control buttons and device discovery results | 33 |
| 5.2 | Device interaction screen: connected device characteristics and log | 34 |
| 5.3 | UWB session screen: TWR configuration | 35 |

Listings

| | | |
|------|--|----|
| 5.1 | Code example of UWB SUPPORT button | 37 |
| 5.2 | Example of UWB SUPPORT button design | 37 |
| 5.3 | Code example of transition to the other screen | 38 |
| 5.4 | Code example of composable element | 38 |
| 5.5 | Code example of UWB parameters | 39 |
| 5.6 | Code example of arrow render | 39 |
| 5.7 | Code example of changing arrow orientation | 40 |
| 5.8 | Macro for enabling BLE advertising | 42 |
| 5.9 | Example command to run the python script | 43 |
| 5.10 | Sample output from a UWB ranging session | 44 |

Introduction

Knowing the exact location of items in any industrial process from manufacturing to storage and logistics has become crucial for optimizing operations, minimizing errors, and boosting overall efficiency. Precise item positioning streamlines inventory management and significantly cuts down the time spent searching for products.

Ultra-wide band (UWB) technology has become an ideal option to reach this level of accuracy [16]. UWB uses short radio pulses with a wide frequency range to locate objects within centimeters, even in complicated settings. Unlike classical methods like RFID, GPS, WiFi, or Bluetooth, UWB enables real-time, very precise tracking in indoor environments, making it suitable for handling warehouse inventory.

Usage of UWB-enabled smartphones improves this problem by making precise location tracking more accessible and adaptable. With UWB chips built into phones, workers can easily locate goods, reducing the need for specialized tracking equipment. This not only simplifies procedures, but also increases worker effectivity by allowing them to quickly locate things in the warehouse.

This thesis will analyze smartphones with ultra-wide band chips and see if and how they can work with the Sewio Real-Time Locating System (RTLS), a platform widely used for indoor localization. The theoretical part will involve a comparative study of various chips integrated in different phones. In the practical part, the research will focus on the development of an application designed to test UWB connectivity for interaction with the Sewio RTLS platform.

The goal of this thesis is to analyze the possibilities of integrating smartphones into the Sewio RTLS localization platform using UWB technology and exploring the current state-of-the-art tools currently available to enable UWB use to developers of mobile apps.

1 Location

The term locationing refers to the determination of the position of objects, devices, and people in space. We can divide locationing into two groups based on the environment, indoor and outdoor.

Indoor locationing, as the name suggests, is used in indoor environments (such as factories, warehouses, hospitals), and it is implemented through indoor positioning systems (IPS).

On the other hand, outdoor locationing occurs outside using outdoor positioning systems (OPS) [17].

1.1 Outdoor positioning systems

The most widely used technology for outdoor positioning is the Global Navigation Satellite System (GNSS), a network of satellites that provides global positioning and information to receivers anywhere on Earth. These satellites send signals to receivers (such as mobile phones, automotive navigation systems and air navigation) which use the data to calculate their location and velocity.

One of the well-known systems is called Global Positioning System (GPS), owned by the US but used worldwide. Other notable systems include Galileo, operated by the European Union; BeiDou, managed by China; and GLONASS, maintained by Russia.

These systems provide very precise locationing under perfect conditions (open sky, no reflective terrain, and no signal blockage). In contrast, a typical smartphone offers accuracy within a radius of 5 meters [34].

1.2 Indoor positioning systems

Because GNSS tends to be very unreliable and ineffective in indoor environments, we use indoor positioning systems (IPS) as an alternative [5]. These systems are specifically designed to provide accurate location within enclosed spaces. Existing IPS use a variety of technologies to accurately locate devices; these include WiFi, Bluetooth, UWB acoustic signals, and Radio Frequency Identification (RFID). Each of these technologies offers different advantages depending on the level of accuracy, range, and specific application required.

1.3 Real-time locating systems

Real-time locating systems (RTLS) are used to identify and track the location of objects or people in real time, typically within an indoor environment [8]. These systems rely on wireless tags that are attached to objects or worn by people. Fixed reference points, also known as anchors, are devices that receive signals from these tags and determine their location. Typical application of RTLS includes locating pallets in a warehouse, tracking vehicles through an assembly line, or monitoring employees.

The physical layer typically involves radio frequency (RF) communication. Some systems use optical technology (such as infrared) or acoustic technology (such as ultrasound) [20, 32]. Anchors can function as transmitters, receivers, or both, resulting in numerous possible combinations of technologies.

1.4 Sewio RTLS

Sewio is a technology company that specializes in RTLS that uses UWB technology to provide highly accurate indoor positioning and tracking [31]. Sewio's RTLS is often used in industries such as manufacturing, warehousing and logistics to track precise locations of assets and people in real-time within indoor environments. This tracking is essential to improve operational efficiency, safety and productivity [18].

1.4.1 Hardware

The Sewio UWB RTLS platform includes two key hardware components: signal transmitters, known as Tags, which track entities, and signal receivers, known as Anchors, which capture signals from the Tags.

Tags are compact electronic devices attached to objects or individuals that are to be tracked. They function as transmitters, emitting pulses, or 'blinks', which are received by anchors and relayed to a location server to calculate the position of each tag.



Fig. 1.1: Sewio's UWB Tags [31]

Anchors are devices that detect UWB signals from tags and forward them to the location server for precise positioning. To establish an effective indoor tracking network, anchors are strategically installed above the designated area to form the necessary location infrastructure.



Fig. 1.2: Sewio's UWB Anchors [31]

1.4.2 Software

Sewio offers RTLS Studio, a comprehensive software solution designed to ease the installation, deployment, maintenance, and management of real-time indoor location tracking systems. The core application is **RTLS Manager**, which is used for deployment, configuration, and system monitoring. It provides real-time health monitoring of anchors, tags, and servers and offers tools for system initialization, re-scanning, and synchronization.

| ID | Anchor Mac | Update | Alias | Status | IP | FW | Product Name | HW | Update (k, m, s) | LED | Location | Master | Sync Profile | Sync Channel | Bk, Profile | Bk, Channel | Power |
|----|-------------|--------|-------|--------|-------------|-------|--------------|-----|------------------|---------|---------------|--------|--------------|--------------|-------------|-------------|---------|
| 1 | 08003925595 | UPDATE | SE40 | ✓ | 10.103.8.59 | 1.024 | N/A | N/A | 24, 00, 33, 05 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 2 | 08003926895 | UPDATE | SE50 | ✓ | 10.103.8.64 | 1.024 | N/A | N/A | 24, 00, 33, 16 | TURN ON | True Location | Yes | RF 4 | 3 | RF 4 | 5 | default |
| 3 | 08003926953 | UPDATE | SE39 | ✓ | 10.103.8.58 | 1.024 | N/A | N/A | 24, 00, 33, 17 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 4 | 08003922017 | UPDATE | SE48 | ✓ | 10.103.8.62 | 1.024 | N/A | N/A | 24, 00, 33, 00 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 5 | 08003925563 | UPDATE | SE55 | ✓ | 10.103.8.69 | 1.024 | N/A | N/A | 24, 00, 33, 13 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 6 | 08003926240 | UPDATE | SE32 | ✓ | 10.103.8.51 | 1.024 | N/A | N/A | 24, 00, 33, 46 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 7 | 08003925446 | UPDATE | SE22 | ✓ | 10.103.8.41 | 1.024 | N/A | N/A | 24, 00, 33, 20 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 8 | 08003922484 | UPDATE | SE20 | ✓ | 10.103.8.39 | 1.024 | N/A | N/A | 24, 00, 33, 27 | TURN ON | True Location | Yes | RF 4 | 3 | RF 4 | 5 | default |
| 9 | 08003922763 | UPDATE | SE18 | ✓ | 10.103.8.35 | 1.024 | N/A | N/A | 24, 00, 33, 10 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 10 | 08003922188 | UPDATE | SE15 | ✓ | 10.103.8.34 | 1.024 | N/A | N/A | 24, 00, 33, 22 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 11 | 08003922828 | UPDATE | SE23 | ✓ | 10.103.8.40 | 1.024 | N/A | N/A | 24, 00, 33, 05 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 12 | 08003925089 | UPDATE | SE39 | ✓ | 10.103.8.28 | 1.024 | N/A | N/A | 24, 00, 33, 40 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 13 | 08003921796 | UPDATE | SE52 | ✓ | 10.103.8.20 | 1.024 | N/A | N/A | 24, 00, 33, 23 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 14 | 08003925546 | UPDATE | SE11 | ✓ | 10.103.8.30 | 1.024 | N/A | N/A | 24, 00, 33, 31 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 15 | 08003921354 | UPDATE | SE35 | ✓ | 10.103.8.54 | 1.024 | N/A | N/A | 24, 00, 33, 10 | TURN ON | True Location | Yes | RF 4 | 3 | RF 4 | 5 | default |
| 16 | 08003922431 | UPDATE | SE34 | ✓ | 10.103.8.53 | 1.024 | N/A | N/A | 24, 00, 33, 06 | TURN ON | True Location | Yes | RF 4 | 3 | RF 4 | 5 | default |
| 17 | 08003925667 | UPDATE | SE03 | ✓ | 10.103.8.22 | 1.024 | N/A | N/A | 24, 00, 33, 36 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 18 | 08003922376 | UPDATE | SE02 | ✓ | 10.103.8.21 | 1.024 | N/A | N/A | 24, 00, 33, 14 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 19 | 08003922050 | UPDATE | SE17 | ✓ | 10.103.8.36 | 1.024 | N/A | N/A | 24, 00, 33, 06 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 20 | 08003922615 | UPDATE | SE25 | ✓ | 10.103.8.44 | 1.024 | N/A | N/A | 24, 00, 33, 01 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |
| 21 | 08003922216 | UPDATE | SE10 | ✓ | 10.103.8.29 | 1.024 | N/A | N/A | 24, 00, 33, 10 | TURN ON | True Location | Yes | RF 4 | 3 | RF 4 | 5 | default |
| 22 | 08003925000 | UPDATE | SE23 | ✓ | 10.103.8.42 | 1.024 | N/A | N/A | 24, 00, 33, 18 | TURN ON | True Location | No | RF 4 | 3 | RF 4 | 5 | default |

Fig. 1.3: Demo of RTLS Manager [30]

Sensmap, another application within RTLS Studio, enables real-time visualization for indoor tracking projects. It allows for the addition of zones, paths, walls, and exits, along with the creation of unlimited, precisely detailed virtual zones with 30 cm accuracy. The built-in geofencing, alerts, and notifications provide flexible support for current and future indoor tracking requirements.



Fig. 1.4: Demo of Sensmap [30]

The **API DOC** module of RTLS Studio enables developers to integrate real-time location intelligence into their solutions using fully documented Open API. RTLS Studio supports seamless, bidirectional data exchange via HTTP REST, WebSocket, and UDP streaming. Developers can access all essential data formats and a single streamlined interface for quick and effective integration.

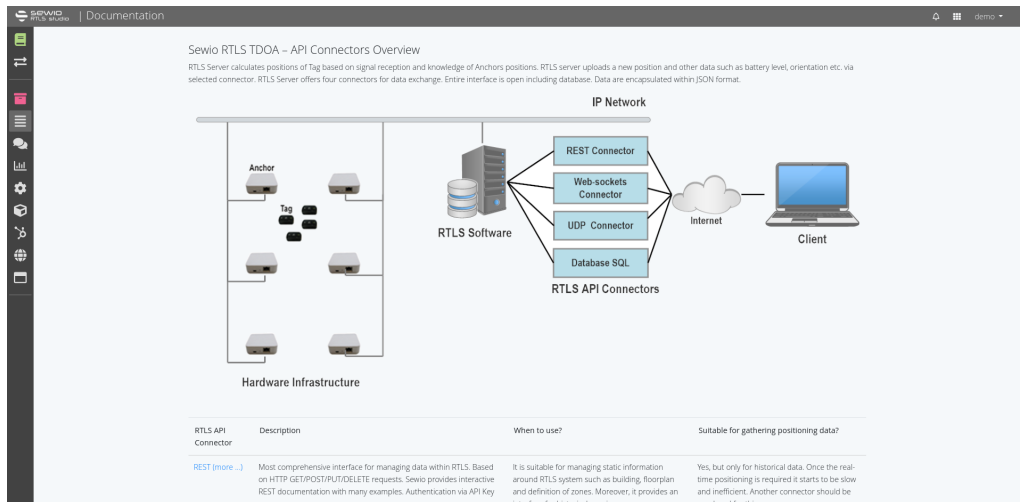


Fig. 1.5: Demo of API DOC [30]

Sage Analytics allows you to derive valuable insights from location data by visualizing and analyzing tracking results using trajectory maps, heat maps, spaghetti diagrams and advanced zone analytics. This powerful tool helps you interpret key metrics and analytics, enabling a deeper understanding of operational patterns. By providing actionable insights, Sage Analytics helps you optimize workflows, increase production efficiency, and ultimately reduce costs.

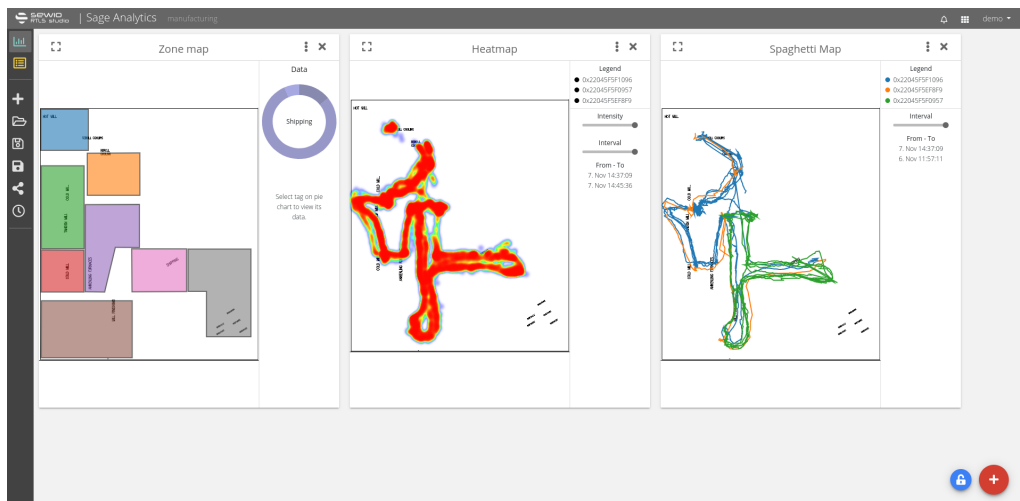
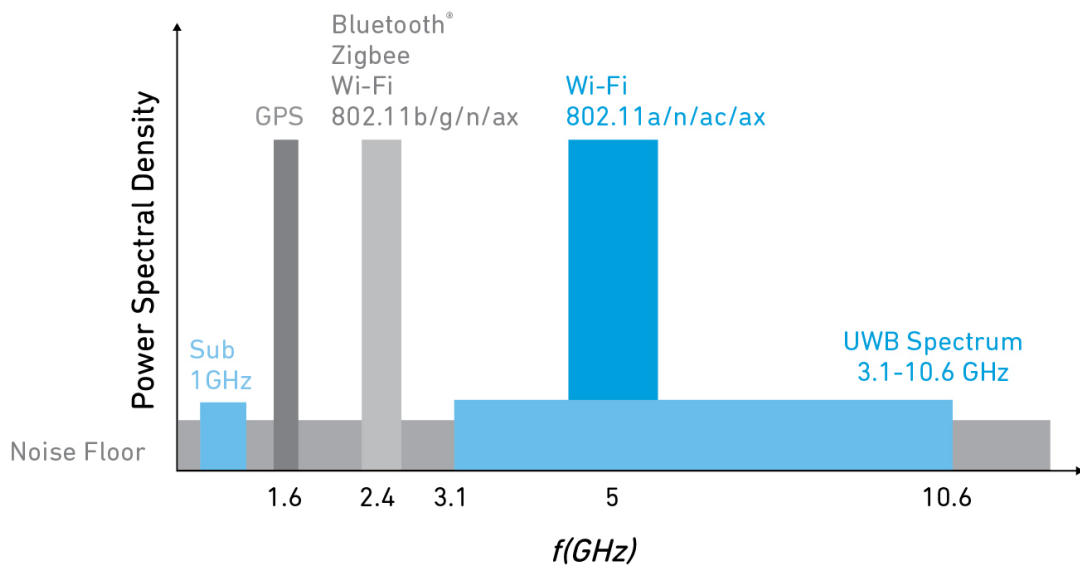


Fig. 1.6: Demo of Sage Analytics [30]

2 UWB technology

UWB is an IEEE 802.15.4a/z standard technology designed for short-range radio communication, characterized by its use of low energy levels combined with a broad bandwidth. By definition, UWB occupies a frequency spectrum that spans more than 500 MHz or at least 20% of the arithmetic center frequency of the operating band. The standardized power level for UWB transmissions is -41.3 dBm/MHz. In 2002, the United States allocated the 3.1–10.6 GHz spectrum for UWB communications, which the WiMedia Alliance subsequently divided into fourteen 500-MHz bands [24].

A key distinction between UWB and traditional radio technologies, such as narrowband and wideband, is the way how information is transmitted. Traditional methods rely on adjusting the signal's power, frequency, or phase, whereas UWB transmits information through pulse-position and time modulation, emitting signal energy in precise time intervals. As a result, UWB signals are time-based rather than frequency-based. The use of short impulses in the time domain for data transmission generates a wide bandwidth.



QORVO

© 2021 Qorvo US, Inc.

Fig. 2.1: Spectral Density for Ultrawide and Narrowband [26]

2.1 Physical principles

UWB offers extremely low latency, measured in nanoseconds, in contrast to Bluetooth, which operates with latencies of a few milliseconds [2]. This capability enables highly accurate and fast time-of-flight (ToF) ranging. Additionally, the UWB operates with a low duty cycle, representing the ratio of impulse time to total transmission time, resulting in significantly lower power consumption compared to conventional radio technologies.

2.1.1 PHY Standards (IEEE 802.15.4a/b/z)

The physical layer of UWB is standardized by IEEE 802.15.4, with several amendments targeting different use cases and improvements. IEEE 802.15.4a introduced support for UWB impulse radio (IR-UWB), enabling precise time-of-flight (ToF) measurements for ranging applications. IEEE 802.15.4b provided clarifications and minor updates. Most notably, IEEE 802.15.4z, which is widely used in modern UWB chips, introduced enhancements in robustness, security (for example, secure ranging) and energy efficiency, making it suitable for commercial and industrial deployments [1, 9].

2.1.2 Power Consumption

UWB communication is very efficient. This efficiency comes from the use of short impulses, approximately 2 ns each, rather than continuously emitting modulated waves. Unlike carrier-based systems such as Bluetooth, UWB operates without a carrier wave (carrier-less), eliminating the additional energy expenditure required for starting, stabilizing, and maintaining an oscillator. Furthermore, the design of UWB transceivers is simpler than that of narrowband or wideband systems, enabling full implementation using complementary metal oxide semiconductor (CMOS) technology [19, 24]. These factors contribute to the lower energy consumption of the UWB in equivalent time periods.

2.1.3 Interference

The interference of ultra-wideband represents a unique case, as the technology shares its bandwidth with commonly used radio technologies such as narrowband and wideband systems. However, because the total transmission power of the UWB, just a few milliwatts, is spread across a wide frequency range (Figure 2.1), its interference with narrowband transmission methods is minimized to an almost negligible

level. The spectral mask introduced by the FCC (Federal Communications Commission) in 2002 mandates that UWB devices operate at the noise floor level (below -41.3 dBm), but also allowing an extensive bandwidth of up to 7.5 GHz. This regulation enables UWB to coexist seamlessly with other radio technologies within the spectrum. UWB transmissions are difficult to detect as their signals are nearly indistinguishable from general background noise [11].

2.1.4 Higher-Level Standards

Beyond the physical and MAC layers, several industry consortiums define higher-level standards and profiles to ensure interoperability and streamline the adoption of UWB in real-world applications.

The **FiRa Consortium** focuses on interoperability between UWB devices, particularly in secure access, indoor positioning, and device-to-device services. FiRa defines profiles and certification procedures to ensure seamless communication between products from different vendors [14].

Omlox is an open standard for real-time locating systems (RTLS), especially in industrial environments. It supports interoperability between different locationing technologies (including UWB) and aims to unify tracking infrastructure between vendors [9, 22].

The **Car Connectivity Consortium (CCC)** develops specifications for digital car keys and short-range communication between smartphones and vehicles. UWB plays a central role in the CCC Digital Key specification by enabling precise, secure ranging to determine the user's proximity to the vehicle [6, 7].

2.2 Ranging and locationing

The distinction between ranging and locationing lies in the scope of information provided: ranging determines the distance to a peer device without offering any additional details on its location, whereas locationing incorporates this additional information [2]. Locationing algorithms typically build upon ranging algorithms. For UWB-enabled devices, common locationing methods can be grouped into four main categories. **Multilateration** determines the position of the device by ranging from multiple anchor points. Another method combines range and **Angle of Arrival** (AoA) measurements from a single anchor point. **Two-Way Ranging** (TWR) calculates location by measuring the round-trip time of signals between the device and an anchor. Similarly, **Time Difference of Arrival** (TDoA) determines the position by analyzing differences in signal arrival times from a single anchor point [15].

2.2.1 Ranging methods

The distance between UWB-enabled devices can be determined by measuring the ToF of the transmitted signal. The distance to a peer device can be accurately measured with line-of-sight up to 200 meters [13]. Multi-path propagation can be an issue, but this can be mitigated by comparing and considering the different ToF trajectories. This consideration improves accuracy by selecting the best (shortest ToF) trajectory to calculate the distance.

Other radio technologies, such as BLE, determine their range using Received Signal Strength (RSS), which is less accurate because signal strength is not standardized and can vary between different devices. RSS-based ranging exhibits limitations in stability when objects obstruct the direct line-of-sight path between devices. This obstruction attenuates the signal, leading to an overestimation of distance. Additionally, the phenomenon of signal superposition, in which reflected BLE signals can either reinforce or diminish RSS values, is known as interference. In contrast, ToF measurements demonstrate greater robustness in such environments. ToF maintains consistency regardless of the strength of the signal, provided that the signal reaches its destination. This characteristic makes ToF more reliable in scenarios with potential obstructions or reflective surfaces [2, 15].

The operational concept between the UWB devices can be seen in Figure 2.2. When UWB devices are within range of each other, they begin the ranging process immediately. After the initiator sends a challenge packet to the responder, the responder timestamps the packet and sends a reply.

The Time of Flight can be calculated by measuring the round-trip time of the challenge-response packets, as described in Equation 2.1. Depending on the context, either the stationary device or the mobile device may compute the range to its peer.

$$ToF = \frac{t_{loop} - t_{reply}}{2} \quad (2.1)$$

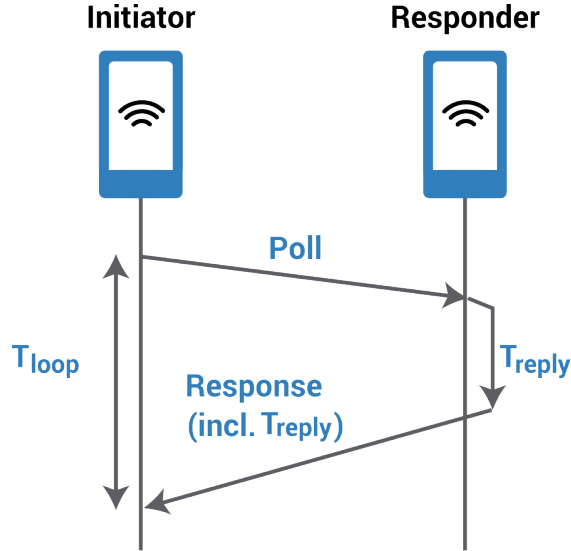


Fig. 2.2: Time of Flight diagram [12]

2.2.2 Locating methods

Two-way Ranging

Two-way Ranging (TWR) works on the principle of measuring the distance between a responder (usually an anchor) and an initiator (a tag) using UWB radio signals. TWR is categorized into two types: Single-sided two-way repositioning (SS-TWR) and double-sided two-way repositioning (DS-TWR). The key difference is that DS-TWR transmits an additional packet, which improves accuracy.

This method measures the distance by calculating the ToF of the signal and multiplying it by the speed of light. Although DS-TWR is more complex, it implicitly corrects clock offset errors, making it more precise, but requires more data packet transmissions, which leads to higher power consumption.

$$ToF_{DS-TWR} = \frac{(T'_1 T'_2) - (T_1 T_2)}{(T'_1 + T'_2 + T_1 + T_2)} \quad (2.2)$$

$$distance = ToF \cdot c \quad (2.3)$$

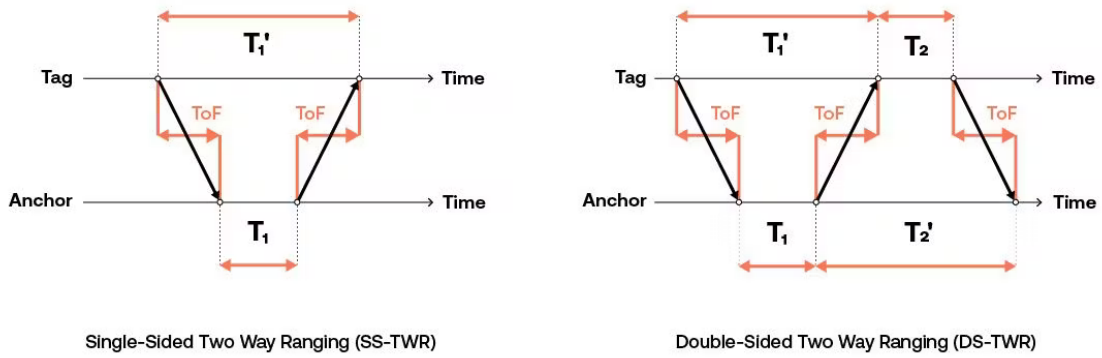


Fig. 2.3: Single-sided two way ranging & double-sided two way ranging [21]

Time Difference of Arrival

The TDoA localization method relies on differences in signal arrival times. The initiator sends a broadcast message (also known as *blink*) containing a timestamp, which is received by multiple anchors. These anchors then relay the information to the locating module, which calculates the tag's position using the multilateration algorithm. Since TDoA is based on the timing differences of the received signals, the anchors must be synchronized to accurately measure the distances [2].

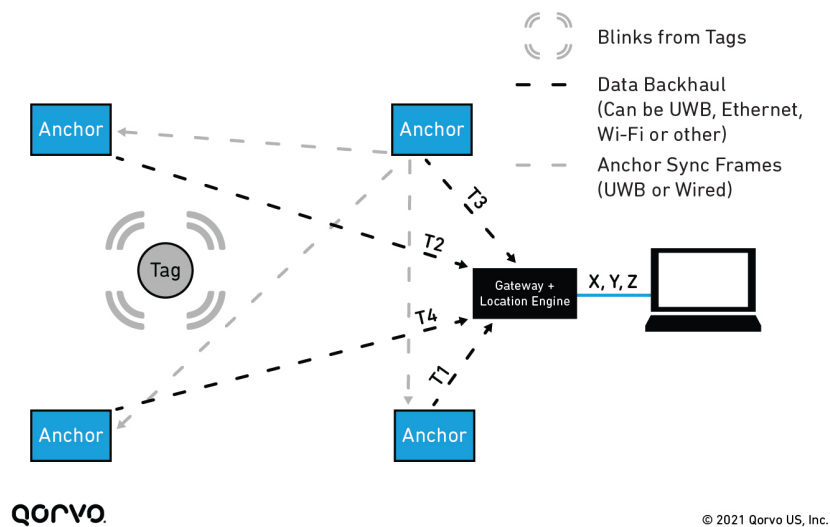


Fig. 2.4: Location determination with TDoA [26]

A **reverse TDoA** system can also be implemented, which operates similarly to GPS. In this setup, anchors transmit synchronized blinks with fixed known offsets to prevent collisions, while tags use TDoA and multilateration algorithms to calculate their location, as illustrated in Figure 2.5.

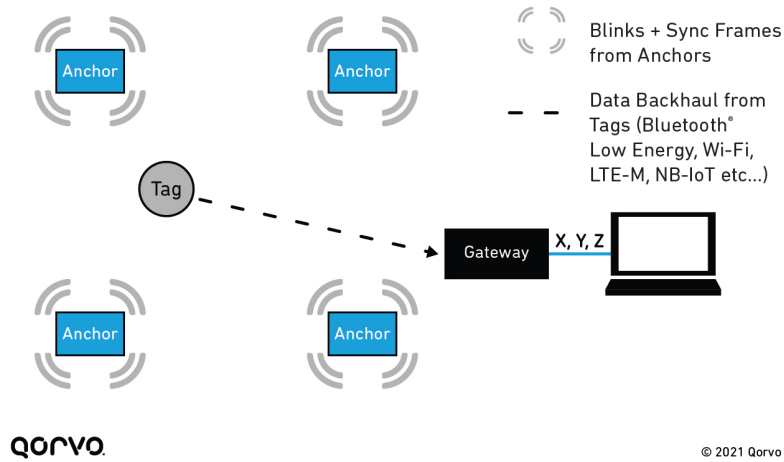


Fig. 2.5: Reverse TDoA [26]

Phase Difference of Arrival and Angle of Arrival

PDoA method allows two devices to determine their relative positions without requiring additional infrastructure by combining distance and directional information. This is important for peer-to-peer applications or for minimizing the infrastructure required for deployment.

One of the devices must have at least two antennas. When this device receives a signal from the other device, it measures the phase difference of the incoming signal at each antenna. Using this information, it can calculate the angle of arrival of the signal. With both directions and distances from the transmitting device, the receiving device can determine its relative position.

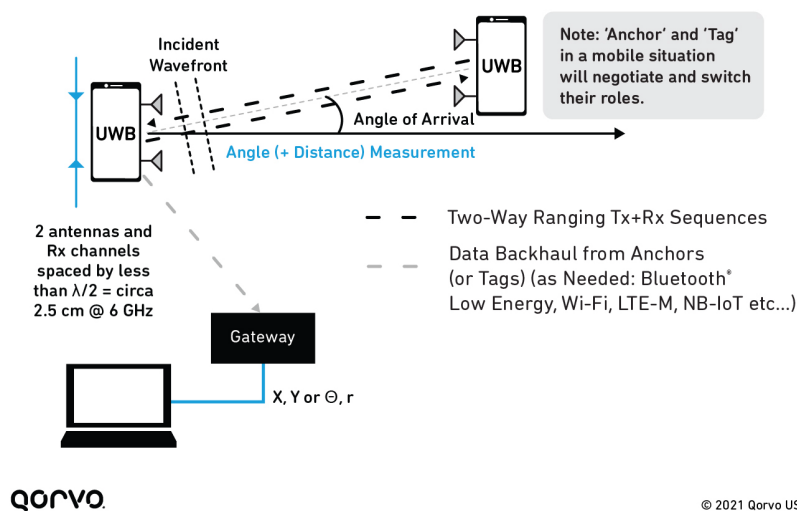


Fig. 2.6: PDoA distance measurement [26]

3 UWB in smartphones

UWB technology has steadily gained popularity in the smartphone market since its introduction in 2019. In 2024, several major manufacturers had incorporated UWB into their flagship devices, offering enhanced capabilities for precise location tracking, seamless device-to-device interactions, and improved connectivity [33].

Apple’s iPhones, starting from the iPhone 11 with their own UWB Apple U1 chip, have consistently featured UWB technology, with the latest iPhone 15 and 16 models having the newer Apple U2 chip.

In the Android ecosystem, Samsung has been at the forefront of UWB adoption, integrating it into their premium Galaxy S, Note, and Fold series. Google has also embraced UWB, incorporating it into their Pixel Pro models, Pixel Fold, and Pixel Tablet. It is important to note that the primary implementation of UWB in these smartphones is based on the FiRa (Fine Ranging) stack, a comprehensive, industry-standard protocol designed to ensure interoperability between devices. The FiRa stack not only facilitates secure and accurate ranging, but also supports various emerging applications such as digital car keys, indoor navigation, and enhanced smart home integration. Other manufacturers like Motorola, Blackview, and Xiaomi have begun to include UWB in select smartphone models [29].

The potential of technology to enable features such as digital car keys, indoor navigation, and enhanced smart home integration is driving its increased adoption, with the expectation that UWB will be present in more than 50% of smartphones by 2028 [35, 36].

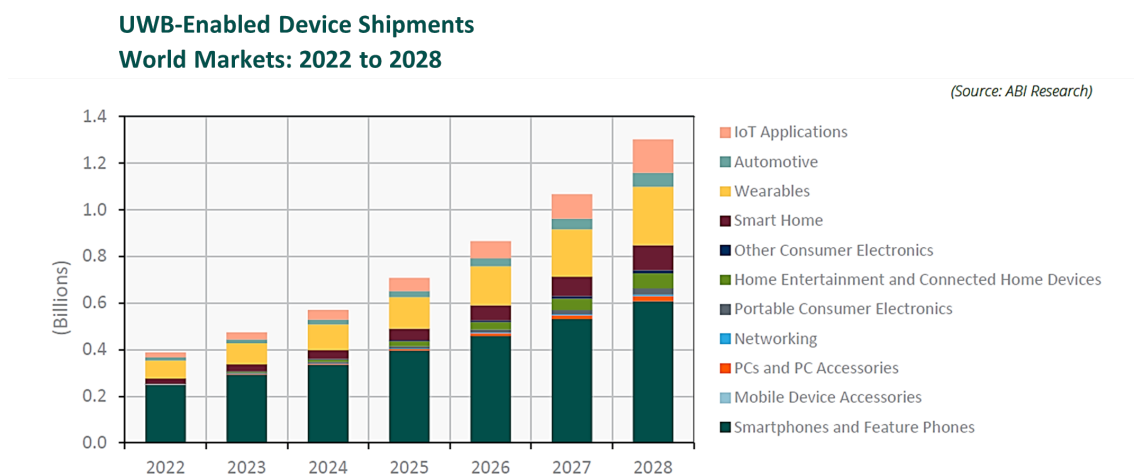


Fig. 3.1: UWB-Enabled device shipments prediction [36]

3.1 Smartphone UWB Chip Manufacturers

The UWB chip market for smartphones is dominated by a mix of established tech giants and specialized semiconductor manufacturers. Apple and Samsung are at the forefront of the industry with their proprietary UWB chips, while NXP Semiconductors and Qorvo provide popular solutions for other Android devices.

Apple has developed its own UWB chips, U1 and U2, which are used in iPhones and other Apple devices. These chips are designed specifically to support Apple’s ecosystem, enabling features such as precise location tracking for AirTags and seamless communication between Apple devices.

Samsung, a leader in consumer electronics, initially used both Qorvo and NXP chips but now manufactures its own UWB chips for its more expensive models, specifically designed to integrate with Samsung’s SmartThings and digital key systems.

Qorvo is a significant player in UWB technology, producing chips for a variety of mobile applications. Its UWB solutions are used by several Android smartphone manufacturers, including Google, to enable advanced connectivity features, particularly for indoor positioning and IoT applications.

NXP Semiconductors produces UWB chips that support digital car keys, secure access control, and other UWB applications. Its technology has been adopted in smartphones, automotive systems, and industrial IoT applications. NXP is a major supplier for companies looking to add UWB to their devices.

Table 3.1: UWB chipsets used in different phones

| OEM | UWB chip | Product |
|---------|---------------|--|
| Apple | U1 | iPhone 11 – 14 |
| | U2 | iPhone 15 – 16 |
| Samsung | NXP SR100T | Note 20 Ultra, S21-23 Ultra & Plus version only, FoldZ 2-5 |
| | Qorvo QM35725 | S24 Ultra & Plus version only |
| Google | Qorvo DW3720 | Pixel 6-7 Pro version only |
| | Qorvo QM35725 | Pixel 8-9 Pro version only, Pixel Fold, Pixel Tablet |
| Xiaomi | NXP SR100T | MIX4 |

3.2 Android UWB module and API

Android Jetpack library is a set of software components, tools, and guidance to help in the development of applications. This library allows developers to build an app based on UWB chips included in smartphones. UWB functionality became accessible with the release of the *core.uwb* library in June 2022, but the technology itself was introduced in 2018 [3]. This library is compatible with devices running Android 12 or newer and supports a range of use cases, including phone-to-phone communication and interactions with Internet of Things (IoT) devices.

With the launch of Android 13 in 2022, a UWB module was introduced. This module includes UWB stacks built above the Hardware Abstraction Layer (HAL) interface, which is based on the FiRa specification. It enables new features and capabilities for UWB chip vendors, expanding the potential applications of UWB technology.

Applications built with this library enable UWB ranging sessions between two devices, with each device assuming the role of either a controller or a controlee. Each role has specific functions, and a Controller can perform ToF measurements between more controlees simultaneously.

The controller defines the communication channel, initiates the session, and configures the parameters to be used. The controlee responds to the controller's calls and returns the signal in a two-way-ranging process as it is shown in Figure 3.2. Before starting a ranging session between two UWB-enabled smartphones, a relative discovery process must be initiated using an Out-of-Band (OoB) mechanism. This mechanism allows UWB devices to locate each other, as alternative discovery methods are not implemented in the Jetpack library. Therefore, Bluetooth-based or WiFi-based discovery is required before initiating a UWB session.

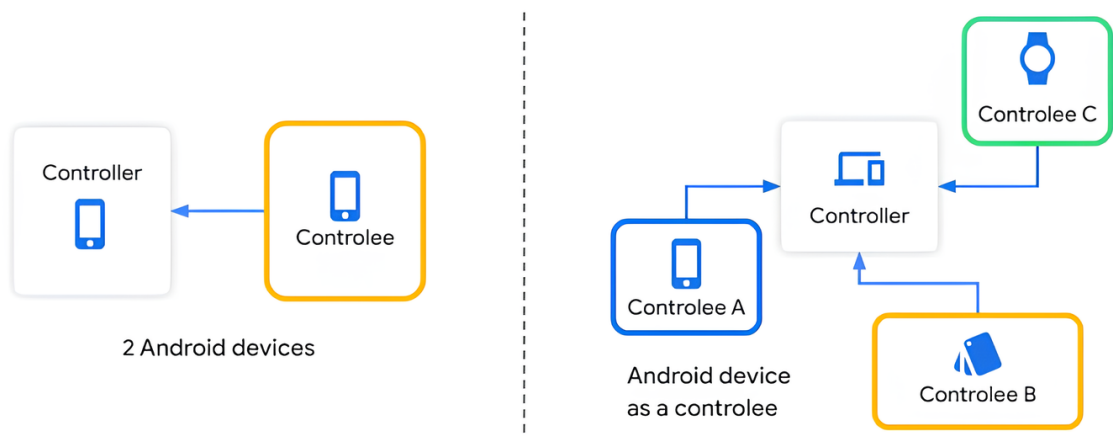


Fig. 3.2: Scheme of UWB connectivity between smartphones [10]

By using the OoB mechanism for connection, all necessary information about the UWB device can be extracted and used to initiate UWB communication. The key data includes the device's UUID, UWB address, and UWB channel. Once these data are exchanged, communication can begin, providing ranging results such as the distance, azimuth, and elevation of the controlee device.

3.3 iOS UWB module and API

Apple provides a proprietary framework called Nearby Interaction (NI), supported in iOS 15, watchOS 8, and later versions, which enables interaction with UWB-enabled devices. Similarly to Android, an initial connection must be established using OoB methods. Nearby Interaction offers four OoB mechanisms: the Core Bluetooth library, LAN (TCP/UDP), a custom server, and the Watch Connectivity framework (specifically used for Apple Watch interactions). In iOS, NI provides both the distance and direction of a peer device, while in watchOS it only provides the distance to a peer device (mainly to an iPhone) [4].

To communicate with third-party devices, Made for iPhone/iPod/iPad (MFi) certification is required. Currently, only selected chips from Qorvo and NXP Semiconductors are certified.

4 Existing applications

UWB technology has gained significant attention for its ability to enable precise location tracking and seamless connectivity. However, only two applications currently available on the Google Play Store or Apple App Store are designed to interact with UWB tags. This chapter explores each of these applications in detail and highlights their features and capabilities.

4.1 Qorvo Connect

The Qorvo Connect app is designed to interact with tags using BLE. Compatible devices can be scanned, and their advertisement data is read and decoded into a human-readable format. Connections can be established with these devices, enabling interaction with their services and characteristics. This application is compatible with both the Android and iOS platforms.

Supported features include dynamic filtering during scanning (based on Received Signal Strength Indication (RSSI), name, and advertisement types) and the ability to review the advertising history. Multiple devices can be connected simultaneously, and detailed information about each device can be accessed through the Device Information Service. Firmware updates for Qorvo devices are supported over-the-air.

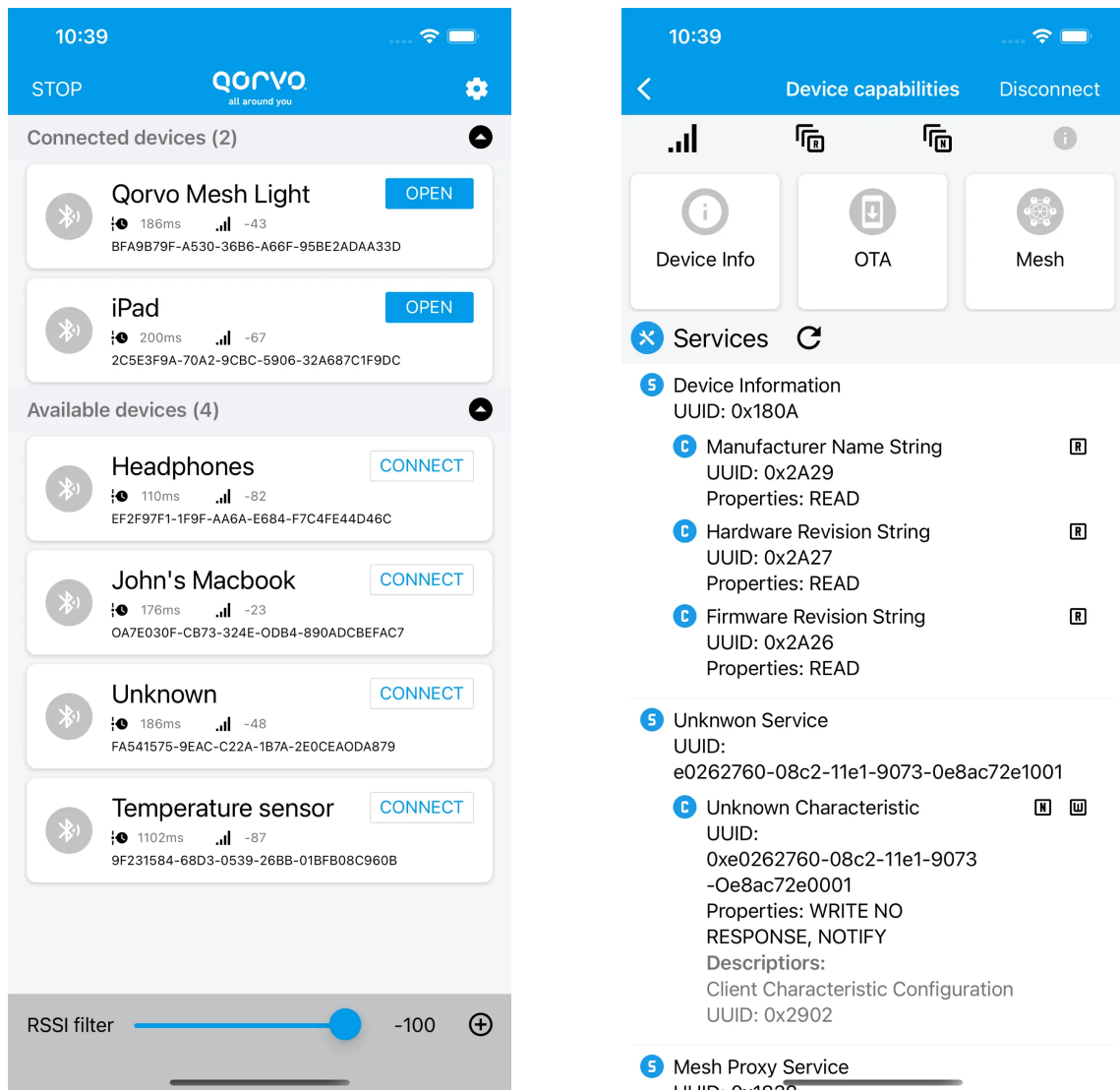


Fig. 4.1: Demo of Qorvo Connect app [27]

4.2 Qorvo Nearby Interaction

The Qorvo Nearby Interaction app is designed to facilitate the evaluation of UWB technology by developers, using the approved development kits that use the Apple NI framework.

With this application, Qorvo UWB development kits can be scanned and connected or disconnected as needed. Ranging information, including distance, angle, and elevation (where supported), is provided simultaneously for all connected development kits. The position of each development kit relative to the iPhone is visually displayed using text, a 2D/3D arrow, or augmented reality (AR) elements.

The app requires an iPhone equipped with a U1/U2 chip (models 11 and newer) and Qorvo UWB development kits. For iPhone 14 and 15 models, elevation is indicated as SAME, ABOVE, or BELOW, replacing the azimuth angle, and the directional arrow is shown in 2D.

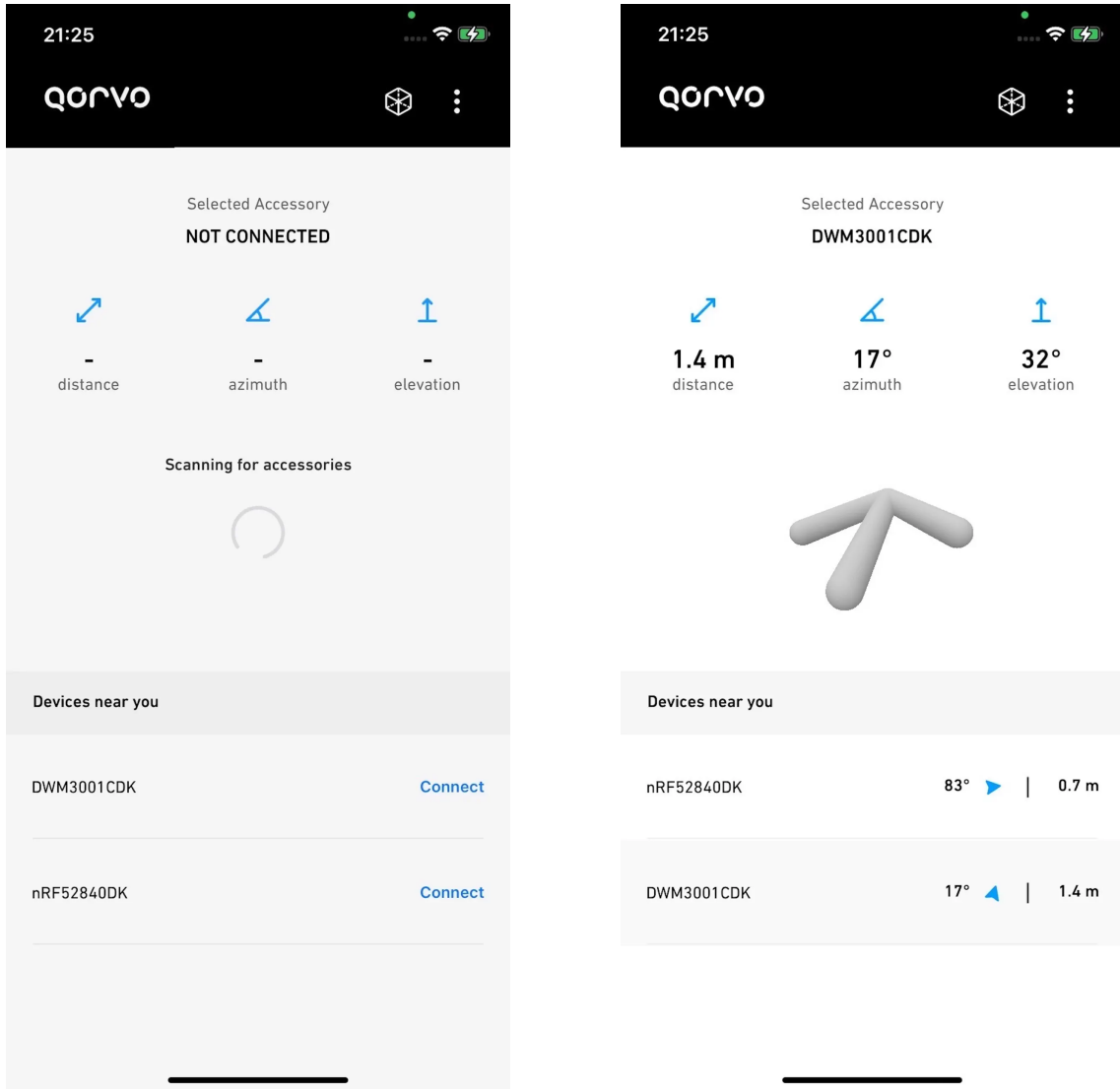


Fig. 4.2: Demo of Qorvo Nearby Interaction app [28]

5 Implementation

This chapter presents a prototype application developed to test UWB technology, with the goal of supporting Sewio RTLS tracking using UWB-enabled smartphones in the future. It provides a comprehensive overview of the development process, including the configuration and setup of Qorvo's UWB module development kit (DWM3001CDK). Key aspects such as concept design and software development are thoroughly documented and described.

5.1 Android application

The goal of this application is to interact with the development kit (referred to as the "devkit" hereafter), which is further detailed in Section 5.2 and also with Sewio RTLS.

The application is capable of determining whether a smartphone supports UWB functionality, obtaining the necessary permissions to perform an OoB BLE scan for nearby devices, establish connections, and facilitate data exchange through reading and writing operations. The acquired data can be used to establish the UWB connection required for subsequent measurements. It can also perform a TWR range with another UWB device.

The first screen of the application is designed to provide basic functionality for interacting with UWB and BLE devices. It features three primary buttons at the top: UWB Support, UWB TWR, and Start Scan. Tapping the UWB Support button displays a toast message that indicates whether the device supports UWB technology, as shown in Figure 5.1. The Start Scan button functions as a toggle to start or stop scanning for nearby BLE devices. The UWB TWR button takes the user to a different screen, where measurements are made.

The scan results are displayed in a list, showing the name (if available) or "Un-named" for devices without a recognizable label. In addition to the device names, their respective signal strengths (in dBm) are displayed, providing an indication of their proximity. Users can tap on any listed device to establish a connection.

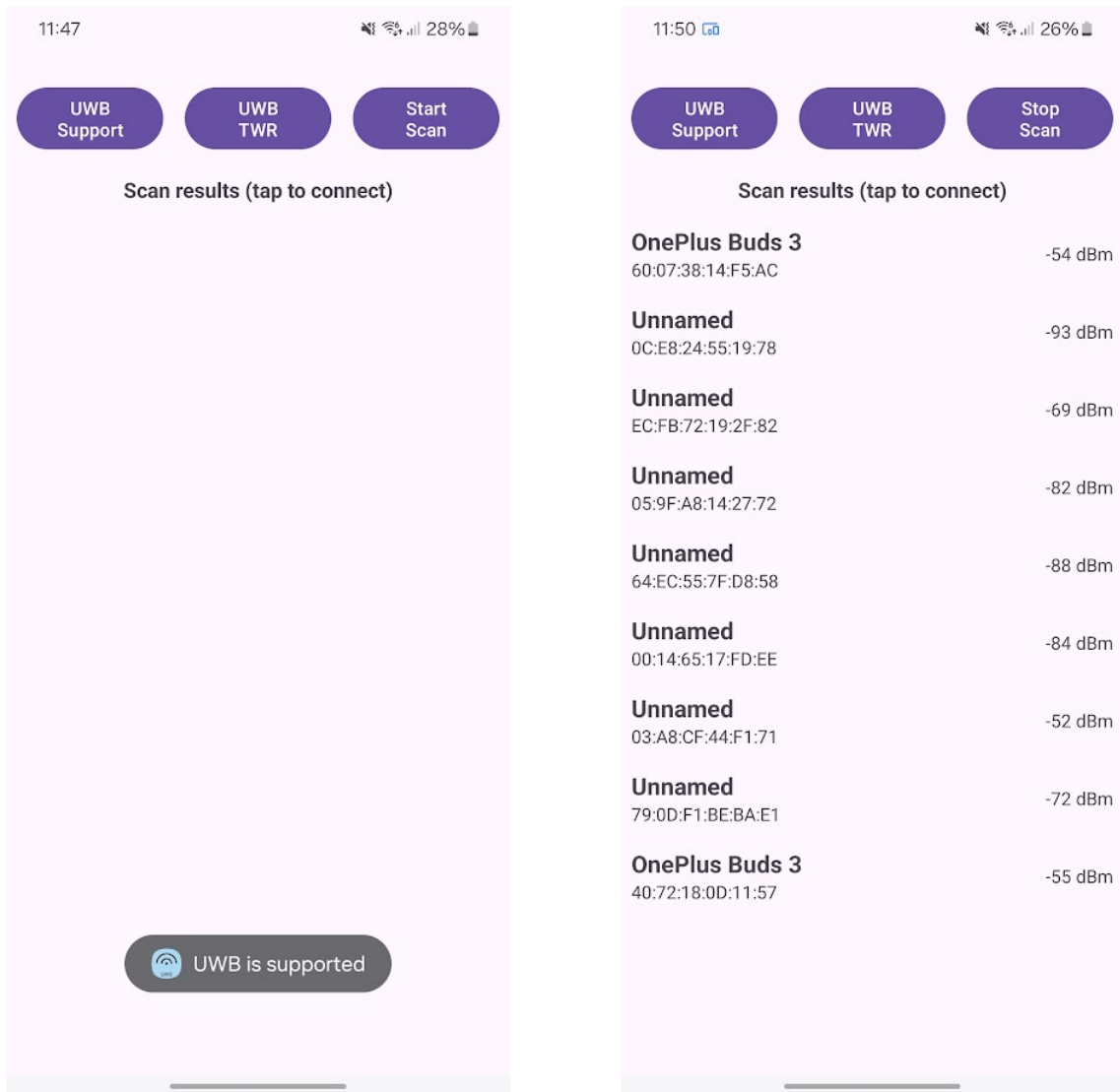


Fig. 5.1: Main screen: Control buttons and device discovery results

The second screen of the application (Fig. 5.2) is displayed after establishing a successful connection to the device. This screen provides advanced functionalities and detailed insight into the interaction between the connected device and the application, serving as a critical tool for the communication and debugging process.

MTU configuration – Allows users to request or change the maximum transmission unit (MTU) size via the text box and the button on the top of the screen.

Device characteristics – Lists device characteristics by UUID with properties (Readable, Writable, Notifiable), allowing interaction with device features.

Interaction log – A log records all communication activities, including data reads, writes, and notifications, along with timestamps. Data are represented in hexadecimal format, providing detailed insight into the interaction process for debugging and verification.

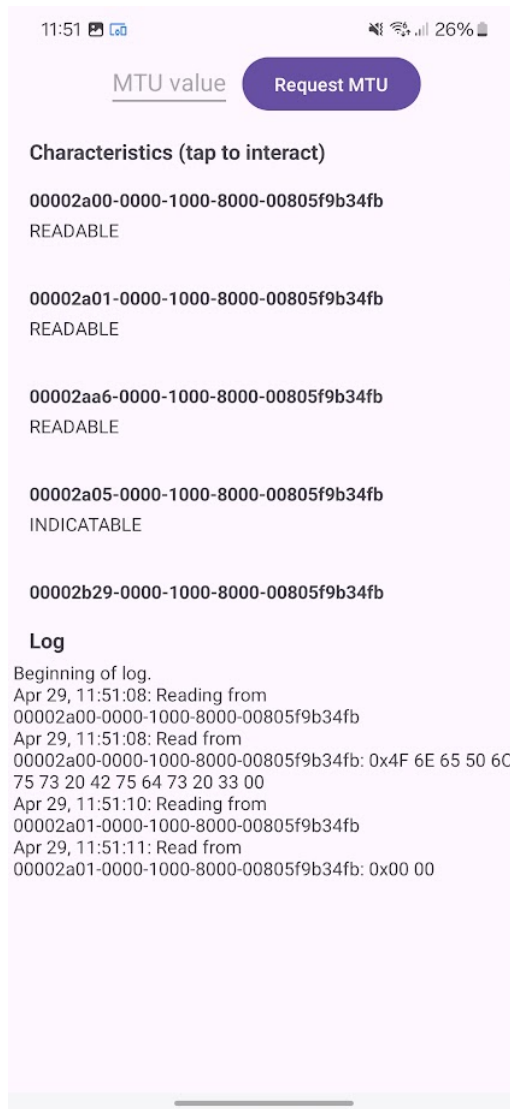


Fig. 5.2: Device interaction screen: connected device characteristics and log

The last and most important screen (Fig. 5.3) is shown after clicking the UWB TWR button on the first screen. This screen is dedicated to initiating and managing a TWR session between UWB devices. At the top is a checkbox, which allows the user to change the devices' role from controllee to controller. Below is a prepare session button that configures the device for the TWR session. The screen displays the local device address and includes a field for entering the destination address. Once a session is prepared, users can begin ranging by pressing the start ranging button (which dynamically changes to "Stop Ranging" when active).

After establishing a session, the real-time ranging data is presented below, including **distance** between devices (in meters), **azimuth** (in degrees), **elevation** (in degrees) and the **elapsed time** of the measurements (in nanoseconds). At the very bottom, a 3D arrow dynamically visualizes the direction toward the second device based on the measurements data.

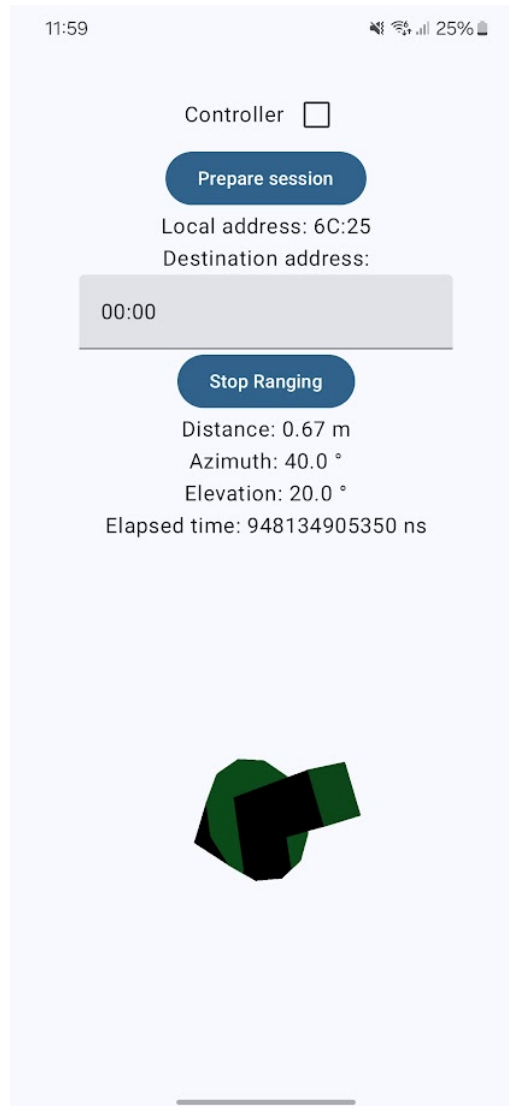


Fig. 5.3: UWB session screen: TWR configuration

5.1.1 Programming

This section provides an explanation of the code developed for this project, including its structure, functionality, and the logic behind key components. Describes the programming languages, frameworks, and tools used, as well as the implementation of core algorithms and processes. In addition, it discusses how the code contributes to achieving the objectives of this study.

The Kotlin programming language was chosen for developing the Android application due to its robust libraries supporting UWB and its status as the primary language for modern Android application development.

Android Studio was selected as the integrated development environment (IDE) for this project due to its comprehensive suite of tools specifically made for Android application development. It offers features such as an intuitive user interface, a powerful code editor with advanced debugging capabilities, seamless integration with the Android SDK, and support for Gradle-based build systems. These functionalities facilitate efficient coding, testing, and deployment, making it an ideal choice for developing robust and scalable Android applications.

The Samsung Galaxy S22+ was selected as the test device for the application due to its support for UWB technology, a critical feature required for the project's functionality. Additionally, the device's advanced hardware capabilities and compatibility with the latest Android features ensure a reliable testing environment for evaluating the application's performance and usability under real-world conditions.

The following directory tree illustrates the layout of the project's files and directories. Each file and directory represents a specific aspect of the functionality, ensuring that the code remains clean and well-organized.

```
/UWBApp
├── /ble
│   ├── BleExtensions.kt
│   ├── BleOperationType.kt
│   ├── ConnectionEventListener.kt
│   └── ConnectionManager.kt
├── /ui.theme
│   ├── Color.kt
│   ├── Theme.kt
│   └── Type.kt
├── BleOperationsActivity.kt
├── CharacteristicAdapter.kt
├── MainActivity.kt
├── ModelRenderer.kt
├── PermissionsHelper.kt
├── ScanResultAdapter.kt
├── UWBRanging.kt
└── UwbTWR.kt
```

UWB Support

The first feature implemented in the application is a UI component that verifies the support for UWB on the device. This feature is made using a button labeled "UWB SUPPORT", which the user can press to determine if the device supports UWB functionality. The logic is implemented as detailed in Listing 5.1.

The *setOnClickListener* method is used to define the behavior of the button when clicked. The application uses the *PackageManager.hasSystemFeature* method to verify if the device supports UWB by checking the *FEATURE_USB* constant. If UWB is supported, a toast message saying "UWB is supported" is displayed to inform the user, otherwise "UWB is not supported" is shown.

```
1 binding.buttonUWB.setOnClickListener{
2     if (packageManager.hasSystemFeature(PackageManager.
3         FEATURE_USB))
4         Toast.makeText(this, "UWB is supported", Toast.
5             LENGTH_LONG).show()
6     else
7         Toast.makeText(this, "UWB is not supported",
8             Toast.LENGTH_LONG).show()
9 }
```

Listing 5.1: Code example of UWB SUPPORT button

The layout of the button is defined in the XML file, as shown in Listing 5.2. The button is positioned centrally within the parent layout using constraints, ensuring a consistent and visually appealing interface. The *android:id* attribute uniquely identifies the button in the layout, and the *android:text* attribute specifies the button label as "UWB SUPPORT". Constraints are applied to align the button with other components and the parent layout.

```
1 <Button
2     android:id="@+id/buttonUWB"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:text="UWB SUPPORT"
6     app:layout_constraintBottom_toTopOf="@+id/scan_button"
7     app:layout_constraintEnd_toEndOf="parent"
8     app:layout_constraintStart_toStartOf="parent"
9     app:layout_constraintTop_toTopOf="parent"/>
```

Listing 5.2: Example of UWB SUPPORT button design

UWB TWR

The second and more significant feature is accessible via the "UWB TWR" button, which is defined similarly to the "UWB SUPPORT" button. When this button is clicked, the application navigates the user to a second screen (Fig. 5.3) using the *setOnClickListener* method, as shown in Listing 5.3.

```
1 binding.buttonUWB_TWR.setOnClickListener{
2     val intent = Intent(this, UwbTWR::class.java)
3     startActivity(intent)
4 }
```

Listing 5.3: Code example of transition to the other screen

This screen layout, defined in the *UWB_TWR.kt* file, is implemented using Jetpack Compose, a modern toolkit that simplifies the development of user interfaces. An example of a composable function displaying a block of text is provided in Listing 5.4.

```
1 @Composable
2 fun PositionText() {
3     val position = uwbRanging.rangingPosition
4     Text("Distance: ␣${position.distance?.value}␣m")
5     Text("Azimuth: ␣${position.azimuth?.value}␣\u00B0")
6     Text("Elevation: ␣${position.elevation?.value}␣\u00B0")
7     Text("Elapsed␣time: ␣${position.elapsedRealtimeNanos}␣
8         ns")
9 }
```

Listing 5.4: Code example of composable element

From this screen, the user can configure and initiate TWR communication with another device without relying on out-of-band (OoB) methods. This is accomplished by specifying the MAC address of a compatible peer device that satisfies the ranging requirements defined in Listing 5.5, which are obtained from the *androidx.core.uwb* library. The implementation of this configuration is located in the *UWB_Ranging.kt* file.

```

1  val partnerParameters = RangingParameters(
2      uwbConfigType = RangingParameters.
3          CONFIG_UNICAST_DS_TWR,
4      sessionKeyInfo = byteArrayOf(0x08, 0x07, 0x01, 0x02,
5          0x03, 0x04, 0x05, 0x06),
6      complexChannel = UwbComplexChannel(9, 9),
7      peerDevices = listOf(UwbDevice(remoteUwbAdr)),
8      updateRateType = RangingParameters.
9          RANGING_UPDATE_RATE_FREQUENT,
10     sessionId = 42,
11     subSessionId = 0,
12     subSessionKeyInfo = null
13 )

```

Listing 5.5: Code example of UWB parameters

The resulting distance, azimuth and elevation measurements are visualized on the screen (Fig. 5.3) and used to dynamically rotate a 3D arrow indicating the direction to the peer device. The arrow is rendered using the *com.google.android.filament* library, with a *.glb* file as the source of the 3D model. Listings 5.6 and 5.7 show the implementation of the arrow rendering and the logic used to update its orientation based on the ranging data. This functionality is implemented in the *ModelRenderer.kt* file.

```

1  private fun createRenderables() {
2      val buffer = assets.open("models/arrow.glb").use {
3          input ->
4              val bytes = ByteArray(input.available())
5              input.read(bytes)
6              ByteBuffer.allocateDirect(bytes.size).apply {
7                  order(ByteOrder.nativeOrder())
8                  put(bytes)
9                  rewind()
10         }
11     }
12     modelViewer.loadModelGlb(buffer)
13     modelViewer.transformToUnitCube()

```

Listing 5.6: Code example of arrow render

```

1 fun updateArrowOrientation(azimuthDegrees: Float,
    elevationDegrees: Float) {
2     val transform = FloatArray(16).also {
3         android.opengl.Matrix.setIdentityM(it, 0)
4         android.opengl.Matrix.rotateM(it, 0, -90f, 0f, 1f
5             , 0f)
6         android.opengl.Matrix.rotateM(it, 0,
7             elevationDegrees, 1f, 0f, 0f)
8         android.opengl.Matrix.rotateM(it, 0,
9             azimuthDegrees, 0f, 1f, 0f)
10    }
11    val rootEntity = modelViewer.asset?.root ?: return
12    val engine = modelViewer.engine
13    val transformManager = engine.transformManager
14    val instance = transformManager.getInstance(
15        rootEntity)
16    transformManager.setTransform(instance, transform)
17 }

```

Listing 5.7: Code example of changing arrow orientation

Bluetooth Low Energy connectivity

The implementation of the BLE connection was adapted from the guide 'The Ultimate Guide to Android Bluetooth Low Energy' by Chee Yi Ong [23]. Modifications were made to suit the requirements of this work.

PermissionsHelper.kt file is designed to manage and check app permissions, particularly related to Bluetooth operations. It includes methods to determine whether specific permissions have been granted. These methods uses Android's *ContextCompat* and *PackageManager* to streamline permission handling in the app, ensuring compatibility with various Android versions. This file is vital for simplifying permission management.

ScanResultAdapter.kt is a class that serves as an adapter for a RecyclerView in the application. The adapter is designed to display a list of BLE scan results. Each scan result is presented in a custom layout, showing the device name, MAC address, and signal strength. A click listener is implemented for each item in the list, allowing user interaction with the scan results.

CharacteristicAdapter.kt works similarly to `ScanResultAdapter.kt`, it is the adapter used to display *BluetoothGattCharacteristic* objects and a click listener function as parameters. Creates and binds `ViewHolder` objects, which represent individual rows in the list. Each row displays the UUID and properties of a characteristic.

BleOperationsActivity.kt file is designed to facilitate BLE operations within an application. Provides a user interface for interacting with their characteristics. The file enables users to read, write, and toggle notifications for BLE characteristics while also supporting the configuration of the MTU.

MainActivity.kt file serves as the central hub of the application. The file handles BLE operations such as scanning for devices, processing scan results, and initiating connections. It also manages permission requests for Bluetooth access, checks if Bluetooth is enabled, and prompts the user to enable it if necessary. The `MainActivity` class implements life cycle management, error handling, and user feedback through toast messages.

ConnectionEventListener.kt file defines a listener class for BLE connection events. It includes callback methods for handling events such as connection setup, disconnection, characteristic and descriptor reads/writes, notifications enabling/disabling, and MTU changes.

ConnectionManager.kt provides the main BLE management logic. It also maintains an operation queue to execute BLE tasks sequentially and manages connected devices' Generic Attribute Profile (GATT) profiles.

BleExtensions.kt file contains utility extensions for BLE-related classes. It includes methods for printing GATT tables, finding characteristics and descriptors, checking properties (e.g. readability or writability), executing writes for characteristics/descriptors, and converting byte arrays to hexadecimal strings.

BleOperationType.kt defines a sealed class hierarchy representing various BLE operations. These operation types are used in the `ConnectionManager`'s operation queue.

5.2 Development kit

The Qorvo DWM3001CDK is a development kit designed to enable engineers to explore and prototype applications utilizing UWB technology. This devkit is fully compliant with the IEEE 802.15.4z standard, ensuring high accuracy, reliability in distance measurement, and secure communication. It features essential capabilities such as BLE connectivity for easy setup, flexible configuration, and support for UWB measurements.

5.2.1 Configuration

The development kit is equipped with an onboard Segger J-Link programmer and has been flashed with a firmware (see Section 5.2.2) using Segger J-Flash Lite software via the micro-USB port. After flashing, the devkit can be configured through the command-line interface (CLI), the included software, or the provided Python scripts.

5.2.2 Firmware

Qorvo Nearby Interaction SDK

The first firmware utilized in this project is a modified version specifically designed to interact with the iOS application, Qorvo Nearby Interaction. Although the firmware can interact with Android, its functionality is currently limited by the absence of a dedicated application for the platform.

The used firmware is `DWM3001CDK-QANI-Freertos_full_QNI_3_0_0`, which includes several modifications compared to the stock release. These include enabling Bluetooth advertising and connectivity and disabling the CLI configuration menu. Bluetooth functionality is essential for the devkit to establish an OoB connection with the smartphone. Disabling the CLI menu is not critical for operation, but it prevents the user from modifying the device settings manually.

The firmware is developed in C, compiled into a binary file, and subsequently flashed onto the devkit. The source code is available for download from the Qorvo website [25], allowing easy access and modification. The firmware can be modified using the SEGGER Embedded Studio software. A significant portion of the firmware's functionality can be adjusted through the use of macros, which serve to enable or disable specific sections of the code. This approach allows for flexible configuration without the need for extensive changes to the underlying code structure. An example of this can be seen in Listing 5.8, where the macro is used to control the BLE advertising feature.

```
1 #ifndef BLE_ADVERTISING_ENABLED
2 #define BLE_ADVERTISING_ENABLED 1
3 #endif
```

Listing 5.8: Macro for enabling BLE advertising

This modified firmware served a dual purpose during development. Firstly, it was used to validate and test the functionality of Qorvo's iOS applications (see Chapter 4), particularly their integration with the Nearby Interaction framework. Secondly, it provided a foundation for experimenting with Bluetooth Low Energy communication in the context of the custom Android application. Although Android

support is limited, the firmware allowed the developer to explore the establishment of BLE connections, service discovery, and characteristic read/write operations. These experiments were important in understanding how the devkit interacts with mobile devices over BLE and in identifying the limitations of the current implementation on Android.

DW3 QM33 SDK

The second firmware used in this project is the original release, version 1.0.2, named DWM3001CDK-DW3_QM33_SDK_UCI-FreeRTOS. This version provides full access to configuration and control features, allowing users to modify parameters and execute functions through the included Python scripts or via the Qorvo One TWR application. This flexibility makes it particularly suitable for development and testing scenarios that require dynamic adjustment of UWB parameters and behavior.

This firmware was used to support the developed Android application (see Section 5.1). The configuration was performed using the provided Python script `run_fira_twr.py`, which was slightly modified to match the specific requirements for Android identified by documentation and debugging. The following changes were made to the script's default parameters:

- `-ranging-span` default changed to 120
- `-mac` and `-dest-mac` modified to accept MAC addresses in the format `XX:XX`
- `-en-rssi` default changed to `True`
- `-preamble-idx` default changed to 9
- `-slots-per-rr` default changed to 6
- `-hopping-mode` default changed to `enabled`

These modifications ensured compatibility with the Android app's configuration and allowed successful establishment of UWB ranging sessions.

The script can be executed from the command line using the following command:

```
1 $ python run_fira_twr.py -p <COMX> --mac 00:00 --dest-mac  
   <XX:XX> [--controlee]
```

Listing 5.9: Example command to run the python script

Here, `<COMX>` specifies the serial port to which the devkit is connected, and `<XX:XX>` represents the MAC address of the Android device, which is generated by the application when the "Prepare session" button is pressed. The optional `-controlee` flag designates the device as a controlee; if omitted, the device acts as a controller by default.

After executing the command and pressing the *Start Ranging* button on the peer device, the UWB ranging session is initiated. Once active, measurement data are streamed through the terminal, as shown in the example output below:

```
1 # Ranging Data:
2   session handle:      4
3   sequence n:         162
4   ranging interval:   120 ms
5   measurement type:   Twr
6   Mac add size:       2
7   primary session id: 0x0
8   n of measurement:   1
9   # Measurement 1:
10  status:              Ok (0x0)
11  mac address:         3f:82 hex
12  is nlos meas:       Unknown
13  distance:           24.0 cm
14  AoA azimuth:        0.0 deg
15  AoA az. FOM:        0.0 %
16  AoA elevation:      0.0 deg
17  AoA elev. FOM:      0.0
18  AoA dest azimuth:   0.0 deg
19  AoA dest az. FOM:   0.0 %
20  AoA dest elevation: 0.0 deg
21  AoA dest elev. FOM: 0.0 %
22  slot in error:      0
23  rssi:               -68.5 dBm
```

Listing 5.10: Sample output from a UWB ranging session

As shown in the output, only the distance and RSSI values contain meaningful data. This is because the devkit is equipped with a single antenna and is therefore incapable of performing Angle of Arrival measurements. In contrast, the Android device has multiple antennas, allowing it to compute azimuth and elevation angles, as shown in Figure 5.3.

5.3 Sewio RTLS integration

Despite the original goal of integrating smartphones with the SEWIO RTLS platform using UWB, several factors made this integration unfeasible. The primary issue is that SEWIO RTLS does not support the TWR method, which is the only method currently supported by smartphones for UWB communication. Another drawback is the limited access to low-level UWB chip features in both Android and iOS operating systems, which prevents the creation of a custom communication layer. Additionally, Android devices introduce specific challenges, such as generating a new MAC address for each session (for security reasons), which leads to unstable and unreliable connections due to constant connecting and disconnecting from the anchors. As a result, the practical part of this thesis shifted focus to the analysis of available UWB technologies and the development of a prototype application that demonstrates UWB capabilities on smartphones, without direct integration into the SEWIO RTLS.

5.3.1 Future Directions

The current state of UWB technology in smartphones and other consumer devices is still in its early stages, but its unique properties, such as high accuracy, low power consumption, and secure ranging, position it as a compelling enabler for real-time location systems. As the technology matures, several trends and developments suggest a strong and expanding role for UWB in both consumer and industrial applications.

One of the most promising developments is the convergence of interoperability standards, notably the collaboration between the FiRa Consortium and the omlox initiative. This cooperation aims to unify the behavior of devices and systems between different vendors, paving the way for the seamless integration of UWB-enabled devices into RTLS platforms. At the protocol level, the upcoming IEEE 802.15.4ab amendment is expected to introduce further enhancements to the robustness and scalability of UWB communications, particularly in dense or complex environments¹.

In support of this trend, recent ETSI technical reports recognize UWB as a key driver of indoor positioning in industrial environments, highlighting its potential in use cases where reliability, precision and security are critical².

¹<https://www.ieee802.org/15/pub/TG4ab.html>

²https://www.etsi.org/deliver/etsi_tr/103800_103899/103862/01.01.01_60/tr_103862v010101p.pdf

From a market perspective, UWB technology is steadily gaining momentum. Analysts project continued growth in the coming years, largely driven by RTLS applications in sectors such as manufacturing, logistics, and smart infrastructure. Although adoption is still in its early stages, the market outlook remains strong, as several reports emphasize the increasing demand for high-precision indoor positioning, particularly in industrial deployments³⁴⁵.

In summary, despite its early adoption phase, UWB is gaining traction across both consumer and industrial markets. The growing alignment around technical standards and the expanding range of real-world applications point to a promising future. UWB is increasingly seen as a foundational technology for the next generation of indoor positioning and smart systems.

³<https://www.marketsandmarkets.com/Market-Reports/ultra-wideband-market-200905786.html>

⁴<https://www.grandviewresearch.com/industry-analysis/ultra-wideband-market-report>

⁵<https://www.databridgemarketresearch.com/reports/europe-ultra-wideband-uwband-market>

Conclusion

This thesis analyzed the current use of Ultra-Wideband (UWB) technology in smartphones and its potential applications, particularly in real-time locating systems (RTLS). It introduced the core concepts of locationing and UWB, followed by an explanation of the origins, functionality, and integration of the technology into smartphones. The study also provided an overview of key manufacturers and their capabilities, along with a comparison of existing UWB-based applications.

The research examined how UWB is implemented in mobile operating systems, highlighting advantages such as high-precision location tracking and secure proximity-based communication. A comparison of the Android and Apple platforms revealed different approaches in terms of developer access and system-level support. These findings offer valuable insight into the growing impact of UWB on the mobile industry.

In the practical part of the thesis, a prototype Android application was developed to demonstrate UWB-based locationing. The application checks for UWB support and establishes a Bluetooth Low Energy connection with the UWB tag as a preparatory step for potential Out-of-Band communication in future versions. However, in its current state, UWB two-way ranging is initiated directly using the tag's MAC address, independently of BLE. The app also features a 3D visualization module that renders real-time spatial data, such as distance, azimuth, and elevation, to help users intuitively interpret the positioning of the tag.

Although one of the original goals was to integrate the prototype with the Sewio RTLS system, this proved to be unfeasible due to software and hardware limitations. As a result, the implementation focused on showcasing the capabilities of UWB communication directly on smartphones.

In summary, this work demonstrates the feasibility of UWB-based locationing on consumer devices while highlighting the current barriers to integrating mobile platforms with established RTLS infrastructure. The developed prototype offers a strong foundation for further exploration and contributes both theoretical and practical value to the advancement of UWB technology.

Bibliography

- [1] IEEE Standard for Low-Rate Wireless Networks—Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, 2020, p. 1–174. [Accessed: 19. 5. 2025].
- [2] ALARIFI, A.; AL SALMAN, A.; ALSALEH, M.; ALNAFESSAH, A.; AL HADHRAMI, S. et al.: Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, may 2016, vol. 16, no. 5, p. 707. Available at: <https://doi.org/10.3390/s16050707>. [Accessed: 20. 4. 2025].
- [3] ANDROID DEVELOPERS: *UWB (Ultra-Wideband) | Android Developers* <https://developer.android.com/jetpack/androidx/releases/core-uwband>. 2022. [Accessed: 29. 4. 2025].
- [4] APPLE INC.: *Using Nearby Interaction* <https://developer.apple.com/documentation/nearbyinteraction>. 2023. [Accessed: 29. 4. 2025].
- [5] BARNWAL, R. P.; ROY, P. and PAL, P. K.: Experimental evaluation of indoor localization methods for industrial IOT environment. *Journal of Scientific and Industrial Research*. NIScPR-CSIR, India, march 2022, no. 81, p. 294–307. ISSN 0975-1084. Available at: <http://nopr.niscpr.res.in/handle/123456789/59308>. [Accessed: 10. 10. 2024].
- [6] BUSINESS WIRE: Car Connectivity Consortium and FIRA Consortium partner on UWB technology used in the CCC Digital Key. *Business Wire*, november 2023. Available at: <https://www.businesswire.com/news/home/20231116765001/en/Car-Connectivity-Consortium-and-FiRa-Consortium-Partner-on-UWB-Technology-Used-in-the-CCC-Digital-Key>. [Accessed: 19. 5. 2025].
- [7] CAR CONNECTIVITY CONSORTIUM: *CCC Digital Key: The Future of Vehicle Access*. Whitepaper v2.1. Car Connectivity Consortium, 2022. Available at: https://carconnectivity.org/wp-content/uploads/2022/12/CCC_Digital_Key_Whitepaper_Approved_v2-1.pdf. [Accessed: 19. 5. 2025].
- [8] CHANTAWEE SOMBOON, W.; SUWATTHIKUL, C.; MANATRINON, S.; ATHIKULWONGSE, K.; KAEMARUNGI, K. et al.: On performance study of UWB real time locating system. In: *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. 2016, p. 19–24. [Accessed: 20. 4. 2025].

- [9] COPPENS, D.; SHAHID, A.; LEMEY, S.; VAN HERBRUGGEN, B.; MARSHALL, C. et al.: An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions. *IEEE Access*, 2022, vol. 10, p. 70219–70241. [Accessed: 19. 5. 2025].
- [10] DI PIETRA, V. and DABOVE, P.: Recent advances for UWB ranging from Android Smartphone. In: *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2023, p. 1226–1233. Available at: <https://ieeexplore.ieee.org/abstract/document/10140133>. [Accessed: 22. 11. 2024].
- [11] FEDERAL COMMUNICATIONS COMMISSION: *Revision of Part 15 of the Commission’s Rules Regarding Ultra-Wideband Transmission Systems* <https://www.fcc.gov/document/revision-part-15-commissions-rules-regarding-ultra-wideband>. 2002. ET Docket No. 98-153, First Report and Order. [Accessed: 20. 4. 2025].
- [12] FIRA CONSORTIUM: *How UWB Works*. 2021. Available at: <https://www.firaconsortium.org/discover/how-uwb-works>. [Accessed: 18. 11. 2024].
- [13] FIRA CONSORTIUM: *What UWB Does*. 2021. Available at: <https://www.firaconsortium.org/discover/what-uwb-does>. [Accessed: 18. 11. 2024].
- [14] FIRA CONSORTIUM: *FIRA Consortium unveils FIRA Core 3.0 specifications and certification Program*. March 2025. Available at: <https://www.firaconsortium.org/news/press-releases/2025/01/fira-consortium-unveils-fira-core-3-0-specifications-and-certification>. [Accessed: 19. 5. 2025].
- [15] GEZICI, S.; TIAN, Z.; GIANNAKIS, G.; KOBAYASHI, H.; MOLISCH, A. et al.: Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 2005, vol. 22, no. 4, p. 70–84. Available at: <https://doi.org/10.1109/MSP.2005.1458289>. [Accessed: 20. 4. 2025].
- [16] HEINRICH, A.; KROLLMANN, S.; PUTZ, F. and HOLLICK, M.: *Smartphones with UWB: Evaluating the Accuracy and Reliability of UWB Ranging*. 2023. Available at: <https://arxiv.org/abs/2303.11220>. [Accessed: 10. 10. 2024].
- [17] HUANG, H. and GARTNER, G.: A Survey of Mobile Indoor Navigation Systems. In: GARTNER, G. and ORTAG, F., ed.: *Cartography in Central and Eastern Europe: CEE 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 305–319. ISBN 978-3-642-03294-3. Available at: https://doi.org/10.1007/978-3-642-03294-3_20. [Accessed: 10. 10. 2024].

- [18] KALYANAM, A. K.: Real-Time Location Systems (RTLS): Harnessing IoT for Enhanced Tracking and Efficiency, november 2020, vol. 2. Available at: <https://www.ijfmr.com/papers/2020/6/23454.pdf>. [Accessed: 20. 4. 2025].
- [19] KSHETRIMAYUM, R. S.: An introduction to UWB communication systems. *IEEE Potentials*, march 2009, vol. 28, no. 2, p. 9–13. Available at: <https://ieeexplore.ieee.org/document/4803808>. [Accessed: 18. 11. 2024].
- [20] MAUTZ, R.: *Indoor positioning technologies*. Zurich, 2012. Habilitation Thesis. ETH Zurich. Available at: <https://doi.org/10.3929/ethz-a-007313554>. [Accessed: 20. 4. 2025].
- [21] NOTANOTHERONE: *UWB: capabilities and Application Perspectives*. June 2023. Available at: <https://hackernoon.com/uwb-capabilities-and-application-perspectives>. [Accessed: 20. 11. 2024].
- [22] OMLOX: *The world's first open locating standard*. Available at: <https://omlox.com/omlox-explained>. [Accessed: 19. 5. 2025].
- [23] ONG, C. Y.: *The Ultimate Guide to Android Bluetooth Low Energy | Punch Through*. April 2024. Available at: <https://punchthrough.com/android-ble-guide/>. [Accessed: 10. 11. 2024].
- [24] OPPERMAN, I.; HÄMÄLÄINEN, M. and IINATTI, J.: *UWB: Theory and Applications*. John Wiley & Sons, october 2004. ISBN 0-470-86917-8.
- [25] QORVO INC.: *DWM3001CDK Ultra-Wideband (UWB) Module Development Kit*. Available at: <https://www.qorvo.com/products/p/DWM3001CDK>. [Accessed: 10. 11. 2024].
- [26] QORVO INC.: Getting Back to Basics with Ultra-Wideband (UWB). *White Paper*, may 2021. Available at: <https://www.qorvo.com/resources/d/qorvo-getting-back-to-basics-with-ultra-wideband-uwband-white-paper>. [Accessed: 6. 11. 2024].
- [27] QORVO INC.: *Qorvo Connect. Mobile app. Version 1.2.3*. 2024. Available at: <https://apps.apple.com/us/app/qorvo-connect/id1588263604>. [Accessed: 1. 12. 2024].
- [28] QORVO INC.: *Qorvo Nearby Interaction. Mobile app. Version 1.3.4*. 2024. Available at: <https://apps.apple.com/us/app/qorvo-nearby-interaction/id1615369084>. [Accessed: 1. 12. 2024].

- [29] SCHNETZER, S.: *Utilisation of Ultra-Wideband for Smartphone Localisation*. 2022. Bachelor thesis. Salzburg University of Applied Sciences. Available at: https://www.researchgate.net/publication/361053274_Utilisation_of_Ultra-Wideband_for_Smartphone_Localisation. [Accessed: 6. 11. 2024].
- [30] SEWIO NETWORKS S.R.O.: *RTLS Studio (version 3.3.0)*. 2024. Available at: <https://demo.sewio.net/>. Screenshots by the author. [Accessed: 3.11.2024].
- [31] SIMEK, M.: *Real-Time Location System (RTLS) on ultra-wideband - SEWIO RTLS*. January 2024. Available at: <https://www.sewio.net/real-time-location-system-rtls-on-uwband/>. [Accessed: 10. 10. 2024].
- [32] UY, K. and DAVIDSON, E.: *RTLS in Healthcare: Comparing Real-Time Location Systems*. May 2024. Available at: <https://www.tagnos.com/rtls-in-healthcare-comparing-real-time-location-systems>. [Accessed: 10. 10. 2024].
- [33] VAN DE VELDE, S.: *The state of UWB in 2024*. February 2024. Available at: <https://www.pozyx.io/newsroom/the-state-of-uwband>. [Accessed: 12. 11. 2024].
- [34] VAN DIGGELEN, F. and PER, E. K.: The World's first GPS MOOC and Worldwide Laboratory using Smartphones. In: *Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*. September 2015, p. 361–369. Available at: <https://www.ion.org/publications/abstract.cfm?articleID=13079>. [Accessed: 10. 10. 2024].
- [35] ZANELLA, G.: *UWB market consumer devices trends*. May 2024. Available at: <https://technotrend.substack.com/p/uwb-market-consumer-devices-trends>. [Accessed: 12. 11. 2024].
- [36] ZIGNANI, A. and TOMSETT, S.: *Ultra-Wideband (UWB) For The IoT—A Fine Ranging Revolution*. ABI Research, june 2021. Available at: [https://f.hubspotusercontent10.net/hubfs/6705264/Marketing/Whitepapers/Ultra-Wideband%20\(UWB\)%20For%20The%20IoT/ABI%20Research_Ultra-Wideband%20\(UWB\)%20For%20The%20IOT.pdf](https://f.hubspotusercontent10.net/hubfs/6705264/Marketing/Whitepapers/Ultra-Wideband%20(UWB)%20For%20The%20IoT/ABI%20Research_Ultra-Wideband%20(UWB)%20For%20The%20IOT.pdf). [Accessed: 12. 11. 2024].

Symbols and abbreviations

| | |
|---------------|-------------------------------------|
| devkit | Development kit |
| GATT | Generic Attribute Profile |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| HAL | Hardware Abstraction Layer |
| IDE | Integrated Development Environment |
| IoT | Internet of Things |
| IPS | Indoor Positioning System |
| MFi | Made for iPhone/iPod/iPad |
| MTU | Maximum Transmission Unit |
| NI | Nearby Interaction |
| OoB | Out Of Band |
| OPS | Outdoor Positioning System |
| RF | Radio Frequency |
| RFID | Radio Frequency Identification |
| RSSI | Received Signal Strength Indication |
| RTLS | Real Time Locating System |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| USB | Universal Serial Bus |
| UUID | Universally Unique Identifier |
| UWB | Ultra-wideband |

A Content of the electronic attachment

Due to its size, the full electronic attachment is not included directly in the submission. Instead, it is accessible via a link provided in a text file named `download_link.pdf`, which is the only file uploaded to the system. The downloadable archive consists of three main folders: `application`, `firmware`, and a compressed `python_script`, along with the thesis in PDF format. The `application` folder includes `UwbApp.zip`, which contains the complete source code of the Android application, and `UwbApp.apk`, the compiled installation file. The application can be opened and built using Android Studio version 2024.2.1 Patch 2, requires Java 17, and targets Android SDK version 34 (Android 14). A UWB-enabled smartphone is necessary to run the application. The `firmware` folder contains the file `DWM3001CDK-DW3_QM33_SDK_UCI-FreeRTOS.hex`, which is the firmware for the Qorvo DWM3001CDK development kit. It can be flashed using SEGGER's J-Flash Lite utility. The archive also includes `python_script.zip`, which contains a Python script and its dependencies, recommended to be used with Python version 3.10. The `thesis.pdf` file provides the full text of the thesis.

```
/. .....root of the attached archive
├── application
│   ├── UwbApp.zip .....source code
│   └── UwbApp.apk ..... application installation file
├── download_link.txt .....link to download the attachment
├── firmware
│   └── DWM3001CDK-DW3_QM33_SDK_UCI-FreeRTOS.hex ....DWM3001CDK firmware
├── python_script.zip .....python script and its dependencies
└── thesis.pdf
```

The file `download_link.pdf` (uploaded with this thesis) contains the URL to download this archive.