Precise Indoor Positioning Using Ultra-Wideband

Walter Mayer*, Szilárd Farkas*, Csaba Sulyok*, Dávid Szopos†, Lóránt-Tibor Fazakas†
*Faculty of Mathematics and Computer Science, Babeş–Bolyai University
RO-400084 Cluj-Napoca, Romania
E-mail: {walter.mayer, szilard.farkas}@stud.ubbcluj.ro, csaba.sulyok@ubbcluj.ro
†Codespring
RO-400458 Cluj-Napoca, Romania
E-mail: {szopos.david, fazakas.lorant}@codespring.ro

***Abstract*—In recent years, Ultra-Wideband (UWB) technology has become increasingly prominent in the field of indoor positioning systems due to its high accuracy and reliability. This paper presents a UWB-based system that demonstrates the advantages of the technology over other wireless solutions not only in theory but in practice as well.**

It is aimed at professionals and researchers interested in communication technologies and IoT systems. The paper details the system architecture, including hardware and software components, data transmission methods, and the client-server infrastructure. It shows how UWB devices can be integrated and deployed into a scalable and efficient communication system that enables real-time positioning. It also analyzes the advantages and limitations of UWB, such as accuracy, interference handling, and energy consumption.

***Index Terms*—Ultra-Wideband; indoor positioning; GPS alternative; telecommunication**

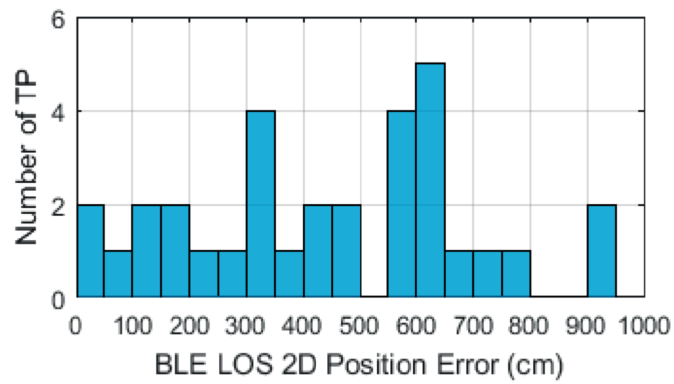

Fig. 1: BLE error distribution

I. Introduction

Over the past 50 years, Global Positioning Systems (GPS) have evolved from military tools to essential components of daily life [1]. Billions of people rely on the navigation and location services it provides on a daily basis. While GPS has had a revolutionary impact, it also comes with several limitations, especially regarding signal interference [2], [3]. Since GPS relies on clear line-of-sight between satellites and receivers, its signals can easily be disrupted. In urban environments, tall buildings, tunnels, or dense vegetation can obstruct, reflect, or distort signals, leading to inaccuracies or even total signal loss. Because GPS is primarily optimized for outdoor use, it performs poorly indoors, where walls and other obstacles further weaken the signals, often resulting in unreliable results [4].

Current indoor positioning systems mainly rely on Wi-Fi, Bluetooth Low Energy (BLE), and RFID-based technologies. BLE [5] is low power and widely available, but typically offers only 4-5 meter accuracy, and is sensitive to obstacles and interference [6] (see Fig. 1). Wi-Fi-based methods such as fingerprinting or CSI-based approaches can achieve better accuracy, but come with calibration and storage overheads, and are difficult to scale across large areas [7], [8]. RFID is cost-effective and easy to deploy, but typically requires line-of-sight (LoS) and struggles with complex indoor obstacles [9]–[11].

The current project introduces a novel solution to precise indoor positioning, leveraging the advantages of the emerging Ultra-Wideband (UWB) [6], [12], [13] technology. UWB offers centimeter-level accuracy in complex environments, it is less sensitive to multipath interference and handles non-line-of-sight (NLoS) situations effectively (see Fig. 2). Furthermore, it supports low-latency, stable communication, making it ideal for applications where precision and reliability are critical. Although UWB's adoption is currently limited, its technological benefits position it as a promising alternative to traditional indoor positioning systems.

The core of the associated application is built upon the Qorvo DWM1001 product family and its accompanying software toolkit, bridging UWB communication with more abstract layers. This paper provides an overview of the UWB-based positioning system, detailing its architecture in the following structure: Section II introduces the key features of UWB technology, Section III focuses on the application architecture, with particular emphasis on the firmware functionalities and the client-server model. Section IV elaborates on the technologies used, while Section V examines the practical application of the system.

II. The UWB Technology from a Top-Down Perspective

Ultra-Wideband (UWB) is a short-range wireless communication technology defined by high bandwidth (>500 MHz), low energy consumption, strong security, high accuracy, and minimal interference [12]. Unlike traditional methods, UWB operates over a wide frequency range, making

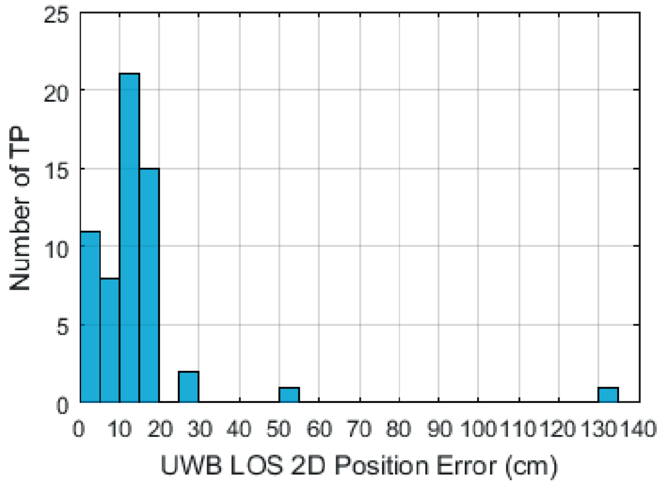

Fig. 2: UWB error distribution

it highly resistant to interference and ideal for positioning. Although UWB has existed since the 1960s, commercial adoption began in 2019, and it has since been integrated into modern smartphones to support asset tracking.

A. Main Application Areas

UWB technology has seen growing adoption in domains where high-precision, short-range communication is essential. In industrial and logistics environments, it enables real-time asset tracking of thousands of items simultaneously [12]. Automotive systems use UWB for secure, keyless entry in premium vehicles, offering improved resistance to signal manipulation. Smart home setups also benefit from UWB's spatial awareness, allowing automation based on user proximity. In healthcare, wearable UWB devices support patient monitoring and remote diagnostics [14].

B. UWB Positioning Methods

A key advantage of UWB is its ability to operate where traditional positioning technologies fail. Unlike BLE, it avoids interference with other systems, improving reliability.

Two primary methods are used for positioning:

- *Time of Flight (ToF):* Measures the time for a signal to travel from transmitter to receiver, via single- or double-sided exchange.
- *Angle of Arrival (AoA):* Uses multiple antennas and phase difference (PDoA) to estimate the signal's arrival angle, enabling 2D or 3D localization.

Most systems use *tags* (receivers) and multiple *anchors* (transmitters) to compute the tag's precise location.

C. The UWB Superframe Structure

UWB communication uses a deterministic time-division multiple access (TDMA) system, built on a 100ms cyclic superframe (see Fig. 3). Key components include:

- *30 Transmitter Message Slots (BCN):* Anchors broadcast their position and network status.

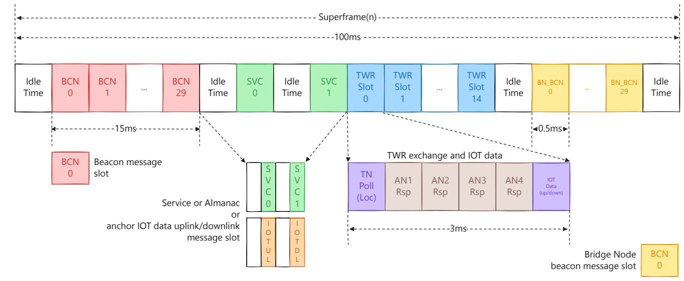

Fig. 3: UWB superframe structure

- *2 Service Slots (SVC):* Used for Almanac messages and IoT data exchange.
- *15 Two-Way Ranging Slots (TWR):* For measuring distances between tags and anchors.
- *30 Bridge Node Beacon Slots (BNB):* Bridge nodes advertise IoT data availability.
- *Guard/Idle Time:* Absorbs timing errors and ensures frame stability.

Positioning typically uses SS-TWR (Single-Sided Two-Way Ranging), which enables tags and anchors to exchange up to 34 bytes securely. This data is transmitted via Bluetooth or UART to a connected device.

All communication is synchronized to the superframe. The initiator anchor controls timing, and other nodes follow the schedule. Tag devices perform ranging in their assigned slots, initiated via Group Poll messages and confirmed with anchor responses. Reserved slots prevent collisions.

This approach supports accurate, scalable positioning, with periodic slot allocation allowing many tags to operate simultaneously. Strict temporal segmentation ensures predictable behavior, even in dynamic or congested networks.

III. Application Architecture

This section outlines the hardware and software components of the application. It describes the physical devices, then covers the software architecture, firmware roles, localization mechanisms, and network scalability. Server and client-side elements are discussed, followed by the communication layer connecting them.

A. Hardware Components

The core of the system is built on DWM1001 modules by Qorvo[1] (formerly Decawave), which integrate the DWM1000 UWB transceiver and the `nRF52832` SoC. The transceiver performs precise Two-Way Ranging (TWR), while the SoC manages BLE communication and overall module control.

The system is modular and hierarchical, with defined communication roles. This decentralized design improves fault tolerance, allowing devices to fail without disrupting the full network, and supports scalable deployment.

[1]Product page: https://www.qorvo.com/products/p/DWM1001-DEV

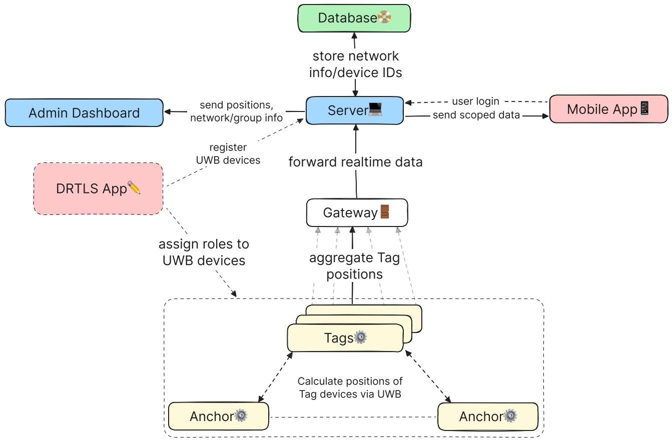

Fig. 4: The system architecture

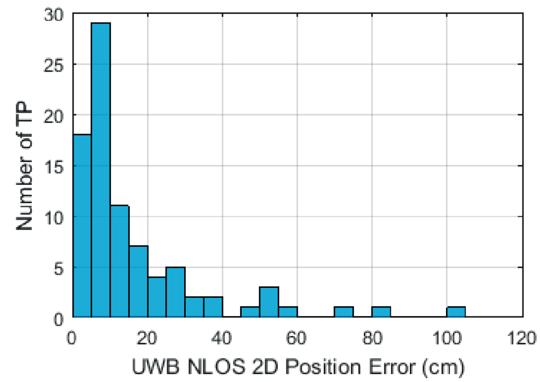

Fig. 5: Distribution of NLoS positioning errors

B. System Architecture

Firmware and Roles: All devices run the same firmware, which enables the DRTLS (Decawave Real-Time Location System) functionality (see Fig. 4) and assigns behavior based on configuration as either anchor, bridge, or tag. The PANS (Positioning and Networking Stack) firmware handles configuration via Bluetooth or shell and supports scheduling, ranging, localization, topology recognition, collision avoidance, and firmware updates.

UWB Devices: The DWM1001 module performs centimeter-level ranging via UWB, supporting both LoS (Line-of-Sight) and limited NLoS (Non-Line-of-Sight) communication. The TDMA-based superframe ensures deterministic, collision-free data flow (see Section II-C). Tags learn network topology from broadcast messages and select 3-4 anchors to range with. The process is repeated periodically, factoring in motion and environmental conditions.

Power efficiency relies on two modes (Responsive, Low Power) and sleep-capable components. Devices can also detect movement using an accelerometer to adjust localization frequency and conserve energy.

Gateway Architecture and Network Integration: Gateways forward UWB network data to LAN/WAN systems (see Fig. 4), using a DWM1001 in bridge mode over UWB and UART. Data is forwarded via MQTT [15], a lightweight protocol standardized by OASIS, while incoming control messages can be sent from cloud systems over HTTP.

Typically on Raspberry Pi, the stack includes a UART module, DWM Daemon, REST proxy, MQTT broker, and config web server.

The gateway handles both upstream data and downstream control commands, enabling two-way interaction with the UWB network. It also initiates firmware updates over the air via the bridge node.

Location Engine – Positioning Algorithm and Position Calculation: The Location Engine is responsible for converting TWR results into position estimates in 3D space. It aims to deliver accurate, low-latency data suitable for visualization, automation, or IoT applications.

Tags use timestamps from TWR with anchors to calculate time-of-flight. Known anchor positions define the coordinate system. The position is then estimated using a maximum likelihood approach, evaluating multiple combinations of available distance data and filtering out high-error solutions. Cached anchor data accelerates repeated calculations, and a moving average across three cycles stabilizes noisy inputs. Each result is scored (0–100) to reflect estimate quality.

The engine adapts to changing network conditions and tolerates limited NLoS by discarding unreliable anchors or down-weighting their influence. It filters multipath effects by detecting distortions in distance data, improving robustness in complex environments.

Scalability and Network Sizing: The DRTLS system is designed for both small and large-scale deployments, but its performance depends on several factors. Effective anchor placement is crucial: each tag should ideally range with at least three or four anchors simultaneously. Anchors should be evenly spaced–typically 5 to 10 meters apart [16]–depending on environmental constraints such as walls or metal structures, which can affect signal integrity.

The TDMA superframe provides 15 TWR slots per 100 ms cycle, allowing up to 150 tags to be queried per second, assuming each uses a dedicated slot. In higher-density deployments, the system rotates devices through available slots, which can increase update latency. This trade-off should be considered in scenarios such as manufacturing floors or hospital tracking systems.

Slot allocation is adaptive: tags monitor anchor response times and change slots if needed. Network topology and current slot usage are shared via Almanac messages, enabling dynamic rebalancing and self-healing.

Bridge nodes and gateways can also be scaled horizontally. Each bridge may handle 30–50 devices [16], and multiple gateways can feed data into a central MQTT broker for aggregation and processing at scale.

NLoS (Non-Line-of-Sight) Behavior and Environmental Effects: UWB is notably resistant to multipath interference,

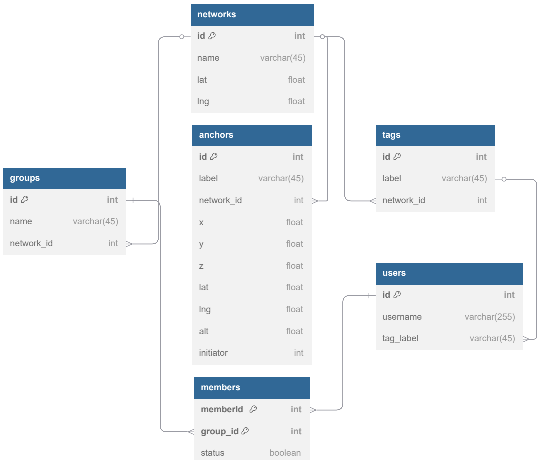

Fig. 6: The data model

but performance may still degrade under NLoS conditions, such as behind walls, metal objects, or dense materials. In such cases, signal reflections may cause the system to overestimate distances by 0.5–1 meters, leading to inaccurate positioning [6] (see Fig. 5).

The Location Engine detects NLoS through indicators like high timestamp variance, inconsistent anchor replies, or measurement errors exceeding thresholds [17]. In response, it may ignore suspect anchors, reduce their weight in position estimates, or switch to more reliable anchor sets to maintain accuracy.

C. Server Components

Data Model: The system uses a relational data model designed for real-time position tracking and user management. Entities include networks, anchors, tags, users, groups, and membership relations (see Fig. 6). Each network links to a geographic origin used to convert relative positions to global coordinates.

Tags store real-time coordinates and a quality score, while each user is assigned a tag for tracking. Groups organize users; a status table tracks membership for filtering and updates.

API: The backend communicates using a bidirectional, event-based model, supporting low-latency updates between server and client. User actions (e.g. device registration, group edits) are sent as events, with real-time system responses.

Key event types include `subscribe`, `registerDevice`, and `createGroup`, along with others for bridge nodes and admin tasks. Server responses include updates for tag positions, anchor changes, user or group creation, and structured error messages. The architecture enables real-time sync and error handling.

D. Clients

Admin Interface: The admin dashboard provides real-time control and visualization of networks, devices, and groups. It is modularly structured into components, shared context definitions, and modal windows for interaction.

State updates are managed via a central service layer that listens to server messages and synchronizes interface data. Two visualization modes are supported: a geographic map view rendered with Leaflet, and a blueprint overlay mode where custom floorplans can be uploaded. The system draws anchors, tags, and connection lines interactively, enabling spatial insight into device relationships.

Mobile Application: The mobile application mirrors the admin structure, providing users with a live overview of tag positions and connection quality. It also supports blueprint overlays with real-time drawing and navigation. Zoom and drag interactions are smoothed with spring-based animations.

Server Communication: Following UWB-based localization (see Section III-B), gateways forward data in JSON format via MQTT. Each tag and anchor is assigned a topic through which updates and control messages are exchanged.

While the default system uses pull-based communication, this introduces scalability issues. To address this, each gateway runs a lightweight MQTT client that pushes incoming data to the central server over a TCP socket (see Section IV). This method enables low-latency transmission and supports optional preprocessing such as filtering or transformation at the edge.

IV. Technologies and Tools Used

The system consists of three main components: a server-side application, client-side interfaces, and an intermediary data service.

The server-side environment, as introduced in Section III-C, relies on the Next.js framework[2] and is built in TypeScript. Although Next.js is traditionally used for server-side rendering aided full-stack web applications, in this project the focus is on modular, file-system-based routing and developer experience tools. The real-time message exchange is facilitated via Socket.IO[3], ensuring low-latency and asynchronous communication.

The data model storage is managed by a MySQL relational database system. Communication between the server and database is handled using the Prisma ORM, providing type-safe, declarative queries and maintainable data models.

For client-side rendering, including the admin interface discussed in Section III-D, React and TypeScript are used, with styling provided by the TailwindCSS framework. This modular component-based structure allows for reuse and consistent design. Map visualizations are handled using Leaflet.

The Bridge component is a Python-based application that collects data from local MQTT brokers and forwards it to the central server. The Linux OS provided by Qorvo lacks the required Python version, so Python 3.6 was compiled from source to ensure compatibility.

The mobile application is built using TypeScript, the React Native framework[4] and Expo. This allows for native iOS and Android applications from a single codebase, using

[2]Official documentation: https://nextjs.org/
[3]Official documentation: https://socket.io/
[4]Official documentation: https://reactnative.dev/

platform-native UI elements. Component updates follow React logic: when *state* or *props* change, components re-render to ensure responsiveness, while background processes run on separate threads.

V. Operation of the Application

This section provides a description of the steps involved in deploying the UWB-based positioning system, as well as the functionality of the administrator and mobile user interfaces.

A. System Deployment

The deployment of the UWB network is a relatively complex process. The first step involves using a flasher program–the tool recommended by Qorvo is Segger, which installs firmware image files onto the chips. Before installation, the appropriate device type must be selected; in this case, the `nRF52832` chip (see Section III-A).

This process must be repeated for each device in the network. One of them must also be connected to a Raspberry Pi, creating the gateway unit. Gateway configuration is handled via a file specifying the PAN ID of the target network.

Once configured, the gateway provides an embedded web interface that displays all devices–both listed and visually positioned on a coordinate system. Roles (anchor, tag, listener) can be assigned directly from this interface.

Alternatively, UWB devices include a "wakeup" button to activate Bluetooth functionality, allowing configuration via the official Qorvo mobile application.

After setting the roles, anchor devices must be physically placed in the space–such as a room or building–where stable connections can be maintained (not overly obstructed or placed too low/high). Automatic position calibration can then be initiated, during which anchors calculate their relative distances and coordinates. The origin is the initiator's position, by default at (`X=0; Y=0`), with the first anchor to the right at (`X=n, Y=0`) defining the OX axis. This constraint must be considered in network design.

After calibration, tags become trackable within the network and begin transmitting position data to the gateway.

B. Admin Interface & Mobile User

The admin interface begins with a connection status indicator, providing feedback on system availability and reacting to error events. A network selector dropdown allows users to choose the active network. Once selected, all associated data–anchors, tags, and groups–are displayed, with toggleable sections.

Groups have a dedicated interface for creation, editing, and deletion. This is useful for managing logical units like warehouse sections or conference rooms.

The map view (see Fig. 7) displays anchors and tags on a Leaflet map, updated in real time. A notable feature is the *Canvas* view (see Fig. 8), which allows uploading custom floor plans and positioning anchors graphically. Reference points can be marked by clicking, and layouts are saved with one click. This combines a visual editor with real-time monitoring.

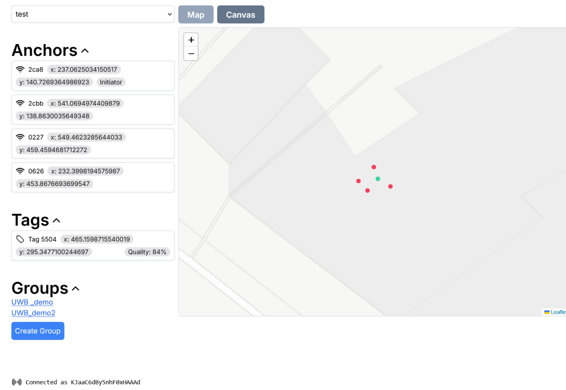

Fig. 7: The map view

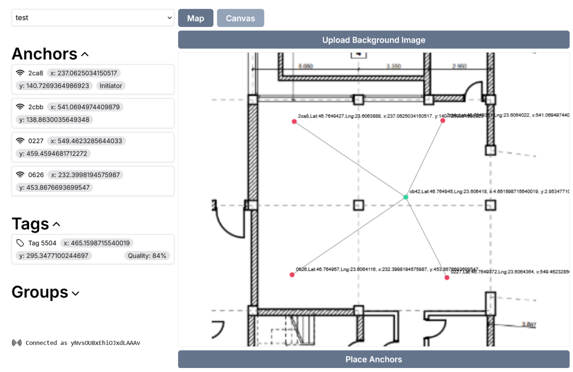

Fig. 8: The Canvas view

The mobile application follows the same principle. Users first register with a chosen name, linked to a specific internal ID, enabling visual tag representation (see Fig. 9). The Canvas view on the mobile app is illustrated in Fig. 10.

After registration, the app displays the current position and system data. If the user belongs to a group, all active devices in that group are shown. Tags are color-coded for clarity. This is particularly useful in scenarios where multiple people or devices are tracked simultaneously, such as events, on-site locations, or healthcare institutions.

The system presents information in three ways:

1) In plain text format.
2) On an interactive map, using Leaflet for geographic positions.
3) On a floor plan-based interface for indoor navigation.

The map and floor plan views allow users to visually track the movement of tags in real time.

VI. Conclusion & Future Work

The current paper successfully proposed an indoor positioning and tracking system using an emerging technology, namely UWB. It presented the advantages above competing technologies, such as BLE and Wi-Fi, and materialized a full system, describing its architecture and usage. Real-world deployment revealed practical and technical challenges, some

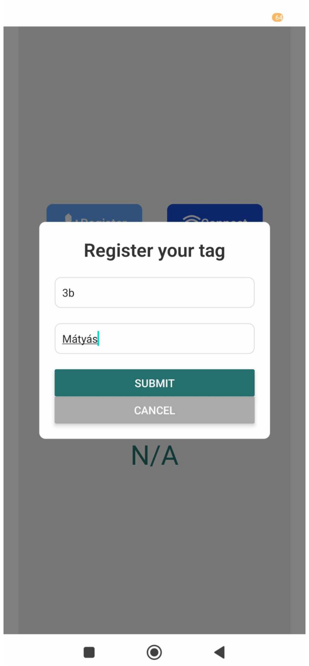

Fig. 9: Mobile app connection menu

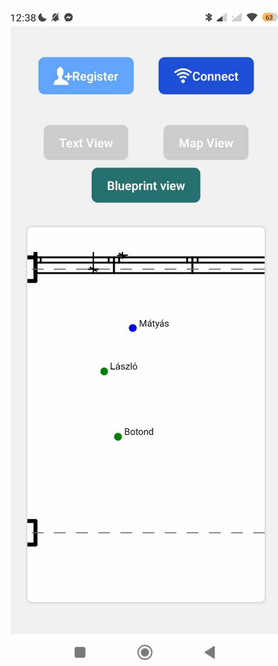

Fig. 10: The Canvas view on the mobile app

absent from official documentation, requiring a more nuanced interpretation of existing claims.

The key challenge has been adapting a theoretically sound design to constrained environments with strict architectural and power-related limits. Despite these compromises, the implementation provides valuable insight into the technology's real-world applicability.

Any discrepancies between observed and documented behavior largely reflect the project's focus: to prototype a working system, not optimize for precision or energy use. Current measurement insights remain preliminary but form a strong foundation for future improvements.

While the current system is functional and promising, several areas remain open for improvement–particularly in long-term scalability, accuracy, and usability.

The DWM1001-based setup has inherent limitations. It is highly sensitive to environmental conditions: without line-of-sight between tag and anchors, accuracy degrades significantly. More critically, the strict anchor placement requirements (see Section V-A) are difficult to adapt to irregular indoor spaces. This restricts deployment flexibility and hinders broader adoption. Future iterations would benefit from more flexible architectures, using newer UWB modules or alternative localization methods.

Power consumption also poses a challenge. The Raspberry Pi and UWB combination has high energy demands, limiting operating time and complicating large-scale deployment. Although energy efficiency was not a primary goal, future work should consider low-power custom hardware capable of sustained battery operation.

The trilateration algorithm also needs refinement. Discrepancies between calculated `(x, y)` coordinates and real-world positions distort visualizations. Improved calibration and algorithmic tuning will be necessary to reduce these errors.

An Extended Kalman Filter (EKF) could offer a promising solution to smooth position updates. By reducing sensor noise and improving motion continuity, the EKF could enhance both reliability and visual stability.

Finally, newer-generation modules such as the DWM3001 present further potential. These support UWB-enabled smartphones (e.g., iPhone 11+[5], newer Samsung and Google Pixel models, enabling mobile integration and new interaction models. The DWM3001 also supports angle-of-arrival (AoA) measurements, adding a new spatial dimension to localization.

References

[1] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.
[2] S. Pace, "The global positioning system: policy issues for an information technology," *Space Policy*, vol. 12, no. 4, pp. 265–275, 1996.
[3] G. T. Schmidt, "GPS/INS Technology Trends for Military Systems," in *Guidance, Navigation, and Control Conference*, 1998.
[4] M. B. Kjærgaard, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, and K. Grønbæk, "Indoor Positioning Using GPS Revisited," in *Pervasive Computing*. Springer Berlin Heidelberg, 2010, pp. 38–56.
[5] Y.-C. Pu and P. You, "Indoor positioning system based on BLE location fingerprinting with classification approach," *Applied Mathematical Modelling*, 2018.
[6] Chen, Yaqin, "Evaluating Off-the-shelf Hardware for Indoor Positioning," 2017, Lund University Publications Student Paper.
[7] S. Shi, S. Sigg, L. Chen, and Y. Ji, "Accurate Location Tracking From CSI-Based Passive Device-Free Probabilistic Fingerprinting," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 5217–5230, 2018.
[8] P. Sadhukhan, S. Gain, K. Dahal, S. Chattopadhyay, N. Garain, and X. Wang, "An efficient clustering with robust outlier mitigation for Wi-Fi fingerprint based indoor positioning," *Appl. Soft Comput.*, vol. 109, p. 107549, 2021.
[9] R. Tesoriero, R. Tebar, J. A. Gallud, M. Lozano, and V. Penichet, "Improving location awareness in indoor spaces using RFID technology," *Expert Syst. Appl.*, vol. 37, pp. 894–898, 2010.
[10] X. Hui, "RFID-Based Positioning Technology," *Computer Knowledge and Technology*, 2013.
[11] C. Tu, J. Zhang, Z. Quan, and Y. Ding, "UWB Indoor Localization Method Based on Neural Network Multi-classification for NLOS Distance Correction," *Sensors and Actuators A: Physical*, 2024.
[12] D. Kocur, *UWB Technology and its Applications*. IntechOpen, 2019.
[13] Murata Manufacturing Co., Ltd., "What Is Ultra-wideband (UWB) Wireless Communication?" 2023. [Online]. Available: https://article.murata.com/en-global/article/what-is-uwb-wireless-communication
[14] G. Tiberi and M. Ghavami, "Ultra-Wideband (UWB) Systems in Biomedical Sensing," *Sensors (Basel, Switzerland)*, vol. 22, 2022.
[15] T. Pulver, *Hands-On Internet of Things with MQTT: Build connected IoT devices with Arduino and MQ Telemetry Transport (MQTT)*. Packt Publishing, 2019.
[16] "DWM1001 System Overview And Performance," 2020. [Online]. Available: https://www.qorvo.com/products/d/da007974
[17] Qorvo, US. Inc., "APS006 Application Note: Channel Effects on Communications Range and Time Stamp Accuracy," Tech. Rep., 2024. [Online]. Available: https://www.qorvo.com/products/d/da008440

[5]UWB support in Apple devices: https://support.apple.com/en-us/109512