

UNIVERSITY OF MORATUWA

Faculty of Engineering

Department of Computer Science & Engineering



CS4203 - Research And Development Project

Dynamic Pixel Localization for Adaptive Crowd Choreography

Project Progress Report

Project Supervisor - Internal

Dr. Kutila Gunasekara

Team Members

Gathsara J.A.S - 210180L

Manchanayake S.M.A.N.A - 210373G

Wickramarachchi N.M. - 210697D

Last Updated On February 19, 2026

Table of Contents

1	Introduction	3
2	Problem Statement	5
2.1	Motivation	5
2.2	Problem	6
2.2.1	Categorization of the Problem	9
3	Research Objectives	10
3.1	General Objective	10
3.2	Specific Objectives	10
4	Literature Review	12
4.1	Analysis of the Environment and Requirements	12
4.2	Localization Measurement Techniques	12
4.3	Existing Wireless Technologies for Localization	13
4.4	Ultra wideband (UWB) Localization	14
4.4.1	Scalability in UWB based systems	15
4.5	Other Localization Systems	18
4.6	UWB Radio Modules	19
4.7	Mapping from an image to location identified-dynamic-pixels	21
4.8	Control Message Transmission	21
5	Methodology	23
5.1	Localization of Individual Pixels	23
5.1.1	Hardware Setup	23
5.1.2	Deployment	24
5.1.3	Ranging Scheme	25
5.1.4	Localization Computation	25
5.1.5	Evaluation / Performance Targets	26
5.2	Pattern Orchestration (Locations → Light Commands)	26
5.2.1	Procedure	27
5.2.2	Evaluation / Performance Targets	28
6	Project Progress	29
6.1	Finalized System Architecture	29
6.2	Hardware Localization Scheme	30
6.2.1	Anchor and Tag Design	30
6.2.2	System Backbone and Gateway Integration	34
6.3	Localization Methods: Double-Sided Two-Way Ranging (DS-TWR)	34
6.3.1	Methodology	34
6.3.2	Analysis of Results and Filtering	36

6.4	Localization Methods: Time Difference of Arrival (TDOA)	41
6.4.1	The Challenge of Time Synchronization: The Need for a Single Time Domain	41
6.4.2	The Ideal (but Impractical) Solution	42
6.4.3	Practical Synchronization Methodology	42
6.4.4	The Master-Slave Sync Model	42
6.4.5	Method 1 Linear Regression (Statistical Smoothing)	42
6.4.6	Method 2: Extrapolation (Instantaneous Correction)	43
6.4.7	Experimental Setup and Results for TDOA	43
6.5	Core Application Logic	46
6.5.1	Human Generator (Simulation)	46
6.5.2	Localization Estimator	46
6.5.3	Pattern Generator	47
6.5.4	Synchronizer	48
6.5.5	Visualizer	49
6.5.6	Database & Pattern Player	49
6.6	RGB LED Data Reception and Control	50
6.6.1	Wireless Color Data Transmission Architecture	51
6.6.2	Receiver Circuit	52
6.6.3	Controlling LEDs	55
7	Research Timeline	57
7.1	Project Milestones	57
7.2	Gantt Chart	58

1. Introduction

In the entertainment industry, enhancing audience involvement and creating interactive experiences have become crucial elements of large-scale events such as concerts, festivals, and sports shows. Traditionally, the use of smartphone flashers symbolized audience participation. However, recent advancements have introduced LED wristbands as an alternative, offering more synchronized and visually engaging experiences. These LED wristbands are currently limited to emitting color patterns block-wise or basic lighting sequences, which restricts their potential for delivering more immersive visual effects.

Existing systems rely on RF (Radio Frequency) or IR (Infrared) technologies to control lighting effects. While these solutions have enabled partial interactivity such as illuminating specific venue sections in chosen colors or projecting beam-based basic light patterns they fall short when it comes to **generating complex visuals** or **maintaining consistency** as audiences move around the venue. The core challenge lies in the inability to dynamically track and localize each individual wristband, which prevents the system from producing coherent large-scale visual patterns in real-time.

To overcome these limitations, our research project, **Dynamic Pixel Localization for Adaptive Crowd Choreography**, proposes a novel solution that transforms every LED wristband into a dynamic pixel within a moving, human display (a display made of humans wearing lighting devices). The system aims to:

1. Accurately localize the position of each individual wristband in real-time.
2. Develop algorithms capable of visualizing the crowd as a point cloud and generating control messages for consistent patterns or visuals.
3. Design a communication framework that transmits the control messages to each wristband, ensuring synchronized lighting effects across the entire venue.

As illustrated in figure 1.1 by integrating **Real-Time Localization**, **Pattern Orchestration**, and **Synchronized Control**, this project will unlock the potential of LED wristbands as dynamic pixels, enabling fully immersive crowd choreography experiences. This approach not only enhances audience participation but also opens new possibilities for creative performances, where the crowd itself becomes an active part of the visual storytelling.

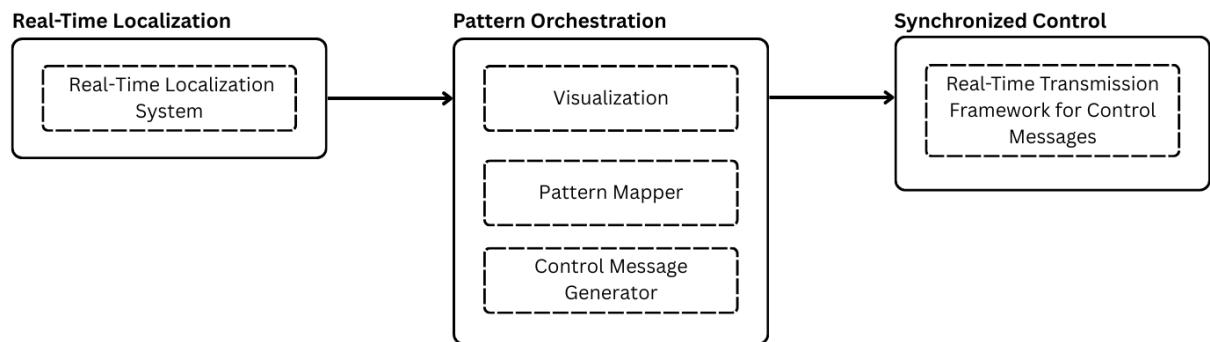


Figure 1.1: High-level workflow of the proposed solution for real-time wristband localization, point cloud visualization, control message generation and synchronized lighting control.

2. Problem Statement

2.1 Motivation

Entertainment events such as global music concerts, sports opening ceremonies, and large-scale festivals attract audiences in the tens of thousands, and in some cases, exceed one hundred thousand participants. These gatherings are not only significant for their cultural and economic impact but also for the immersive experiences they aim to deliver. To enhance audience participation and create collective visual spectacles, event organizers have explored a variety of interactive technologies. Among them, LED wristbands have emerged as a widely adopted solution, transforming spectators into active contributors to the visual narrative of the event. As visualized in Figure 2.1, from aerial perspectives or in post event media, the synchronized lighting of thousands of wristbands has turned crowds into part of the storytelling medium, making the audience itself a part of the performance.

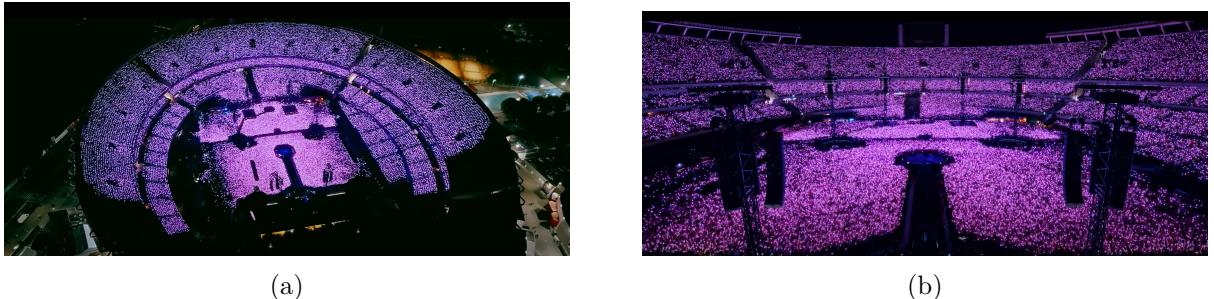


Figure 2.1: Coldplay - A Sky Full Of Stars Concert (Live at River Plate) [1] crowd is wearing PixMob [2] light-bands

Several companies [2, 3] already produce LED wristbands for such applications, essentially treating individuals in the crowd as pixels in a vast, human centered display. This concept highlights the potential of transforming audiences into a living screen, capable of rendering visuals, animations, and even sponsor driven media. However, current implementations remain limited to simple lighting effects and basic color patterns. The inability to dynamically address individual devices and account for the ever changing positions of participants restricts the possibility of displaying coherent and complex graphics across the entire audience.

A critical barrier lies in the dynamic and unpredictable behavior of the devices. Unlike fixed displays, wristbands cannot be assumed to remain in predetermined positions or regions throughout an event. As a result, localization of each individual device becomes an essential yet highly challenging requirement. This challenge aligns with the broader field of indoor localization, which is an active area of research not only for entertainment applications but also in domains such as smart manufacturing, healthcare, logistics, retail, and large scale IoT systems. Although numerous solutions exist, they continue to face trade offs in terms of accuracy, power consumption, interference resilience, and cost effectiveness.

Addressing the problem of large-scale, real-time localization of thousands of moving devices in highly dynamic environments has the potential to extend far beyond entertainment. It can contribute to the advancement of indoor localization technologies for critical industries, while simultaneously enabling next generation immersive experiences in live events. Beyond localization, additional challenges arise in developing algorithms and software capable of visualizing device positions as a real-time point cloud, feeding custom patterns or graphical content, and retransmitting synchronized control signals to individual devices. Solving these problems would not only improve the performance and scalability of existing solutions but also elevate crowd based visual choreography into a truly dynamic, interactive medium.

2.2 Problem

The concept of crowd choreography using LED wristbands has already been applied in concerts and large entertainment events. However, current implementations offer only simple color/pattern options and coarse control. As shown in Figure 2.2, basic LED bands often emit a single fixed color or a simple flashing pattern restricting the range of visual expression.

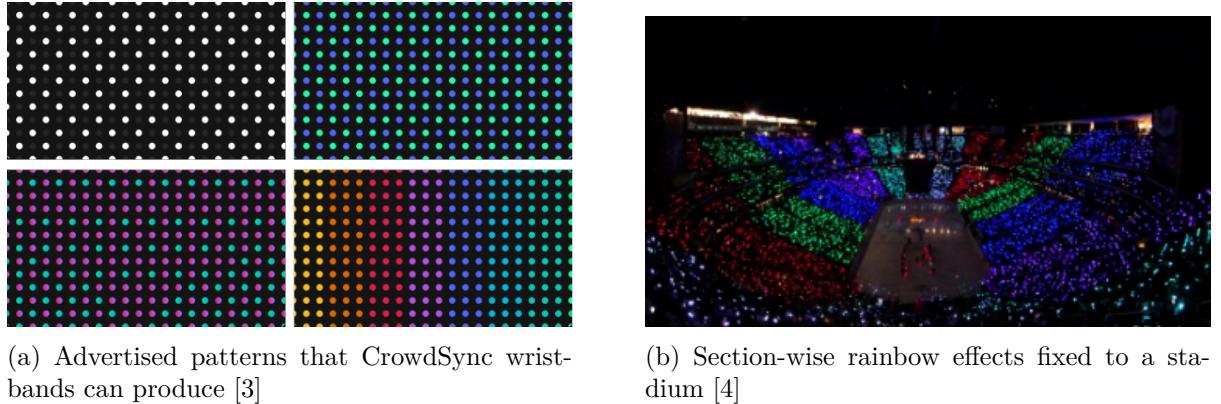


Figure 2.2

Even advanced systems must have pre-program effects such that IR-controlled bands are driven by pan-tilt transmitters (moving IR heads) sweeping the audience as depicted in Figure 2.3, activating pre-set color sequences rather than allowing spontaneous design [5]. Likewise, RF-based bands, as shown in Figure 2.4 can only light up broad sections at a time, they require heavy pre-planning “to ensure each section of the audience is individually addressable” [5].

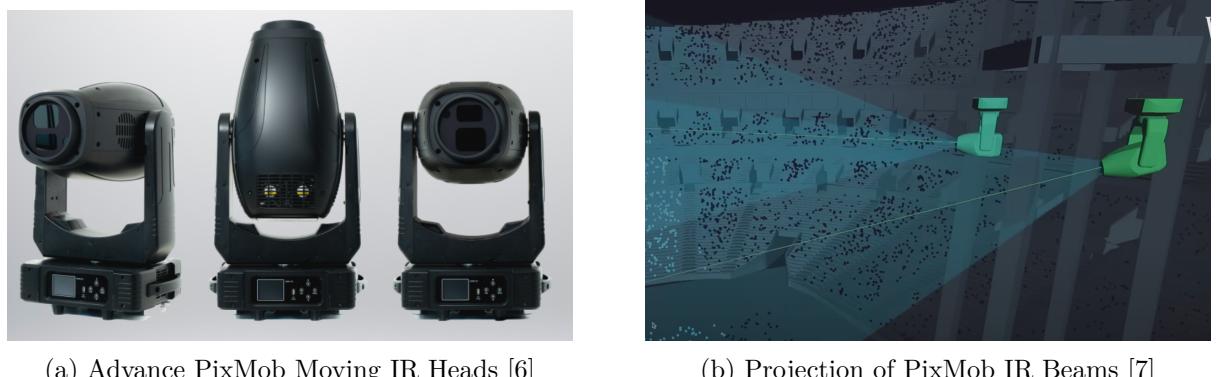
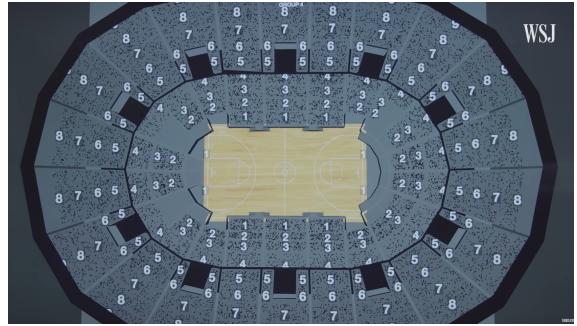


Figure 2.3



(a) RF Based Wristband System with transmitter on the left[5, 7]



(b) Venue is divided into sections where RF Signals are transmitted into each section with pre-positioned wristbands [7]

Figure 2.4

In practice, this means concert lighting operators can only choreograph zone-level effects, not fine-grained patterns (e.g. text or small symbols) across the crowd. These fundamental limitations primarily arise from two factors: **the absence of knowledge regarding the precise physical location of each device** and the **lack of individual device addressing capability**. Together, these shortcomings impose several critical constraints on the system's creative potential. Which can be:

- 1. Restricted Visual Patterns** Current LED wristband systems are restricted to only basic lighting effects such as single-color or multi-color blinks, simple fades, pulses, and section-level chases, typically controlled using RF broadcast methods. Because devices are addressed in groups rather than individually, the audience can only be divided into large color blocks, which prevents fine detail and results in very low spatial resolution. The available color depth is also limited to a few discrete RGB steps and coarse intensity levels, making smooth gradients or nuanced transitions impossible. As a result, complex visuals such as logos, texts, marketing media, or detailed imagery with varied RGB values and brightness levels cannot be rendered reliably. Even attempts at fluid animations like waves or ripples often appear disjointed due to timing jitter and uneven device responses. Figure 2.5 shows why current wristband systems cannot produce complex patterns.

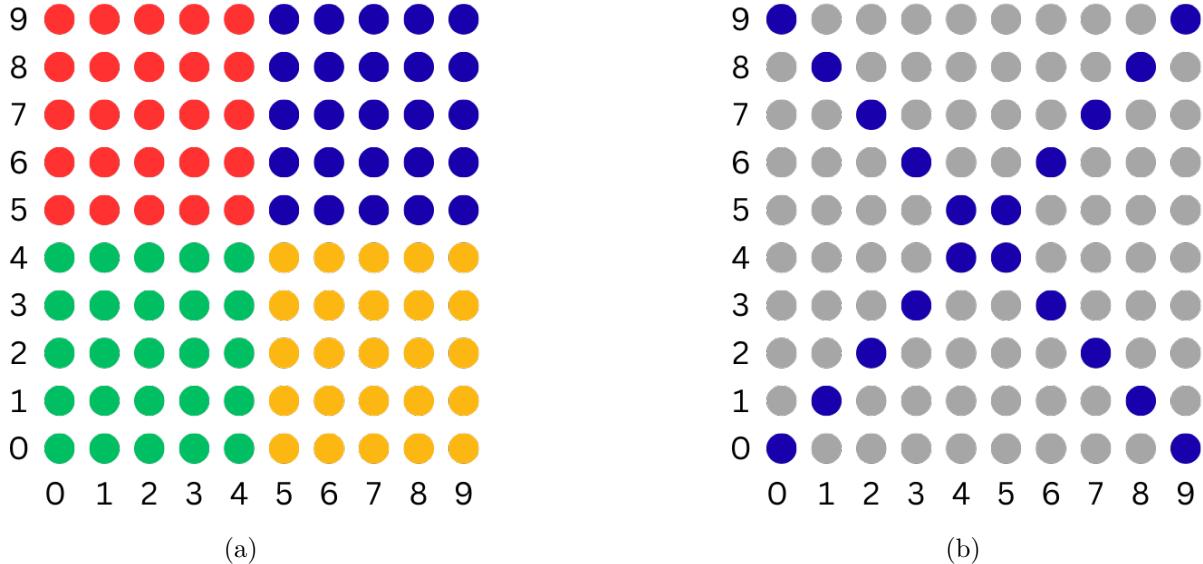


Figure 2.5: Only Zone or Section based lighting is possible, no Individually addressed Colors Patterns are possible

2. **Lack of Adaptivity** A major limitation of current LED wristband systems is their lack of adaptivity to changes in crowd positioning. As shown in Figure 2.6, once a pattern is broadcast, the wristbands maintain their assigned colors or effects regardless of whether people move, swap seats, or cluster differently. This means that multi-color patterns or image-like effects quickly lose coherence when participants shift around, often distorting or breaking the intended visual entirely. In some cases, large sections of the audience can break the intended effect., creating mismatched patches of color or gaps in the display. Because the system has no awareness of the audience's real-time spatial arrangement, there is no way to visualize or leverage the natural diversity and movement of the crowd. In essence, since there is no dynamic adaptation, designers are limited to fixed patterns instead of visuals that respond to the crowd.

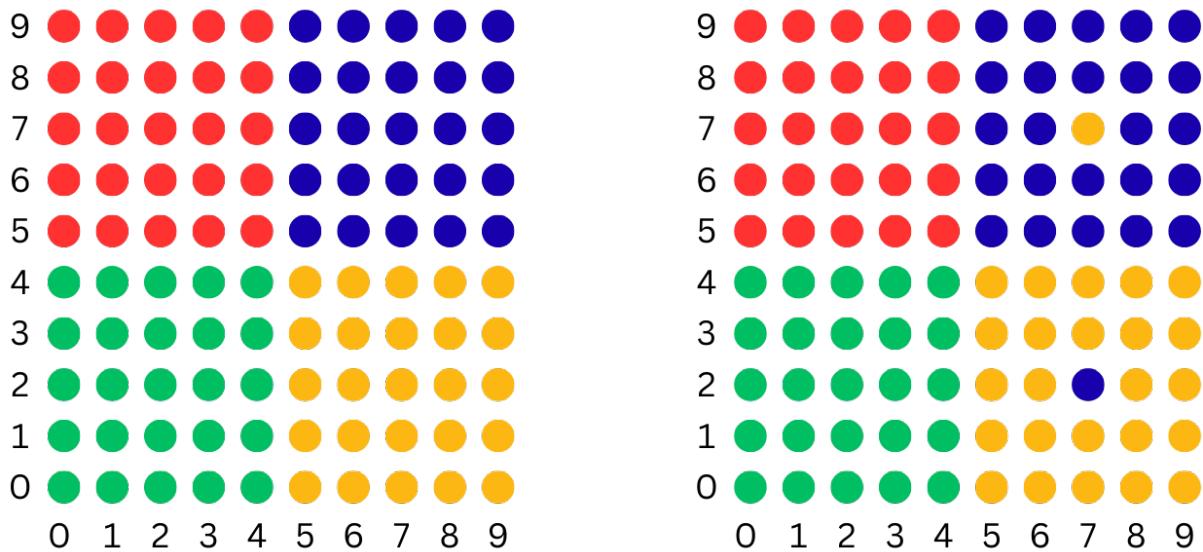


Figure 2.6: In RF section-based systems, if pixel (2,7) is swapped with pixel (7,7), the corresponding color assignment remains unchanged, thereby disrupting the overall pattern consistency. In contrast, IR-based methods inherently avoid this issue.

2.2.1 Categorization of the Problem

To address these shortcomings, the problem can be broken into three interconnected challenges:

1. Localization of Individual Pixels

Without knowing the exact position of each device, it is impossible to map visual effects onto the crowd with spatial coherence. Current RF/IR systems cannot provide this granularity. Each wristband must be localized in real-time with high accuracy and low latency.

Challenge: A key research problem is to achieve **scalable, real-time localization of thousands of moving participants** with sub-meter accuracy in a noisy and dynamic environment.

2. Pattern Mapping Algorithm

Once audience positions are known, the system must map this dynamic and irregular distribution into consistent and adaptive visual patterns. This requires real-time algorithms that can take desired shapes, animations, or interactive effects and translate them onto the point cloud of audience locations.

Challenge: The challenge is to design **algorithms that generate stable visuals that adapt to crowd movement and uneven densities**.

3. Low-Latency Transmission of Lighting Commands

After generating the commands (Device ID, target color, brightness level, and any other effect parameters), they must be reliably transmitted back to each wristband with very low latency and minimal jitter. The system must synchronize thousands of devices so that patterns appear coherent and fluid across the entire venue.

Challenge: This raises the question of building a **scalable communication framework** that ensures real-time, synchronized actuation of thousands of devices even under interference, packet loss, and network constraints?

3. Research Objectives

3.1 General Objective

To Design and implement a real-time crowd choreography system where each wristband is localized continuously, visual patterns (images/animations) are mapped to each device's current position, and low-latency control frames (device ID, target color, brightness, and effect parameters) are transmitted so the audience becomes an adaptive, spatially-aware display.

3.2 Specific Objectives

1. **To Develop a Real-Time Localization of Large Number of Nodes using Ultra Wide Band(UWB) Technology**

Build a scalable UWB-based localization subsystem (anchors + tags) that provides continuous position updates for thousands of devices

2. **Real-time point-cloud visualizer**

Visualize the live 2D point cloud of localized devices overlaid on an arena map (stadium/venue seating plan) shown in Figure 3.1, Where the map of the arena can be configured using exact dimensions.

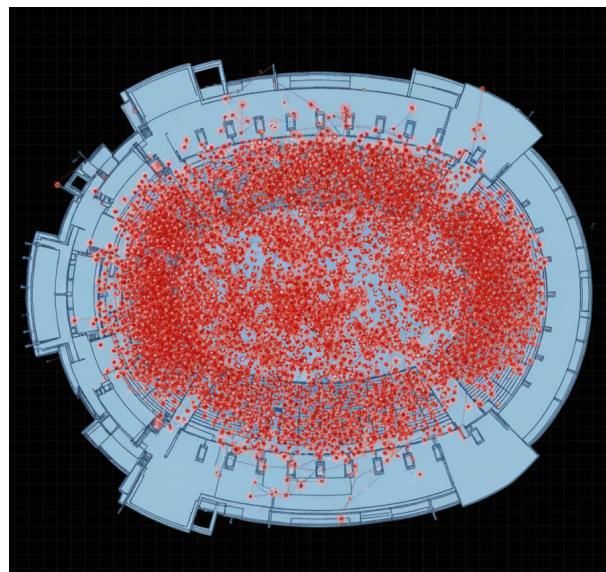


Figure 3.1

3. **To Develop a Pattern Mapping Algorithm and Control Message generator**

Develop algorithms that map an input pattern (image, video frames, or parametric animation) onto the irregular, dynamic point cloud of devices and generate relevant control signals (per-device control frames)

4. Literature Review

4.1 Analysis of the Environment and Requirements

Indoor localization for massive IoT deployments must handle large, semi-indoor spaces with high radio interference and multi-path reflections. In such environments, interference and multi-path effects are major problems [8] [9]. Multi-path effects occur when signals reflect off surfaces like stage structures, speaker walls, and temporary installations, creating multiple signal paths that cause phase distortions and inaccurate ranging measurements, particularly in phase-based systems like Phase of Arrival (PoA) which become "useless in many cases" under these conditions [10]. Furthermore, musical equipment such as wireless microphones, in-ear monitoring systems, and frequency-hopping communication devices operate in crowded RF spectrum (2.4 GHz and 5 GHz bands), causing interference that disrupts RF-based ranging precision. This white paper [11] from the FiRa Consortium presents a study on the interference impact of IMT (International Mobile Telecommunications, Cellular Base Stations) systems on Ultra Wide band (UWB) technology. Using Monte Carlo simulations, the study concludes that high-power IMT base stations both outdoor macro cells and indoor small cells cause severe and unacceptable interference to UWB devices. This interference significantly degrades UWB performance, reducing communication range and coverage area, and causing extreme delays.

4.2 Localization Measurement Techniques

Numerous basic methods exist to estimate the position. The main approaches are Received Signal Strength Indicator (RSSI), Time of Flight (ToF), Return Time of Flight (RToF), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), Phase of Arrival (PoA) [12].

Received Signal Strength Indicator (RSSI) is the simplest and most cost-effective method. Since it relies on existing hardware, but its accuracy is often poor due to multi-path interference and signal fluctuations [12]. Angle of Arrival (AoA) techniques achieve high precision by estimating the direction of the incoming signal with antenna arrays. However, they require complex directional antennas and their performance degrades over longer distances [12]. Time of Flight (ToF) approaches the estimate distance by measuring the signal travel time, offering high accuracy in line of sight (LOS) conditions with precise time synchronization, but this requires additional hardware and tightly synchronized clocks [12]. Time Difference of Arrival (TDoA) mitigates some of these limitations by depending only on synchronized anchor nodes instead of device clocks, but it requires wide bandwidth and strict time synchronization between anchors [12]. Return ToF (RToF), a two-way ranging technique, also achieves high accuracy by measuring round-trip delays. However, it remains vulnerable to clock synchronization issues and processing delays in short-range links [12]. Finally, Phase of Arrival (PoA) methods determine position by analyzing the phase shift of received signals, providing fine-grained accuracy under LOS conditions, but are highly susceptible to multi-path effects and require coherent receivers with precise calibration [12].

Based on these observations, the above localization techniques can be categorized by their accuracy, scalability, power efficiency and trade-offs between them. Accuracy is generally strong in the TDoA, ToF, and AoA methods, while scalability is best achieved with RSSI, secondly with TDoA and ToF. Power efficiency can be maintained across TDoA, ToF, and AoA with proper design. Phase of Arrival (PoA), though theoretically superior, is rarely applied as a standalone method and is instead combined with others to balance limitations. In conclusion, TDoA and ToF emerge as the most suitable solutions for the intended use case. Table 4.1 provides a summary of the features of different localization schemes.

Feature	RSSI	ToF	AoA	TDoA
Accuracy	Moderate	High	High	High
Scalability	Higher	High	Moderate	High
Power Efficiency	Moderate	High	High	High

Table 4.1: Summary of comparison between Localization schemes in terms of Accuracy, Scalability, Power Efficiency

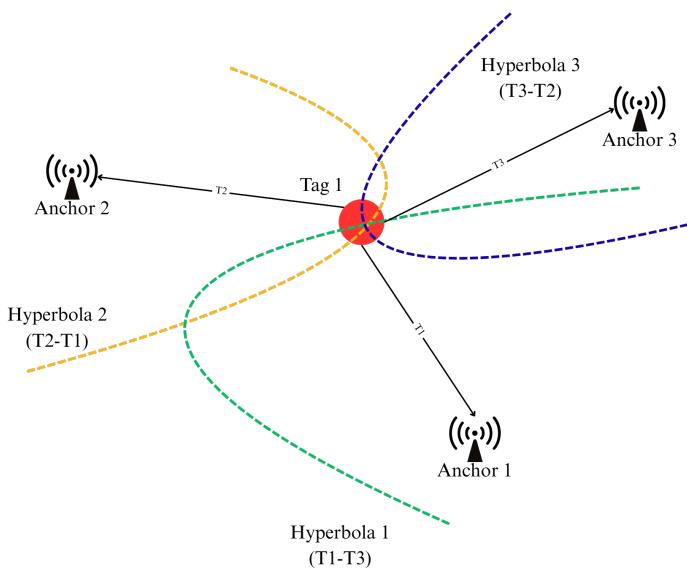


Figure 4.1: The system of hyperbola equations can be solved using nonlinear least-squares optimization techniques such as Levenberg Marquardt algorithm [12, 13]

4.3 Existing Wireless Technologies for Localization

The measurement techniques mentioned above can be applied to various radio or light technologies. Common media include Wi-Fi, Bluetooth (BLE), Ultra-Wideband (UWB), RFID, and Visible Light. Each has its own range, power, and accuracy characteristics [12] .

A specific version of Wi-Fi developed for IoT networks, known as Wi-Fi HaLow (based on IEEE 802.11ah specifications), has been applied in real-time localization systems due to its notable characteristics, such as a long-range coverage of up to 1000 m [14]. Furthermore, a single access point can connect up to 8192 devices [14], making it highly scalable. Wi-Fi HaLow generally consumes less power, except when operating at its maximum range [15],

16]. However, a major limitation of Wi-Fi HaLow (and Wi-Fi technologies in general) is their susceptibility to interference [12], particularly from multi-path fading and the use of the crowded ISM band. Additionally, Wi-Fi protocols are primarily optimized for data transmission rather than localization [12].

Bluetooth Low Energy (BLE) is another common technology used in indoor localization. It is popular because it is low-power and low-cost [17]. However, its accuracy (about 1–3 m) is quite limited, especially when using RSSI-based methods [17, 8], which are much less precise than Angle of Arrival (AoA) techniques available only in BLE 5.1 and newer versions. In addition, BLE often suffers from interference since it works in the crowded 2.4 GHz ISM band, which reduces its reliability in environments with many devices.

To overcome these issues, some studies have tried combining BLE with other methods. For example, creating a mesh network with DECT-2020 NR (based on 5G standards) has been tested [18], but it is not cost-effective compared to other options. Other BLE mesh network approaches also failed because of heavy interference [19]. On the other hand, techniques such as Extended Advertisements and channel diversity have shown some improvements in localization accuracy when using BLE with RSSI-based methods [17].

Ultra-Wideband (UWB) is another technology that is widely studied and has become very popular for localization. UWB can achieve high accuracy (around 10–30 cm) and is resistant to problems like multi-path fading and even non-line-of-sight conditions [20]. Techniques such as Time of Flight (ToF) and Time Difference of Arrival (TDoA) are often used with UWB to measure distances. Since it works with a large bandwidth and high frequency, UWB also coexists well with other wireless systems like Wi-Fi and Bluetooth [12].

With the release of the IEEE 802.15.4z standard in 2020, UWB has further improved its power efficiency [21]. The technology has recently been adopted in many consumer products (such as Apple's U1 chip, iPhones, and AirTags) as well as in industrial IoT devices [21, 9]. Currently, UWB modules are still more expensive compared to BLE or Wi-Fi, but studies show that prices are gradually decreasing over the years [9].

Radio Frequency Identification (RFID) uses either passive or active tags together with readers. RFID-based localization is usually coarse, often limited to identifying a zone rather than an exact position, but the tags are very low-cost. Passive RFID tags do not require batteries, making them very low power, but their range is short (about 1–2 m) and their accuracy is poor. Active RFID tags use a local power source, which extends the range up to around 100 m, but they still cannot achieve sub-meter accuracy [12]. Because of these limitations, RFID is mostly used in static indoor localization applications such as inventory management.

According to Zafari et al. (2019) [12], Wi-Fi and Bluetooth offer high availability but cannot provide sub-meter accuracy, while UWB is immune to interference and achieves much higher accuracy.

4.4 Ultra wideband (UWB) Localization

UWB radio is characterized by its very large signal bandwidth ($\leq 500MHz$) and extremely short pulses (<1 ns) [12, 9]. These features, along with its high temporal resolution [21], allow UWB to provide precise distance estimates using methods such as ToF and TDoA. The ultra-short pulses also make UWB naturally resistant to multipath fading, while the wide frequency range reduces interference with other wireless systems and allows signals to penetrate materials [9]. Because of this combination of high accuracy and robustness, UWB is especially well suited for real-time locating systems (RTLS) in complex indoor environments.

4.4.1 Scalability in UWB based systems

Even though UWB can achieve centimeter-level accuracy, its low-power transmission (designed to reduce interference with narrowband systems) results in range limitations and scalability challenges [21, 9]. Most research so far has focused on improving accuracy rather than scalability.

Ridolfi et al. [9] provided a detailed analysis of UWB scalability and reviewed several approaches aimed at addressing this issue. For example, the non-commercial system OpenRTLS [22] demonstrated the ability to localize up to 7500 devices within a 20 m radius, while the commercial solution Sewio supports around 1000 devices within a 30m radius. Both of these systems rely on TDoA as their main localization method. In contrast, a simulated RSSI-based distributed system was only able to support about 200 devices.

Ridolfi et al. [9] identified three key factors that influence the scalability of UWB systems:

1. Localization Schemes
2. MAC Protocol
3. Physical Layer (PHY) Settings

Their study analyzed the impact of these factors under different configurations, such as:

- **Localization Scheme:** Time Difference of Arrival (TDoA) vs. Two-Way Ranging (TWR).
- **MAC Protocol:** Scheduled access (TDMA) vs. Random Access (ALOHA).
- **PHY Layer Settings:** Data rate, preamble length, and pulse repetition frequency (PRF), all of which directly affect packet duration.

In addition, the authors proposed a generalized three-step methodology (including mathematical equations) to calculate the maximum supported user density for any given UWB system configuration.

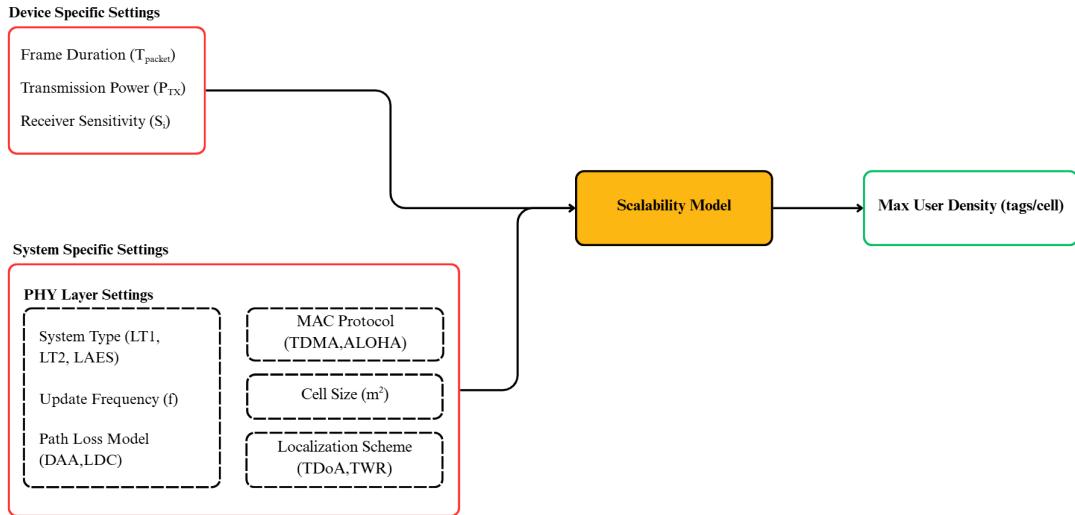


Figure 4.2: Scalability Model proposed by [23] and its Input Parameters and Output Parameters

The scalability model was validated using the specifications of the commercially available Decawave DW1000 chip and simulated in MATLAB. Based on this validation, a system using TDoA, a TDMA-based MAC layer, and short packet configurations was shown to theoretically support localization of up to 6171 devices per second according to the mathematical model. The

MATLAB simulation achieved a similar result, successfully localizing 5546 devices per second. However, the detailed implementation of the MATLAB simulation was not provided.

Another localization system, SnapLoc [24], introduced a completely different approach to address the scalability problem, enabling the localization of an "unlimited" number of devices. SnapLoc uses a method called Concurrent Ranging, where tags localize themselves by analyzing the Channel Impulse Response (CIR) of a single combined waveform transmitted simultaneously by all anchors (without delay).

Also they addresses issues associated with Concurrent Ranging which affects the scalability such as:

1. Difficulty in distinguishing between peaks in the CIR that correspond to specific anchors due to strong multi-path components.
2. High packet loss when multiple anchors are positioned at similar distances from a tag.
3. Limited timestamp resolution in current UWB radios, such as the Decawave DW1000 chip.

By addressing the above challenges, SnapLoc was able to derive the TDoA for each tag in quasi-simultaneous time. Instead of having each tag communicate sequentially with the anchors, a single reference anchor broadcasts an INIT message. All other anchors then respond quasi-simultaneously (within the same 500 μ s window) with empty RESP messages.

The tag's position is not determined by decoding the payloads of these packets, but rather by analyzing the Channel Impulse Response (CIR) generated by the combined signal. To make this possible, each anchor's response is intentionally delayed by a unique nanosecond-scale offset (δ_i), which allows the tag to distinguish individual anchors within the composite CIR signal and compute the TDoA values between them.

The SnapLoc system was thoroughly evaluated using real hardware setups. Tests were conducted in both an office space (4 anchors, 1 reference anchor, 28 tags) and a large laboratory space (4 anchors, 1 reference anchor, 14 tags). The system achieved decimeter-level positioning accuracy, with a 90% error of 33.4cm and a median error of 18.4cm. However, the study did not provide details about the exact distances between anchors and tags.

When compared with the scalability approach discussed in [9], SnapLoc has some limitations. In particular, localization occurs passively and anonymously within the tags themselves, which increases processing requirements at the tag level and complicates integration with centralized systems [24].

Another study, called MuLoc [25], introduced a system that can localize an unlimited number of UWB tags with millimeter-level accuracy. According to the authors, MuLoc is the world's first UWB localization system to achieve such fine precision while supporting an unlimited number of tags. Similar to SnapLoc, MuLoc relies on passive Localization (tags only listen) to enable infinite tag concurrency. MuLoc addresses two major challenges in UWB localization:

1. Obtaining accurate phase estimates from unsynchronized UWB devices using a novel method called **Anchor Overhearing (AO)**.
2. Achieving fine-grained localization by combining Time of Flight (ToF) and Phase of Arrival (PoA) with a frequency-hopping mechanism in the DW1000 chip.

This system also validates the observation in [12] that PoA is superior for fine-grained localization but is rarely used as a standalone method. In MuLoc's "anchor overhearing" scheme, anchors sequentially transmit and then listen to each other. Both tags and anchors measure

the time and phase of each reception. By comparing differences between tag and anchor measurements, MuLoc cancels out synchronization and clock offset errors.

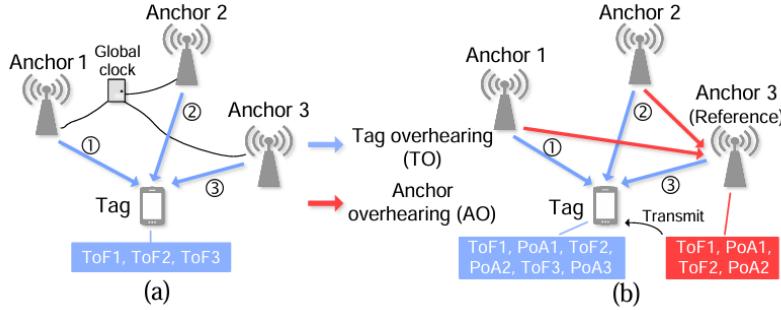


Figure 4.3: Comparison of localization schemes between traditional TDOA and MuLoc. (a) Traditional TDOA scheme, (b) MuLoc scheme [25].

The result is a hybrid ranging method that leverages UWB phase information (capable of resolving sub-centimeter distances) without requiring tightly synchronized anchor clocks. In experiments, MuLoc achieved a median error of 0.47cm, reducing error by 91% compared to standard TDoA.

The study in [25] provides a comprehensive guideline with equations and examples on how to recover the UWB signal, prepare it for localization, and perform fine-grained tag localization. The authors also released source code to allow re-implementation of their experiments [26].

In the first test environment, designed to evaluate overall accuracy, 4 anchors and 1 tag were used. Similar to SnapLoc, MuLoc also requires a reference node. However, instead of using a dedicated reference anchor, MuLoc employs a reference switching mechanism, where one anchor at a time is designated as the reference. The tag was mounted on a sliding track to simulate dynamic movement. To demonstrate the contribution of PoA, the authors also developed a simplified version called MuLoc-, which used only ToF estimation.

The second test environment evaluated scalability, using 25 tags. A third set of experiments was conducted to assess multi-path resilience, with tests performed in three scenarios: a hall with minimal multipath, a meeting room with moderate multipath, and a narrow corridor with strong multi-path. The results showed that while the hall and meeting room maintained millimeter-level accuracy, the corridor scenario degraded to 22.87cm accuracy. Additional experiments were carried out under no-line-of-sight (NLOS) conditions, long-term stability tests, as well as with predefined and arbitrary trajectories, all of which produced promising results.

Both SnapLoc [24] and MuLoc [25] demonstrate that passive TDoA (anchors transmit, tags listen) can, in principle, support an unlimited number of tags. However, this approach places significant processing requirements on the tags themselves, and it is not always clear how the tags' position data can be efficiently centralized. In the MuLoc implementation, for example, the test node included a DWM1000 module connected to an STM32, which was further connected to an ESP32 that transmitted the location data back to an access point via Wi-Fi. This highlights the additional overhead associated with tag-based localization.

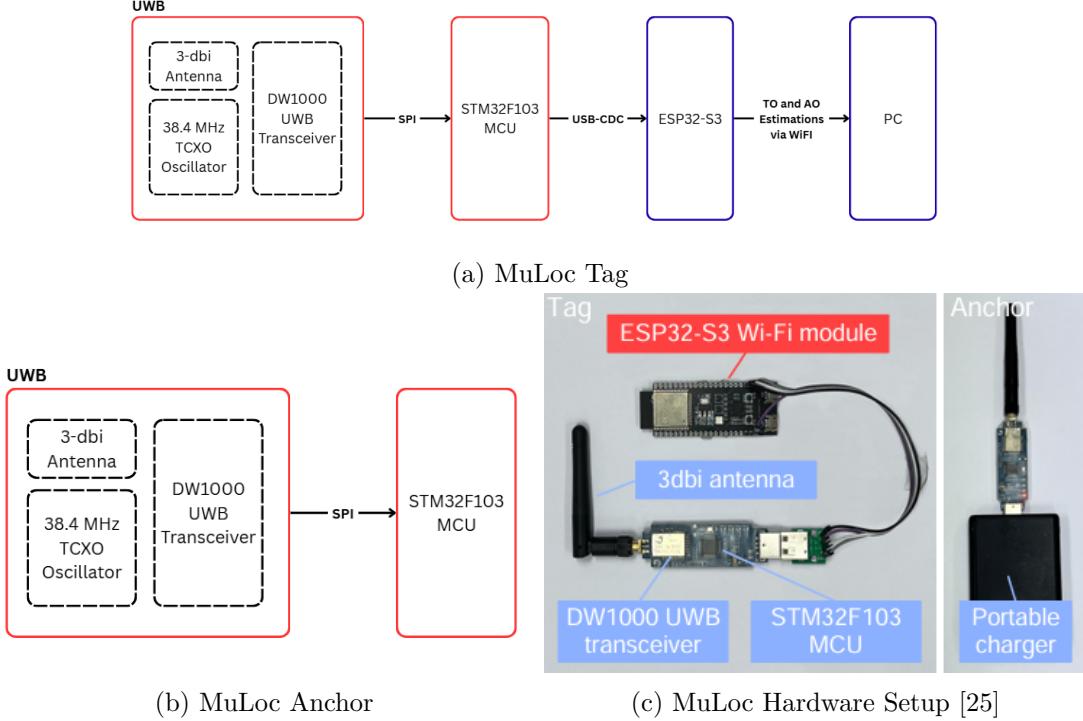


Figure 4.4: MuLoc’s Setup

In one study [27], an Ultra-Wideband (UWB) positioning system was introduced using six identical custom-designed boards, each built using an ESP32 micro-controller and a Quorvo DWM3000 module. The setup comprised one designated tag and five anchors, with localization achieved via Two-Way-Ranging (TWR) measurements. To enhance precision, the tag employed an Extended Kalman Filter (EKF) to fuse ranging results and estimate its position relative to the anchors’ fixed locations. This architecture enabled the system to achieve positioning accuracy of up to 10 cm. The authors highlight that the integration of local processing on the tag board makes the solution suitable for real-time indoor positioning and tracking applications. This work is evident that the ESP32 can be integrated to DWM3000 module for processing of location since all the previous studies use STM32 Micro Controllers [27]. Also it mentioned that using TWR affects the performance of the system with increasing the tag count linearly and suggests using TDOA method.

4.5 Other Localization Systems

[28] present a high speed, low latency localization system using a Dynamic Vision Sensor (DVS) and Active LED Markers (ALMs). While this approach achieves remarkable update rates (1 kHz) and is immune to motion blur, its fundamental operational requirements make it unsuitable for large-scale venue applications. The system is entirely dependent on precise optical line-of-sight between the specialized DVS camera and the blinking LED markers attached to each target. This characteristic renders it highly error-prone in cluttered or dynamic environments where sight-lines are obstructed. Furthermore, scaling this system to cover a large venue would be an infrastructure-heavy network of multiple DVS cameras to maintain coverage, significantly increasing complexity and cost.

4.6 UWB Radio Modules

In recent years, several companies have introduced ready-to-use UWB chips and modules that make it easier and cheaper to build Real-Time Localization Systems. At the forefront is Qorvo (formerly Decawave) [29], whose DW1000, DW3000 transceiver and its integrated modules Ex: DWM1000 (DW1000 IC, an antenna, and power management and clock components), DWM1001 (DW1000 IC, a Nordic Semiconductor nRF52832 system-on-chip (SoC), a 3-axis accelerometer, and integrated antennas) DWM1001-DEV(which includes the DWM1001 module, battery connector and charging circuit, LEDs, buttons, Raspberry Pi-compatible connectivity, and USB connector) [30] have become staple components in RTLS applications, delivering centimeter-level accuracy and support for TDoF and TDoA ranging schemes while adhering to IEEE 802.15.4 standards. NXP's Trimension family (SR040, SR150, NCJ29D5) [31] is another widely used UWB RF module that offers highly integrated UWB solutions with embedded micro-controllers and BLE support tailored for low-power wearable or IoT applications. At the consumer device level, Apple's U1/U2 SoCs have popularized UWB positioning in smartphones(Apple iPhones) and accessories(Apple Watches, AirTags, and Airpods) [21]. Complementing this hardware diversity, industry bodies like the FiRa Consortium are driving interoperability standards and certification, ensuring cross-vendor compatibility across UWB deployments [21] [32].

[21] provide a comprehensive review on standards and compliances UWB uses such as the base UWB standard IEEE 802.15.4 and how it evolved over the time to latest IEEE 802.15.4z by improving better security, accuracy, and interoperability while explaining underlying PHY and MAC Layer Changes. And how FiRa consortium helps to maintain EcoSystem and Interoperability among UWB Devices and systems. Also other associated standards like IEEE 802.15.6, IEEE 802.15.8, ETSI UWB standards , ISO 24730 International Standards.

IEEE 802.15.4a - 2007	IEEE 802.15.4 - 2011	IEEE 802.15.4f - 2012	IEEE 802.15.4 - 2015	IEEE 802.15.4z - 2020
<ul style="list-style-type: none"> Amendment that adds 1st standardization of IR-UWB <ul style="list-style-type: none"> 2ns pulse width In unlicensed UWB spectrum Support for precision ranging 0.11, 0.85, 6.81, or 27.24 Mb/s data rates PRF of 4, 16 or 64 MHz BPM+BPSK modulation 	<ul style="list-style-type: none"> The 'a' amendment was incorporated into the main body of the standard. Popular DW1000 chip is based on this standard 	<ul style="list-style-type: none"> Amendment that adds additional LRP UWB PHY. <ul style="list-style-type: none"> Transmits less but stronger pulses 6.28 GHz to 9.18 GHz spectrum No precision ranging support 0.0312, 0.250 or 1 Mb/s data rates PRF of 1 or 2 MHz PPM or OOK modulation 	<ul style="list-style-type: none"> The 'f' amendment was incorporated into the main body of the standard Two UWB PHY: <ul style="list-style-type: none"> 1. HRP (from 802.15.4-2011) 2. LRP (from 802.15.4f amendment) 	<ul style="list-style-type: none"> Amendment that enhances the UWB PHYs HRP <ul style="list-style-type: none"> <u>Security</u>: addition of STS <u>Accuracy</u>: (1) additional preamble and SFD codes, (2) higher PRFs: 128 and 256 MHz and (3) a new modulation: BPSK. LRP <ul style="list-style-type: none"> Additional higher data rates: up to 10 Mb/s, higher PRF: 4 MHz Additional modulations: PBFSK and PBFSK+PPM Ranging support

Figure 4.5: Overview of UWB IEEE standards, their release years, and the changes introduced over time to improve accuracy, security, and interoperability [21].

It also introduces commercially available UWB transceivers and provides a technical comparison between them. Specifically, [21] compares the Qorvo DW1000 and DW3000 transceivers, highlighting how the DW3000 improves over the DW1000 by complying with the IEEE 802.15.4z PHY and MAC specifications.

1. **DWM1000 Module:** Foundation of Modern UWB Localization The DWM1000 module, based on the DW1000 IC (Released in 2014), has been a turning point for UWB-based localization systems [33]. It integrates an antenna, RF circuitry, power management, and clock circuitry into a single module, complying with IEEE 802.15.4-2011 This is the basis

for popular modules like the Makerfabs ESP32 UWB [34]. This has been the module used in [9], SnapLoc [24] and MuLoc [25].

2. **DWM3000 Module:** Evolution for Scalability and Interoperability The DWM3000 module, based on the DW3110 IC (Released in 2021), addresses the limitations of the DWM1000 [35]. It is designed for next-generation UWB applications, emphasizing interoperability, power efficiency, and regulatory compliance.

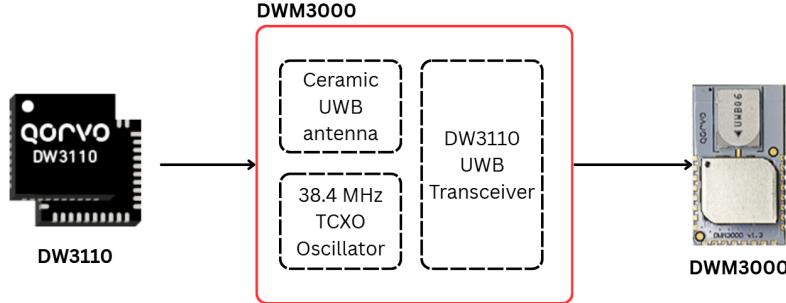


Figure 4.6: High-level Architecture of Qorvo DWM3000 UWB Module

Feature	DW1000	DW3000	Implication for Scalability
IR-UWB Standard	IEEE 802.15.4-2011 UWB compliant	IEEE 802.15.4-2015 and IEEE 802.15.4z BPRF compliant	Improved first path detection, higher reliability, and better security [21]
Power Consumption (TX,RX,TX cycle)	~233 mW [36]	~129 mW [36]	DW3000 reduces power consumption by about 50% [36]
Ecosystem Integration (Interoperability)	IEEE 802.15.4-2011 compliant but not FiRa™ certified	FiRa™ & Apple U1/U2 compliant	Only DW3000 is compliant to the FiRa™ PHY and MAC specifications enabling interoperability with other FiRa™ compliant devices [35]

Table 4.2: Comparison between DW1000 and DW3000 transceivers in terms of Scalability

As shown in Table 4.2, the DW3000 offers lower power consumption and better interoperability compared to the DW1000. The DW3000 represents a significant improvement over the DW1000 by directly addressing several key challenges such as:

- **Power Efficiency:** Lower power consumption supports large-scale deployments with battery-operated tags, extending operational lifetimes from months to years [36, 35].
- **Interoperability:** FiRa compliance and Apple ecosystem integration make the DW3000 more future-proof for both consumer and enterprise applications.

- **Security:** The addition of a Scrambled Timestamp Sequence (STS) field in the PHY layer, as defined in the IEEE 802.15.4z standard, enables more secure communication between UWB devices [21].

Although DW3000 shows better accuracy and precision at lower range (<30cm) in Line of Sight conditions there is no significant difference from DW1000 in long range and no line of sight conditions [36].

4.7 Mapping from an image to location identified-dynamic-pixels

There are trivial ways to send the relevant color data to the pixels for faster moving images/animations to be displayed easily like in normal LED displays. But when the pixels are moving in the 2D space and the current location of each pixel is known, there should be a way to calculate the correct color according to a given input image/animation, a specific pixel should be illuminated in real time.

While related domains such as persistence of vision (POV) displays and volumetric/swept-volume displays demonstrate concepts where moving LEDs or light sources are synchronized with image content [37], the specific problem of dynamic pixels being recognized in arbitrary 2D positions and assigned colors from an input image in real time appears to have very limited coverage in existing academic literature. Most published work focuses on mechanically constrained scenarios (e.g. rotating LED arrays, swept volumes, or projection mapping on rotating surfaces), where pixel positions follow predictable trajectories that simplify synchronization and mapping. In contrast, the idea of free-moving pixels whose instantaneous positions are tracked and mapped to image coordinates on the fly so that each pixel “picks up” its correct color dynamically remains largely unexplored in research papers.

Much of the relevant knowledge comes not from peer-reviewed publications but from hobbyist projects, prototypes, or engineering reports, which, while conceptually related, do not provide a comprehensive academic framework for this problem space. This highlights a gap in the literature and an opportunity for new contributions in real-time dynamic pixel mapping and color assignment.

4.8 Control Message Transmission

For transmitting the Color that a specific pixel needs to be lightened up, we need to send a data packet wirelessly. In principle, several Sub-GHz and 2.4 GHz downlink options can meet the requirement of per-device color updates over ~ 200 m, without acknowledgements. These include:

1. Sub-GHz 2-FSK/GFSK (e.g., TI CC1310 class) [38]
2. LoRa used as a proprietary PHY (non-LoRaWAN) [39]
3. UWB (IEEE 802.15.4a/4z) data frames [40]
4. BLE 5.4 PAwR for scheduled, connectionless broadcasts [41]

In all cases, per-device delivery is achieved by broadcasting a packet that contains many tiny per-device records and lightly repeating it on different channels; each wristband extracts only its own record and applies the color. These methods are therefore technically possible for stadium-scale crowd lighting. The transmission can be received using Sub-GHz Wrist-Worn antennas [42] rather than 2.4GHz. This offers long-range wireless communication with low power consumption for these types of wrist-worn applications.

Considering range margin (\sim 200 m LoS), per-device update rate, battery life, regulatory practicality, and integration effort, the most suitable primary method is Sub-GHz 2-GFSK (TI CC1310-class), using Wake-on-Radio and frequency hopping. This approach:

- Provides comfortable 200 m range at 868/915 MHz with modest antennas.
- Supports higher downlink capacity than LoRa for fast per-device updates (pack many device records per packet).
- Is battery-friendly: wristbands remain in sleep mode and are woken just-in-time by long-preamble packets.
- The CC1310 supports data rates of 50–500 kbps (e.g. ≈ -110 dBm @ 50 kbps; ≈ -97 dBm @ 500 kbps) using 8-FSK modulation [43].

5. Methodology

The system is an end-to-end pipeline that 1) localizes each wristband as a dynamic pixel, 2) maps a given lighting pattern into dynamic pixels and generates control messages , and 3) transmits those control messages with tight time sync and low latency. The control loop runs continuously during a show.

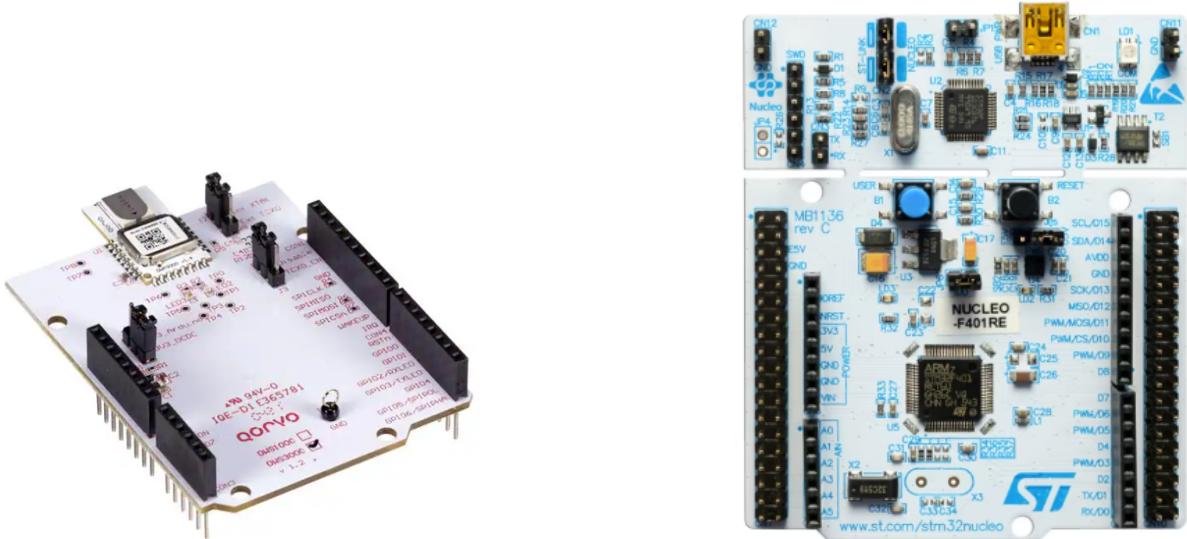
5.1 Localization of Individual Pixels

Goal: To design, implement, and evaluate a scalable and real-time localization system capable of tracking the 2D coordinates with sub-meter accuracy and low latency, using a Time Difference of Arrival (TDoA) scheme with UWB radios in a large semi-outdoor arena.

5.1.1 Hardware Setup

Tags(nodes)

Consists of Qorvo DW3000 UWB transceiver and STM32F446RET6 MCU. For ease of development Qorvo DWM3000 shield (based on DW3000 UWB transceiver) Figure 5.1a will be mounted on an STM32F446RE Nucleo development board Figure 5.1b. Each node will be battery powered and equipped with a WS2812B RGB LED for visual feedback. The STM32 MCU will run the firmware for UWB ranging, data processing, and LED control.



(a) DWM3000EVB Ultra-Wideband (UWB) Module Arduino Shield

(b) STM32F446RE Nucleo development board

Figure 5.1

Anchors

Use the same hardware (DWM3000 + STM32F446RE) but are stationary and mains-powered. Additionally a UART TTL to Ethernet converter module is connected to provide an RJ45 interface to wired Anchors to Master PC.

Master PC

A centralized computer runs the localization engine.

5.1.2 Deployment

Environment

50 x 20m Rectangular Semi Outdoor Space with less NLoS conditions.

Anchor Placement

A minimum of 4 anchors (minimum is 3 for 2D TDoA) will be placed at known, optimized coordinates at the corners of the 50m x 20m arena to ensure good Geometric Dilution of Precision (GDOP) [44]. Will consider 6-8 anchors to improve robustness to NLoS conditions, such that, anchors at corners + 2 mid-longside positions as shown in Figure 5.2. Anchors will be mounted in high trusses to maximize LOS fraction and minimize human body shadowing.

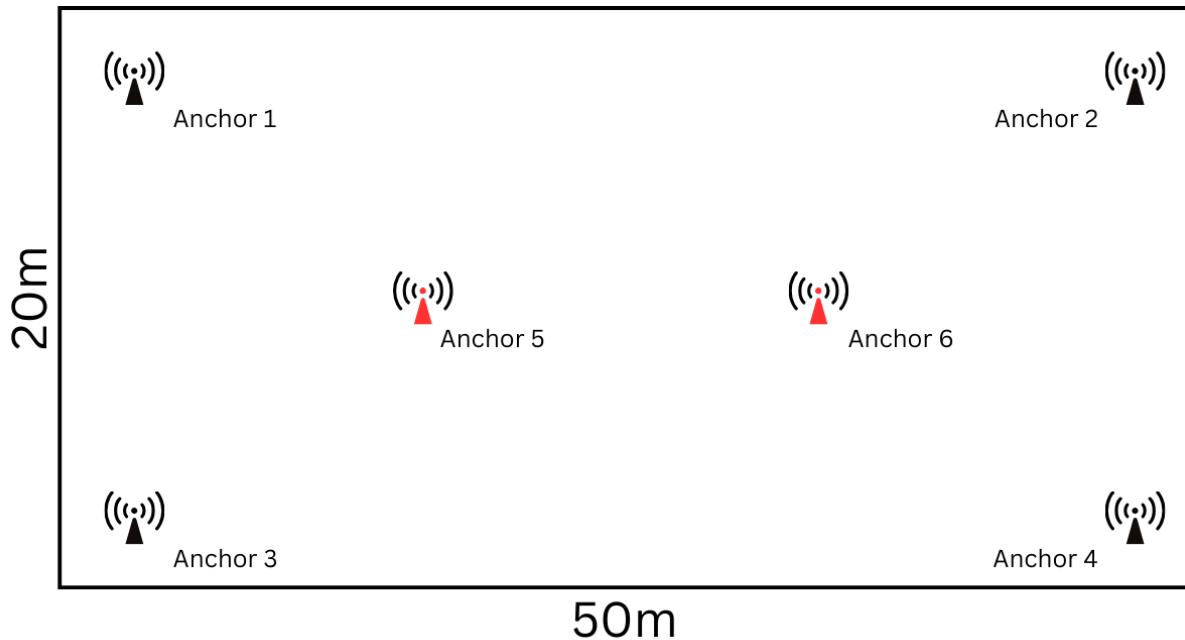


Figure 5.2: Anchor placement in a 50×20 m arena, with the main anchors positioned at the corners. Additional anchors, shown in red, are placed depending on the NLoS conditions.

TDoA ranging requires tight time synchronization between anchors and the reliable, real-time transfer of raw timestamp data to the Master PC. To achieve this, each anchor will be connected to the Master PC via a wired Ethernet link. The STM32 board at each anchor will interface with a UART-to-Ethernet module, enabling bidirectional communication. This setup allows the Master PC to send configuration commands and receive timestamp data from all anchors over a single, stable LAN connection, thereby avoiding wireless congestion. Note: Since the STM32

platform lacks built-in SLIP (Serial Line Internet Protocol) support, an ESP32 microcontroller may be used as a SLIP to Ethernet bridge for each anchor.

Tag Placement

Initially, several tags will be installed at surveyed, known positions to compute static localization error metrics. Subsequently, selected tags will be moved along a predefined path to assess dynamic accuracy, latency, and tracking robustness.

5.1.3 Ranging Scheme

The system uses TDoA-based ranging with centralized positioning, where the timestamps of arrivals are calculated at the anchors. This reduces massive transmissions from the nodes. Although this approach aligns with Ridolfi's scalability model, it differs from Muloc and SnapLoc implementations, which use node-side positioning and support an infinite number of devices.

Firmware Development for the nodes and anchors are done using Qorvo DW3000 driver API. It will provide all the relative function calls for getting the raw TDoA timestamps for each anchor.

As proposed by M. Ridolfi et al., and through proper tuning and testing, the PHY layer configurations can be adjusted using the Qorvo DW3000 Driver API to enhance scalability. Such as,

- Pulse Repetition Frequency (PRF)
- Preamble length and preamble code
- Data rate
- STS (Scrambled Time Sequence)
- UWB Channel selection

However, a custom MAC layer must be implemented. This logic comprises several key components,

- TDMA state machine to manage frame duration, time slots, and their assignment
- Scheduling logic layer
- Network management protocol

5.1.4 Localization Computation

Since the DWM3000 API provides only the fundamental primitives for obtaining Time Difference of Arrival (TDoA) timestamps, the actual localization algorithm must be implemented as a separate software component [45] defining this as “Central Localization Engine (CLE)”. The proposed pipeline consists of the following stages,

1. **Timestamp Acquisition:** For each tag transmission, the system collects an array of precise arrival timestamps from all receiving anchors, as provided by the DWM3000 API.
2. **Conversion to Distance Differences:** The time differences of arrival between anchor pairs are converted into corresponding distance differences by multiplying by the speed of light.
3. **Hyperbolic Equation Formation:** The estimated distance differences are used to form a system of nonlinear hyperbolic equations as shown in Eq 5.1. Hyperboloid where $(X_i - Y_i)$

are the coordinates of the anchor i , (X_j, Y_j) are the coordinates of the anchor j , and (x, y) are the coordinates of the tag.

$$L_{D(i,j)} = \sqrt{(X_i - x)^2 + (Y_i - y)^2} - \sqrt{(X_j - x)^2 + (Y_j - y)^2} \quad (5.1)$$

4. **Position Estimation via Nonlinear Least-Squares Optimization:** The system of equations is solved using a numerical optimization algorithm, such as the “Levenberg-Marquardt algorithm”, to find the tag position that minimizes the squared residuals across all anchor pairs [13]. This step effectively finds the point that best satisfies the set of hyperboloids. The values are the x,y coordinates of the localized node from the origin.

5.1.5 Evaluation / Performance Targets

Quantitative Metrics

- **Localization accuracy:** Sub-meter level accuracy. Median error $\leq 0.5m$
- **Update rate:** Should be localized within $\leq 1000ms(1Hz)$
- **Scalability:** Should be able to localize up to the amount suggested by M. Ridolfi et al. scalability model.

Test scenarios

- **Static Line of Sight Test:** Deploy tags (no of tags depends on Hardware Availability) at known positions scattered across the arena. Compute statistics. This measures best-case accuracy.
- **Dynamic Line of Sight Test:** Deploy tags (no of tags depends on Hardware Availability). Measure tracking at target update rate, inter-tag independence, and interference.
- **Large-scale simulation and Interference Test:** Use ns-3 Network Simulator [46] to test the scalability as well as to model MAC collisions, anchor schedule, update rate, and nominal packet loss. Also to validate Ridolfi scaling predictions and choose TDMA schedule parameters.

5.2 Pattern Orchestration (Locations \rightarrow Light Commands)

Goal: To map a moving, irregular point-cloud of participants (wearing LED devices) to target visuals that remain coherent and stable despite continuous motion, uncertainty, and density variations. The algorithm should be robust, scalable, and responsive in real time Figure 5.3.

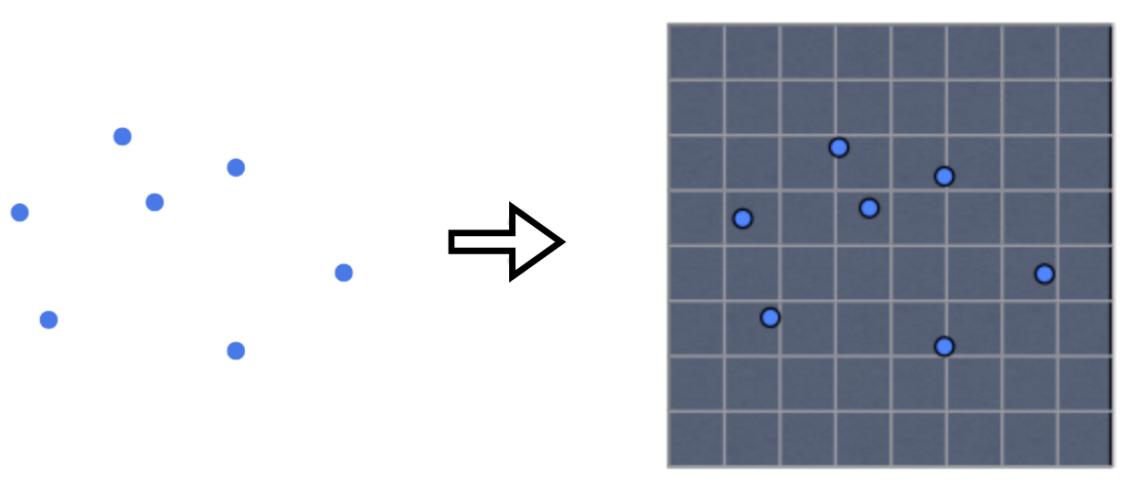


Figure 5.3: Mapping of physical locations of the devices to a visual digital point cloud at time t .

Sampling and Assignment - a high level idea For each participant:

1. Sample the visual field at the normalized position.
2. Extract the corresponding color, intensity, or pattern value.
3. Transmit the light command to the participant's LED device.
4. Ensure the update loop runs at sufficient frequency to handle interpolation, synchronization, and communication overhead in real time.

5.2.1 Procedure

Two methods are considered in this approach: one prioritizing **ease of implementation** for straightforward visualization, and the other offering **greater flexibility and richer effects** such as procedural patterns, continuous gradients, or flow fields that go beyond simple pixel-based imagery.

Method 1: Coordinate Normalization

- Align raw participant coordinates to a fixed show frame.
- Use anchors and tagged reference points to ensure consistency between real-world positions and designed visual effects.
- Transform all participant positions into normalized coordinates (e.g. scaled to an $[0, 1] \times [0, 1]$ stage grid).

This method provides a simple, robust way of mapping participants to a fixed visual space, making it straightforward to implement and reliable in real-time scenarios.

Method 2: Visual Field Synthesis

The notion of “continuous visual field mapping” is not a widely established term in the literature but is introduced here as a conceptual distinction between two approaches:

- Direct Mapping (in method 1) : Each discrete LED/pixel samples color directly from an image.

- Field-Based Mapping: A continuous function (a “visual field”) is first defined over the display space to encode the desired effect. Any participant’s position can then be queried against this function, treating the display not as a fixed pixel grid but as a continuous domain. Each participant’s LED thus becomes a sample point of this visual field.

Steps in the procedure include:

1. Generating a target visual field over the normalized show frame.
2. Encoding desired visual attributes such as color, intensity, gradients, or procedural effects using either:
 - **Direct Image Mapping:** A fixed image where pixel values represent colors.
 - **Continuous Visual Field Mapping:** A function $F(x, y)$ that defines visual values for any coordinate.
3. Optionally apply interpolation (e.g. bilinear, Gaussian kernels, or neural field approximations) to ensure smooth transitions.

This method enables smoother, more adaptive visuals that can accommodate irregular crowd distributions and dynamic movement patterns.

5.2.2 Evaluation / Performance Targets

Stability

- Visuals remain coherent even under irregular spacing and movement.
- Field-based methods provide resilience to density fluctuations and missing participants.

Latency and Responsiveness

- Update rate: $\leq 3Hz$
- End-to-end latency (location input → assignment → sending to transmission): $\leq 1s$.

Scalability

- Support up to thousands of pixels with parallelized sampling.

Accuracy / Visual Fidelity

- **Direct image mapping:** Clear reproduction of source image when participant distribution is uniform.
- **Continuous field mapping:** Smoother effects and robustness under non-uniform, sparse, or dynamic distributions.

Robustness

- Handles uncertainty in localization (noise, drift, dropout).
- Field-based approach ensures global coherence even when local data is incomplete.

6. Project Progress

6.1 Finalized System Architecture

The system architecture is a unified framework integrating physical hardware, a high-fidelity simulator, and a core application layer. This design enables seamless transition between real-world deployment and software-based testing. As shown in figure xx the complete system consists of 3 main components,

1. **Hardware-Based Indoor Localization System:** This component represents the physical deployment layer responsible for real-time tracking.
2. **Simulation Environment:** A digital twin built to replicate the hardware system, allowing for development and validation without physical constraints.
 - **Human Generator:** Simulates realistic crowd dynamics, creating random and complex movement paths to mimic human behavior.
 - **Localizer:** A software-based engine that models UWB signal propagation and executes the localization algorithms (TDoA) to output virtual tag coordinates.
3. **Core Application Logic:** The central processing application that drives the visual experience, agnostic to whether the input comes from hardware or simulation.
 - **Pattern Generator:** A mathematical engine that creates complex, algorithmic visual effects and color waves based on spatial coordinates.
 - **Synchronizer:** The bridge module that maps the generated visual patterns to specific Tag IDs, transmitting the correct lighting commands to each user in real-time.

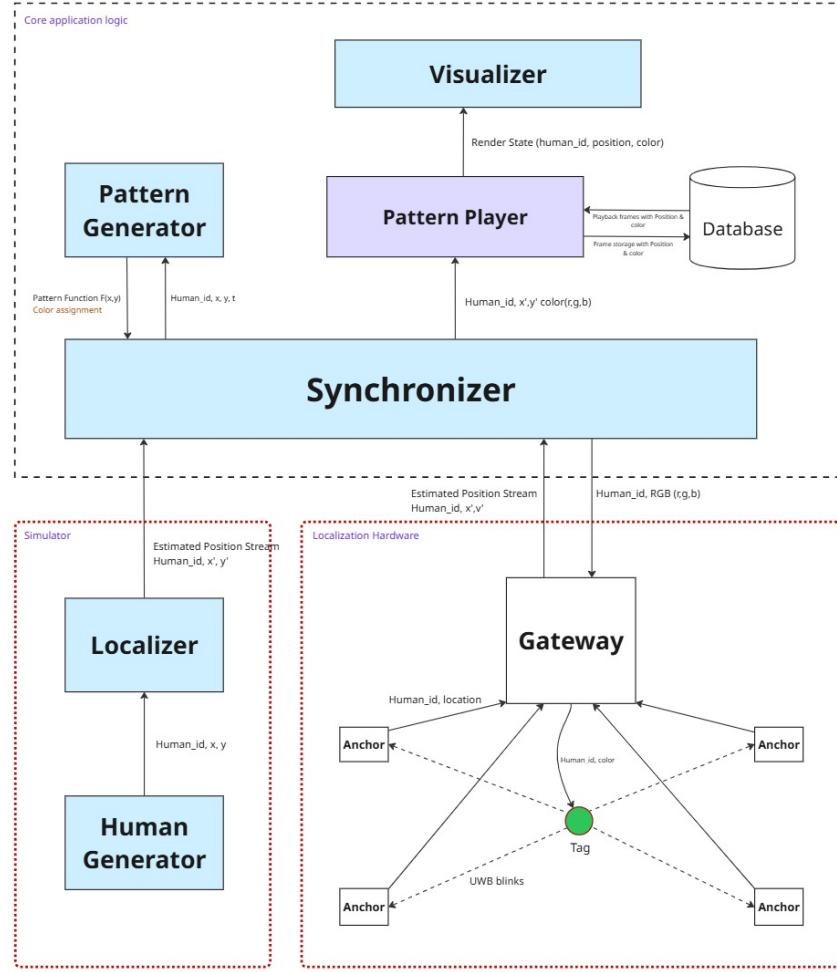


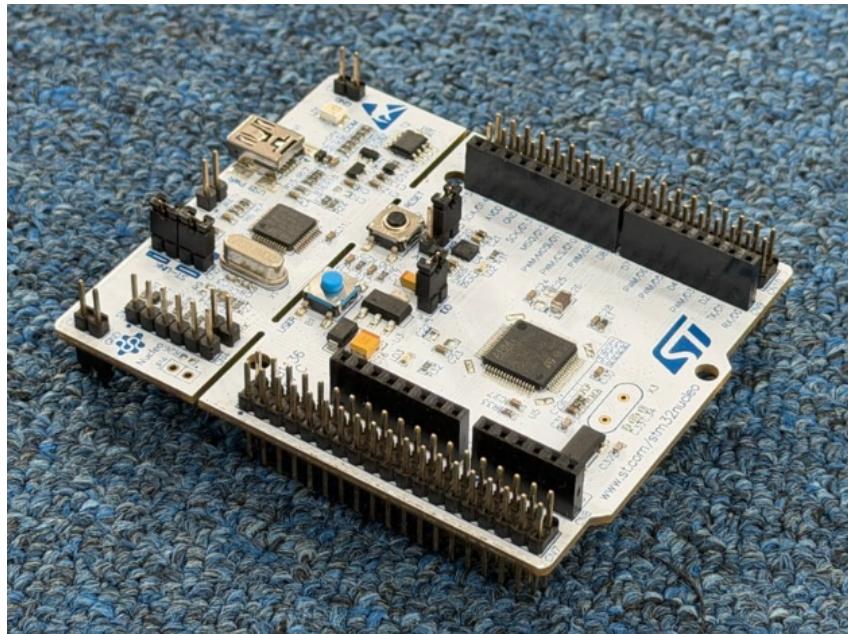
Figure 6.1: CrowdSync High Level System Architecture.

6.2 Hardware Localization Scheme

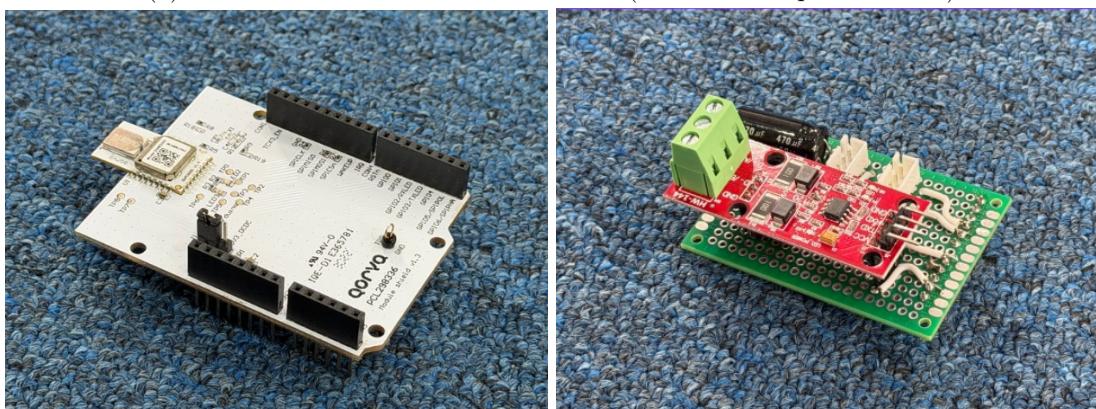
The hardware Localization Scheme is primarily used for localizing the tags using Ultrawide band (UWB) technology. The physical layer of the localization system comprises three primary components such as Tags which are dynamic in movement, anchors fixed in known positions and a Gateway which collects ranging data from anchors and runs the localization engine to calculate the position of each tag.

6.2.1 Anchor and Tag Design

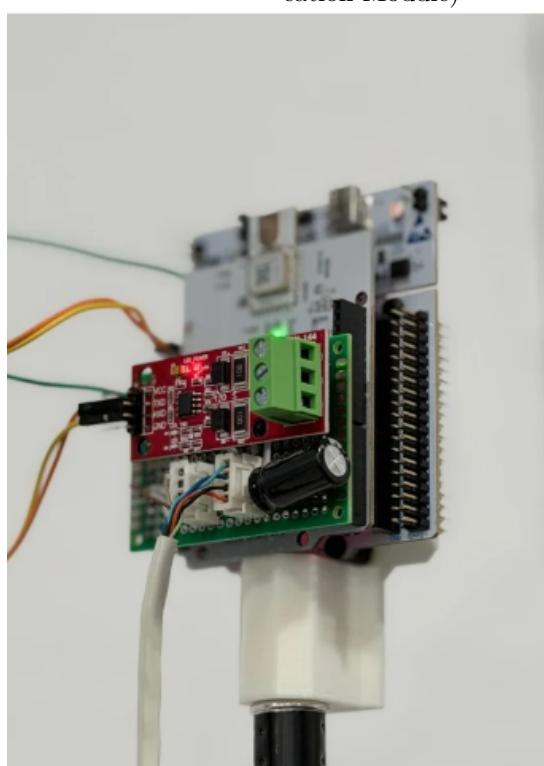
The Anchor nodes serve as the fixed reference points within the localization infrastructure. Each Anchor consists of a STM32 Microcontroller and interface with a UWB transceiver module.



(a) Microcontroller STM32F446RET6 (Nucleo development board)



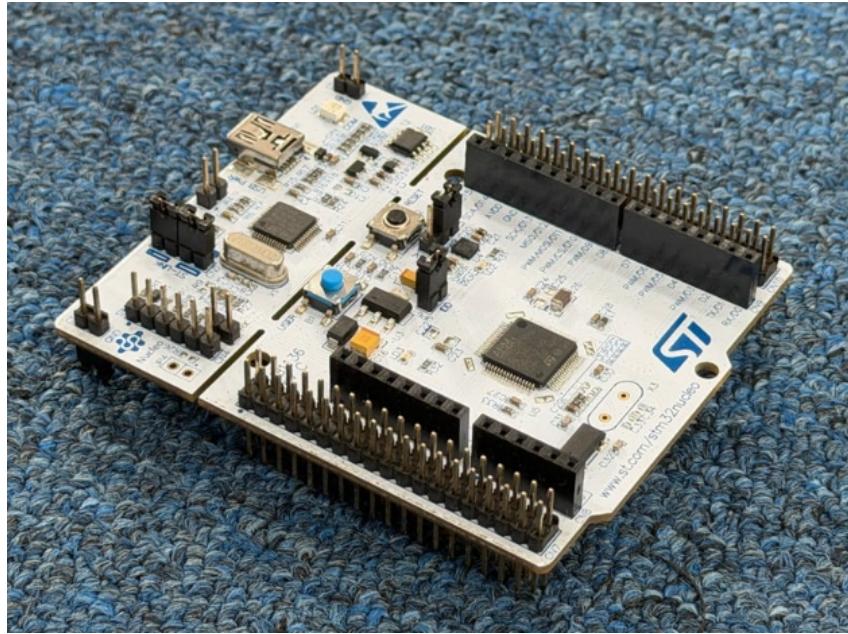
(b) UWB Module DW3000 (DWM3000EVB) (c) Top Shield (Power Delivery + Communication Module)



(d) final Anchor setup with tripod

Figure 6.2: Anchor Design and Setup. The Anchor consists of a STM32 Nucleo development board interfaced with a DWM3000EVB Shield, which contains the DW3000-UWB transceiver.

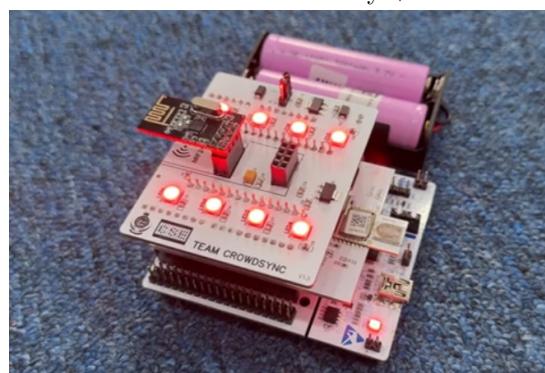
The Tag Nodes are designed for mobility and energy efficiency. They feature a minimized form factor and sleep-mode logic to extend battery life. Unlike the anchors, the tags operate primarily in a responsive or strictly transmission-based state depending on the localization scheme employed (TWR or TDOA). The Tag nodes are also made using the same hardware as Anchor hardware with STM32 Microcontroller and UWB Transceiver but with a custom developed shield with leds and voltage regulators and the tag is battery powered.



(a) Microcontroller STM32F446RET6 (Nucleo development board)



(b) UWB Module DW3000 (DWM3000EVB) (c) Receiver and Lighting Module (Power Delivery + Communication Module)



(d) final Tag setup with battery

Figure 6.3: Tag Design and Setup. The Tag consists of a STM32 Nucleo development board interfaced with a DWM3000EVB Shield, which contains the DW3000 UWB transceiver. The top PCB shield is custom-designed consists of RF Module and voltage regulators. The complete Tag setup is powered by a battery for mobility and ease of deployment within the localization environment.

6.2.2 System Backbone and Gateway Integration

The system employs a wired backbone to connect with the gateway. This wired backbone is primarily used for Data Aggregation, Wired Synchronization and Power Delivery. The Gateway, a Raspberry Pi 4 acts as the bridge between the localization network and the central processing server (pattern generator and the Synchronizer).

Rationale for Wired RS-485 Backbone

We selected the RS-485 standard for the communication backbone rather than a wireless mesh for the following reasons,

- **Noise Immunity:** RS-485 utilizes differential signaling, which provides superior immunity to electromagnetic interference (EMI), a critical factor in crowded or industrial environments where Crowd Sync is deployed.
- **Cable Length and Topology:** It supports long-distance cabling (up to 1200m) in a daisy-chain as well as star topology [47]. This will improve the scalability of the system.
- **Determinism:** A wired bus eliminates the nondeterministic packet loss and latency associated with wireless backhaul, ensuring that timestamp data reaches the solver reliably.

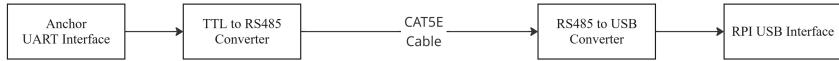


Figure 6.4: The anchor STM32 MCU’s UART interface is connected to a TTL-to-RS485 converter module. On the other end, an RS485-to-USB converter is used and connected to the Raspberry Pi via its USB interface. In later stages, this connection will be migrated to the Raspberry Pi’s native USART interface to enhance scalability. A CAT-5e cable is used to interconnect the modules, as it provides reliable signal transmission over long distances and offers improved noise immunity due to its twisted-pair structure and shielding against electromagnetic interference

6.3 Localization Methods: Double-Sided Two-Way Ranging (DS-TWR)

To mitigate the clock drift errors inherent in Single-Sided TWR, we implemented Double-Sided Two-Way Ranging (DS-TWR) as mentioned in [48] . This method involves three messages per ranging exchange, allowing the system to cancel out clock frequency offsets between the tag and the anchor.

6.3.1 Methodology

In standard Single Sided TWR the initiator transmits a radio message to the responder and records its time of transmission (transmit timestamp) t_1 . The responder receives the message and transmits a response (a radio message) back to the initiator after a particular delay t_{reply} . The initiator then receives this response and records a receive timestamp t_2 .

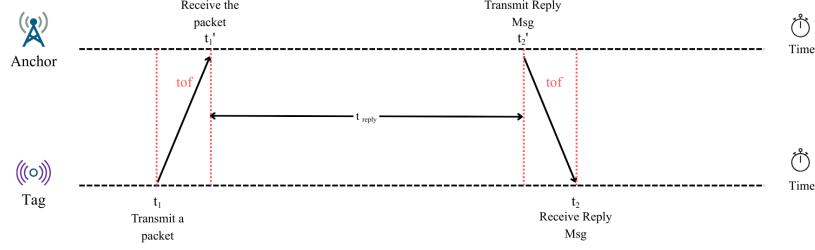


Figure 6.5: High Level diagram on how messages are exchanged on standard Two Way Ranging

Now using the timestamps t_1 and t_2 , the initiator can calculate the round trip time troundtrip and knowing the reply time in the tag, treply, the TOF can be determined by by equation,

$$t_{roundtrip} = t_2 - t_1 \quad (6.1)$$

The Double Sided TWR (DS-TWR) reduces the error that happens due to clock and frequency drift. [50]

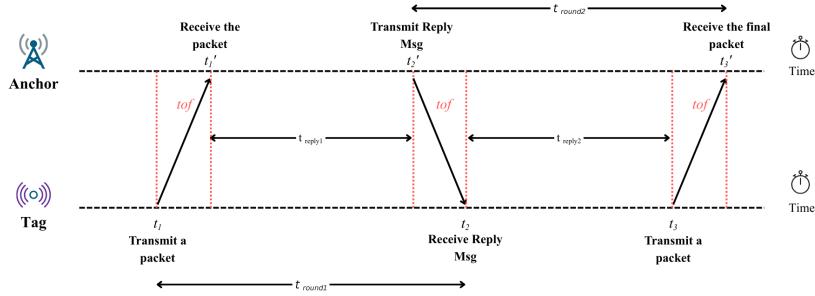


Figure 6.6: High Level diagram on how messages are exchanged on Double Sided Two Way Ranging

The Time of Flight Tprop is calculated using the following expansion:

$$t_{prop} = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \quad (6.2)$$

Where Tround represents the round-trip time measured by the initiator, and Treply represents the response delay measured by the responder.

Implementation Variants and Scalability Analysis

We implemented and compared two variations of this protocol to assess network capacity.

Tag-Initiated DS-TWR (Standard Approach)

In this configuration, tags wake up and transmit a Poll message asynchronously. Each tag operates in a round-robin manner, where it performs ranging with the anchors in turn. Overall, this effectively operates as a random-access ALOHA protocol.

Limitation: As tag density increases, the probability of packet collisions rises exponentially. Since there is no central coordination, multiple tags attempting to range simultaneously result in mutual interference and packet loss, severely limiting scalability.

Anchor-Initiated DS-TWR (Optimized Approach)

To address the collision bottlenecks, we inverted the control logic. The Anchor acts as the master coordinator for UWB network. This allows the system to schedule ranging events centrally. This includes following stages

1. **Registration Phase:** When a tag first enters the network, it performs a brief registration handshake. The anchor adds this tag ID to a known registry.
2. **Round-Robin Scheduling:** Once registered, the anchor iterates through its list of known tags, initiating ranging with each tag sequentially in a Round-Robin fashion.

Scalability Improvement: By deterministically scheduling which tag communicates at any given millisecond, we eliminate RF collisions during the ranging phase. This centralized control ensures that airtime is utilized efficiently, allowing for a higher density of tags compared to the random-access nature of the tag-initiated approach. However, in both approaches, the localization frequency (frame rate) decreases as the number of tags increases.

6.3.2 Analysis of Results and Filtering

Raw distance measurements from DS-TWR are susceptible to noise from multipath propagation and Non-Line-of-Sight (NLOS) conditions. To enhance position accuracy, we tried out several filtering strategies,

1. **Initial Noise Reduction (Linear Kalman Filter)** Initially, a standard 2D Linear Kalman Filter (LKF) was applied to the raw coordinates derived from trilateration. While this successfully smoothed high-frequency jitter, it struggled to track dynamic movement accurately, often introducing lag during rapid turns.
2. **Extended Kalman Filter (EKF) with Constant Velocity Model** To address the dynamic limitations of the LKF, we transitioned to an Extended Kalman Filter (EKF) utilizing a Constant Velocity (CV) process model. This model assumes that the tag's velocity remains constant between short sampling intervals, with deviations modeled as a discrete time white noise model. The state vector x_k is defined as,

$$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \quad (6.3)$$

The process noise is modeled as discrete white noise acceleration (DWNA), allowing the filter to account for sudden changes in speed or direction as "acceleration noise" rather than measurement error.

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \quad (6.4)$$

This advanced filtering significantly reduced the Root Mean Square Error (RMSE) compared to raw trilateration, providing smoother trajectories even when individual ranging packets were lost or corrupted.

To evaluate the efficacy of the filtering pipeline, a static test was conducted where the tag was placed at a fixed known coordinate for a duration of $N = 256$ samples. We compared

the performance of the Raw Trilateration, Linear Kalman Filter (KF), and Extended Kalman Filter (EKF) using Root Mean Square Error (RMSE) and Standard Deviation (SD) as key performance metrics.

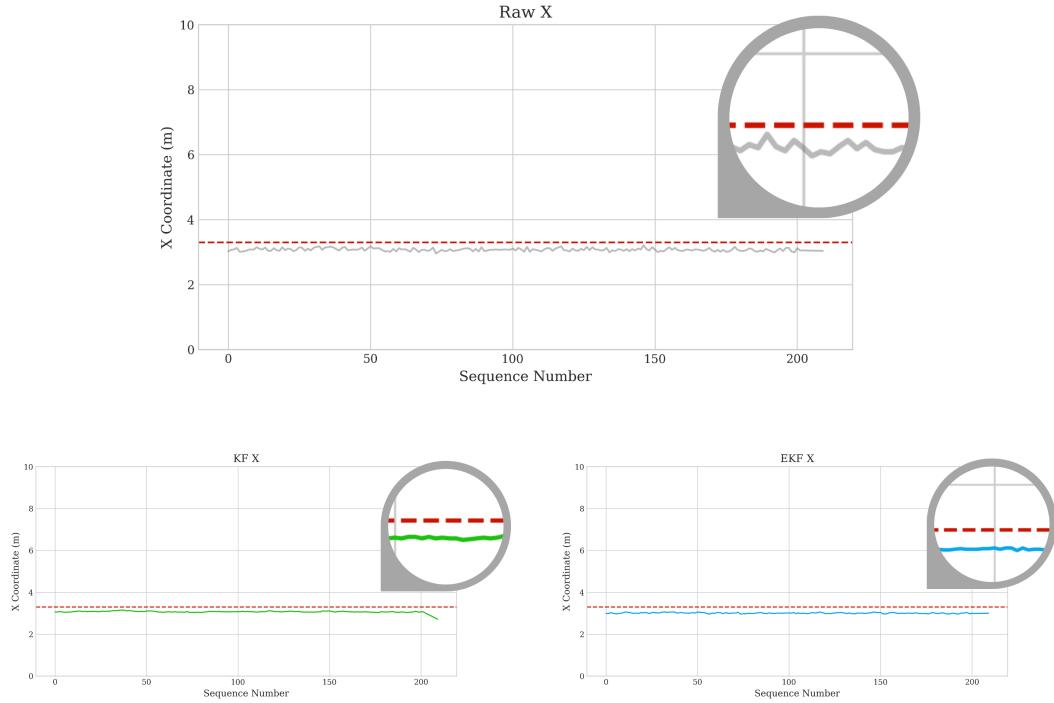


Figure 6.7: Time-series analysis of the X-coordinate estimation during static testing. The plot compares the raw trilateration output (gray) against the Linear Kalman Filter (green) and Extended Kalman Filter (blue), highlighting the noise reduction capability of the EKF.

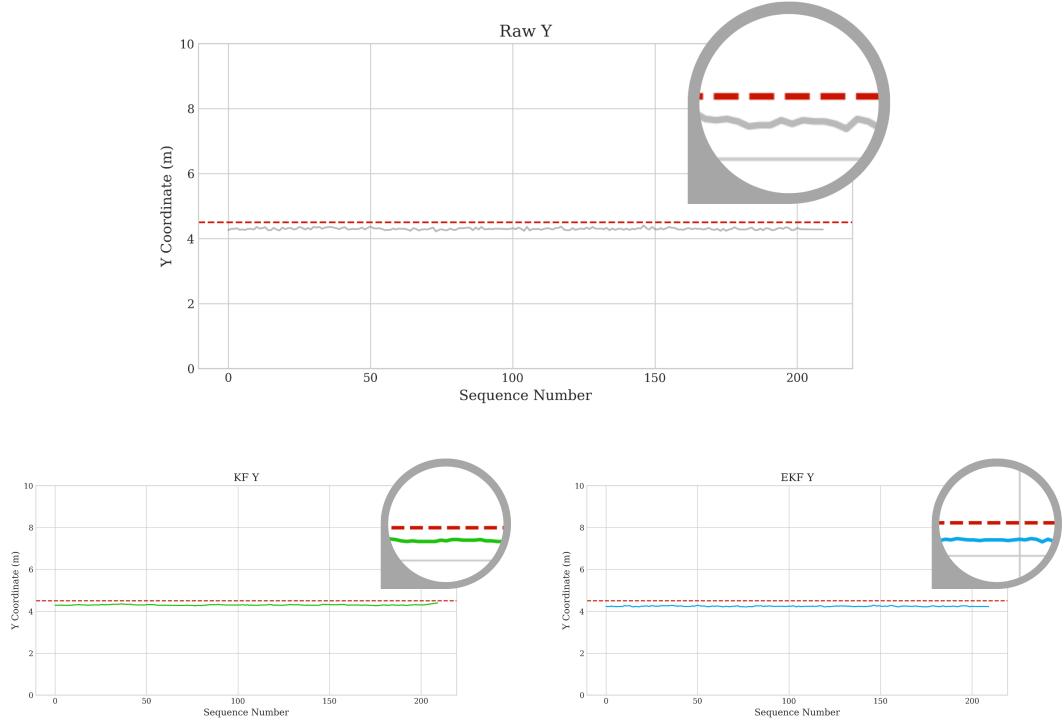


Figure 6.8: Time-series analysis of the Y-coordinate estimation. The EKF response demonstrates significant attenuation of high-frequency jitter compared to the raw signal and standard KF but stable over the time.

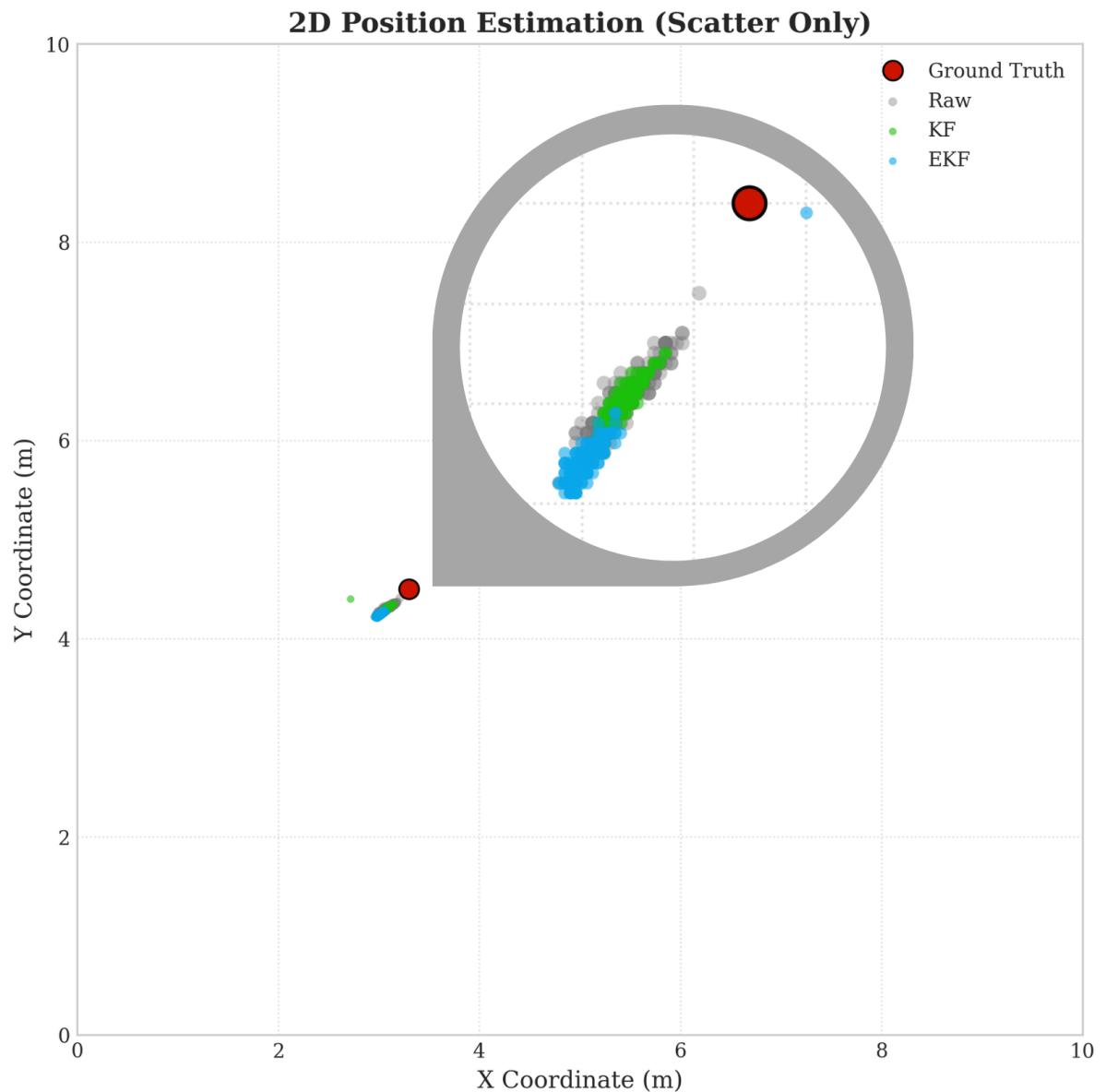


Figure 6.9: 2D Spatial Scatter plot of estimated positions relative to the ground truth. The dispersion of the raw data points (wide scatter) contrasts with the tight clustering of the EKF estimates, visually representing the improvement in precision.

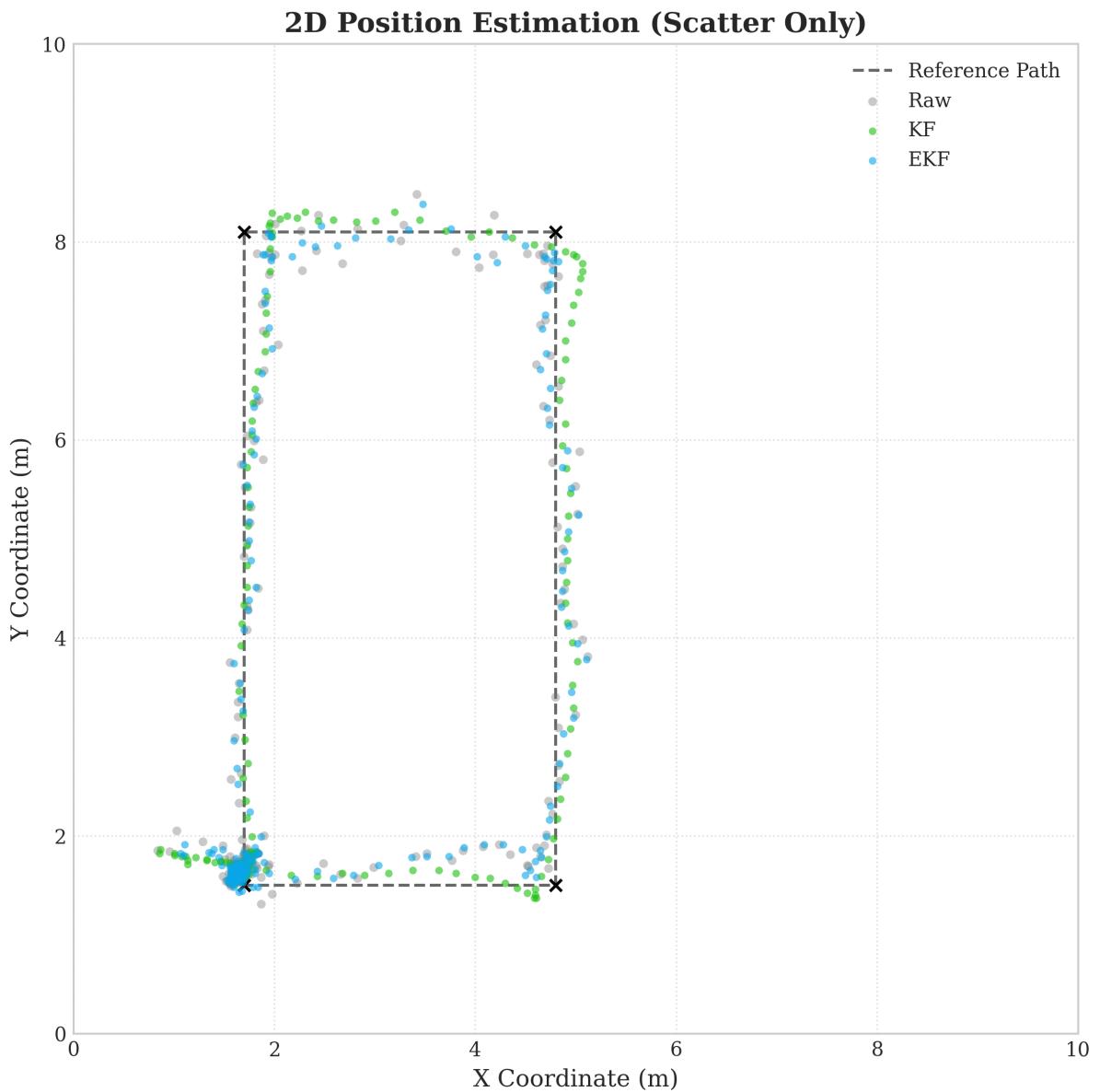


Figure 6.10: Dynamic test data showing the trajectory of a moving tag over time. The raw trilateration points (gray) exhibit significant noise and outliers, while the EKF estimates (blue) form a coherent path that closely follows the expected movement pattern, demonstrating the filter's ability to maintain stability in dynamic conditions.

Method	RMSE(m)	MEan(m)	Std Deviation(m)
Raw	0.3046	0.2999	0.0533
KF	0.3027	0.3008	0.037
EKF	0.3911	0.3901	0.0272

Table 6.1: Performance Metrics for Static Localization Test.

The static test data highlights a critical trade-off between absolute accuracy and signal stability.

- **Precision and Stability:** The Extended Kalman Filter (EKF) demonstrated superior noise suppression, reducing the Standard Deviation () by approximately 50% (from 0.0533m to 0.0272m).
- **Accuracy and Bias:** While the EKF shows an increase in RMSE (0.3046 m) compared to the raw input (0.3911 m), this discrepancy is attributed to the filter converging on a systematic bias rather than random error.

Conclusion: The implementation prioritizes precision over absolute accuracy. Although the EKF does not correct the mean calibration offset, it successfully transforms a noisy, fluctuating signal into a smooth, coherent trajectory , significantly improving the system's stability.

6.4 Localization Methods: Time Difference of Arrival (TDOA)

To achieve the massive scalability required for the full CrowdSync deployment, we transitioned to a Time Difference of Arrival (TDOA) scheme. In this architecture, tags broadcast a single "blink" message. Multiple anchors receive this message, and the position is calculated based on the difference in arrival times.

6.4.1 The Challenge of Time Synchronization: The Need for a Single Time Domain

For TDoA to function, all anchors must operate within a unified time domain. Mathematically, the TDoA equation assumes that TRx,A and TRx,B are measured against the exact same reference zero. If Anchor A's clock is even slightly ahead of Anchor B's, the resulting time difference will be skewed, translating directly into a position error. In UWB systems, where signals travel at the speed of light (C 30 cm/ns), a synchronization error of just 1 nanosecond results in a 30 cm positioning error.

Clock Drift and Offset

Even if two anchors are powered on at the exact same instant, their internal clocks will immediately begin to diverge. This is due to two primary factors:

- **Initial Offset:** No two startup sequences are identical. One oscillator may stabilize a few microseconds faster than another, creating a constant time bias from t=0.
- **Clock Drift:** The crystal oscillators (XTALs) that drive these clocks are physical components subject to manufacturing tolerances, temperature variations, and voltage fluctuations. A "10 MHz" crystal might actually oscillate at 10.00001 MHz on Anchor A and 9.99999 MHz on Anchor B. Over time, these minute frequency differences accumulate, causing the time count on each anchor to drift apart linearly or non-linearly.

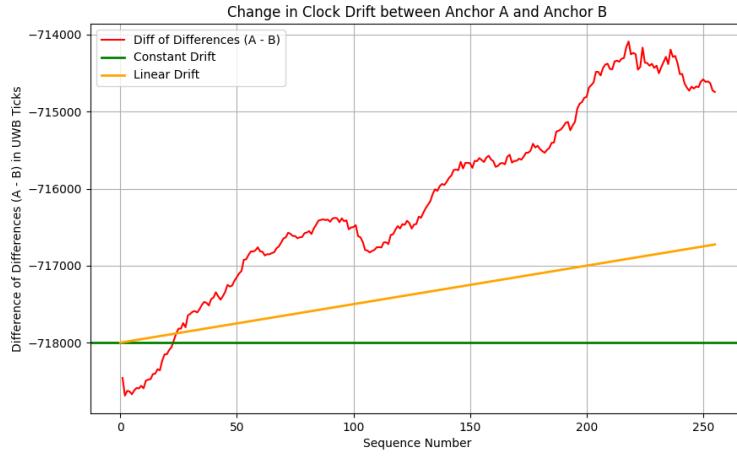


Figure 6.11: Analysis of non-linear clock drift between two anchor nodes

Figure 6.11 shows that clock drift between two anchors that start synchronously (at the same time) while the tag remains stationary. Over time, the differences between the difference of the received raw timestamps (red line) for the same sequence number is not constant and does not vary linearly. Instead, it exhibits nonlinear behavior due to clock drift caused by frequency mismatch and instability in the anchors' crystal oscillators.

6.4.2 The Ideal (but Impractical) Solution

The theoretically perfect solution would be to drive all anchors using a single, high-precision external oscillator connected via cables. This would ensure that every clock tick happens simultaneously across the entire network. However, this is highly impractical for real-world deployments due to the cost, complexity, and signal degradation associated with running high-frequency clock lines over long distances.

6.4.3 Practical Synchronization Methodology

Since a single physical clock is impractical, the standard industry approach is to model the relationship between the separate clocks mathematically. This involves designating a "Master Clock" (reference time) and periodically correlating the "Slave Clocks" to it.

6.4.4 The Master-Slave Sync Model

A Master Anchor broadcasts a synchronization pulse (either wired or wireless) at fixed intervals (e.g. every 100ms).

1. **Reference Event:** The Master sends a pulse at global time T_{Global} .
2. **Local Capture:** Each Slave receives this pulse and records its local timestamp T_{Local}
3. **Offset and Drift Calculation**

6.4.5 Method 1 Linear Regression (Statistical Smoothing)

This method is designed to mitigate high-frequency noise (jitter) in the synchronization signal, common in wired trigger setups. The system maintains a sliding window of the last N synchronization pairs. We fit a linear equation $y = mx + c$ to these points using the Least Squares method, where y is Global Time and x is Local Time.

- **Slope m:** Represents the relative frequency difference (clock drift rate) between the Master and Slave crystals.
- **Intercept c:** Represents the time offset at $t = 0$.

Prediction: When a Tag Blink arrives at local time $T_{Tag, Local}$, the corrected time is calculated as:

$$T_{Tag, Global} = m \cdot T_{Tag, Local} + c \quad (6.5)$$

This method acts as a low-pass filter. It is excellent for smoothing out the "jittery" edges of a wired sync pulse. However, it creates "inertia," meaning the system reacts slowly to sudden changes in crystal frequency (e.g., caused by rapid temperature shifts).

6.4.6 Method 2: Extrapolation (Instantaneous Correction)

This method, derived from the ATLAS[49] system architecture, assumes the synchronization timestamps are precise but the clock drift is continuous. It uses only the most recent synchronization interval to calculate parameters. The model calculates the instantaneous clock skew based on the difference between the current sync pulse k and the previous sync pulse k- 1.

Equations from ATLAS[49]:

1. **Offset Calculation (sigman,k):** The difference between the ideal sync arrival time ($t_{r,k}$) and the measured local time ($t_{Sn,k}$):

$$m = \frac{T_{Global,k} - T_{Global,k-1}}{T_{Local,k} - T_{Local,k-1}} \quad (6.6)$$

2. **Drift/Skew Estimation (s):** The rate of change of the offset over the sync interval (s):

$$s = \frac{c - c_{prev}}{T_{Local,k} - T_{Local,k-1}} \quad (6.7)$$

3. **Tag Time Correction:** When a tag frame arrives at local time $T_{Tag,Local}$, the offset is extrapolated:

$$T_{Tag, Global} = T_{Tag, Local} + c + s \cdot (T_{Tag, Local} - T_{Local,k-1}) \quad (6.8)$$

4. **Final Global Time:**

$$T_{Tag, Global} = m \cdot T_{Tag, Local} + c + s \cdot (T_{Tag, Local} - T_{Local,k-1}) \quad (6.9)$$

5. **Wireless Adaptation:** For the wireless implementation, the Offset c was further adjusted to subtract the Time-of-Flight (ToF) of the synchronization packet DistanceMaster to Slave C, ensuring the Slave clock is aligned with the Master's transmission time, not its arrival time.

6.4.7 Experimental Setup and Results for TDOA

To validate synchronization, we placed a tag at the exact geometric center between two anchors. Ideally, the Time Difference of Arrival (t) should be zero. Any deviation from zero represents the synchronization error + measurement noise.

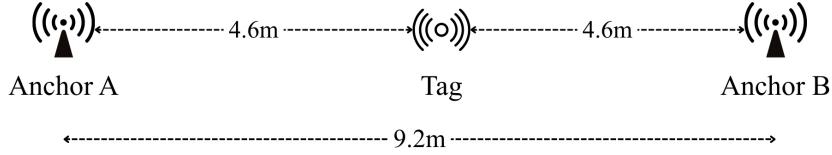


Figure 6.12: TDOA Clock Synchronization Test Setup for Two Anchors and a stationary Tag in the exact middle between the anchors

Wired Synchronization Results (Using Method 1)

- Implementation** Wired GPIO triggers from raspberry pi using Linear Regression ($N = 10$). Observation: The wired signal had significant edge jitter (100-500ns) and unknown phase shifts.
- Outcome** Method 1 (Regression) stabilized the readings, but the resulting position data still exhibited a "wandering baseline" error and overtime shifts to different values due to the physical limitations of the wire capacitance.

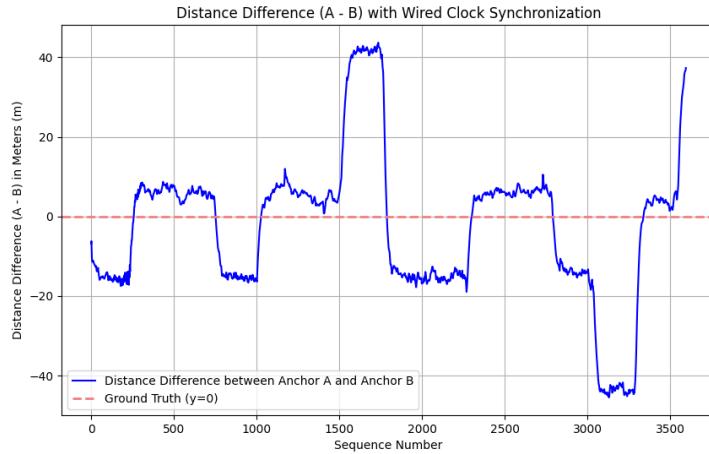


Figure 6.13: Uneven Distance Shifts Caused by Wired Anchor Synchronization Imperfections

The figure 6.13 shows the behavior when using the wired synchronization method. The distance differences are calculated from the timestamp differences of tag blink receptions at Anchor A and Anchor B, multiplied by the speed of light. Even after applying basic outlier rejection and a basic Kalman filter, the results exhibit uneven shifts. Ideally, the expected plot should be a straight line with acceptable noise and a constant offset. The observed phase shifts are likely caused by errors introduced by the wired synchronization setup, including physical and electrical differences in the synchronization cables, as well as variations due to environmental or connection changes.

Wireless Synchronization Results (Using Method 2)

- Implementation** Periodic UWB broadcasts (10Hz) using the ATLAS[49] Extrapolation method. The DW3000's dwtsetdelayedtrxtime feature ensured packet transmission with sub-nanosecond precision.
- Outcome** Because the input data was precise, Method 2 (ATLAS) performed superiorly. It correctly tracked the crystal drift (10 ppm) without the lag associated with regression.

- **Precision:** The raw, unfiltered data exhibits a standard deviation of 7.4cm. This low variance indicates that the wireless synchronization logic is successfully maintaining sub-nanosecond coherence between anchor clocks, effectively mitigating the random jitter often associated with wireless packet exchange.
- **Offset Analysis:** A consistent mean offset of approximately -5.1 meters is observed in the data. This systematic error is attributed to the geometric placement of the Master Anchor, which was not equidistant from the Slave Anchors during this specific test setup. However, the constancy of this offset confirms that the synchronization lock is stable and does not drift over time.

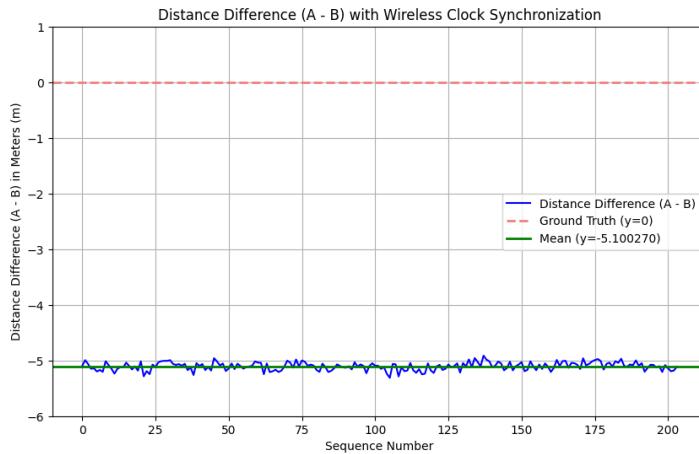


Figure 6.14: Comparative analysis of wireless synchronization stability using midpoint tag validation. The plot displays the raw TDOA measurements, demonstrating a high degree of stability ($= 7.4\text{cm}$), despite a constant geometric offset (-5.1m) due to Master Anchor placement offset.

Conclusion

The experimental results demonstrate that for any synchronization methodology whether wired or wireless to be successful, two critical conditions must be strictly met:

1. **Precision of the Master Clock Sync Events:** The Master Clock must generate synchronization signals with absolute periodicity. This was proven in the wireless experiment, where utilizing the DW3000's hardware-based `dwtsetdelayedtrxtime` feature eliminated software-induced jitter. By triggering transmissions based on the exact tick count of the internal system clock rather than CPU processing loops, timing inconsistencies were effectively removed.
2. **Consistency of the Transmission Medium** The medium through which the synchronization signal travels must be identical and predictable for all slave nodes.
 - **Wired Limitations:** In the wired experiments, physical discrepancies such as mismatched cable lengths, varying impedance, and signal noise introduced significant phase shifts and edge jitter. These hardware imperfections caused the "wandering offset" observed in the data, limiting the achievable accuracy.
 - **Wireless Superiority:** The wireless mechanism proved superior in this regard. In Line-of-Sight (LoS) conditions, the UWB signal travels at the constant speed of light C in a predictable manner for all anchors. By mathematically accounting for the

Time-of-Flight (ToF), the wireless approach bypassed the physical inconsistencies of cabling, resulting in a significantly more stable and accurate synchronization grid.

6.5 Core Application Logic

6.5.1 Human Generator (Simulation)

This is a '**high-performance crowd simulation engine**' that generates and updates the positions of simulated humans within a venue. It models random Brownian-like motion with damping.

To achieve high performance, **OpenMP (Open Multi-Processing)** is utilized in the generator code to instruct the compiler to distribute loop iterations across multiple CPU cores. This is implemented using the `#pragma omp parallel for` directive. Each core executes its assigned chunk of the loop simultaneously, drastically reducing the total time taken.

SIMD (Single Instruction, Multiple Data), which is a hardware-level acceleration technique that allows a single CPU instruction to operate on an entire "vector" of data simultaneously, is also used to further boost efficiency by allowing a single CPU instruction to process multiple data points at once.

In contrast, SIMD allows the processor to pack multiple values into specialized, wide registers. **SoA (Structure of Arrays)** layout rather than the more common **AoS (Array of Structures)** is the critical data layout prerequisite that enables SIMD to work efficiently. Because x-coordinate values are being contiguous in memory (with SoA layout), by utilizing 256-bit **AVX2 (Advanced Vector Extensions)** registers, the system can process four 64-bit double-precision values in a single clock cycle. To make this work optimally, memory is aligned using `alignas(32)`, ensuring the data is perfectly positioned for the CPU to "grab" it at maximum speed.

When combined, OpenMP provides multi-core scaling and SIMD provides instruction-level speed, resulting in a "**force multiplier**" effect that maximizes the total throughput of the generator.

The screenshot shows a window titled "CrowdSync - Real-time Stats". It displays performance metrics: "Human Generator: 0.432 ms" and "Pattern Mapping: 4.036 ms (avg: 4.036 ms)". Below this, it says "Showing 0 - 30 of 10000 humans (Scroll: 0%)". A table follows, with columns "Human_ID", "X_Coord", and "Y_Coord". The data shown is:

Human_ID	X_Coord	Y_Coord
0	26.8198	11.458
1	1.3075	25.267

Figure 6.15: Human generation delay per a frame is less than 0.5 ms

6.5.2 Localization Estimator

This is the mathematical "engine" that bridges the gap between the perfect simulation world and messy real-world physics. It takes the **True Positions** of humans from the Human Generator

and converts them into **Estimated Positions** by simulating the behavior of Ultra-Wideband (UWB) sensors.

The process ensures the simulation is not "too perfect," adding the natural inaccuracies found in wireless localization systems. The estimation happens in two distinct mathematical stages:

1. **Simulating the Signal:** In this first phase, the system mimics how UWB anchors at the venue corners would "hear" a tag carried by a human. For each human, the system calculates the exact distance to four fixed anchors (placed at the venue corners) and determines how long a signal would take to travel that distance. To make it realistic, this component adds **Gaussian Noise** to these times. This represents real-world interference, such as signal bouncing (multipath) or atmospheric conditions, that makes measurements slightly "fuzzy."
2. **Solving the Position:** In this second phase, once the system has these noisy "arrival times," it works backward to figure out where the human is. It uses a **Gauss-Newton (GN) Solver**. Finding a position based on time differences is a non-linear problem. As the solution, the GN solver starts with an "initial guess" and iteratively refines it. It adjusts the (x, y) coordinates until the calculated distances to the anchors best match the noisy timing data it received.

6.5.3 Pattern Generator

To achieve infinite visual variety without the heavy memory footprint of traditional textures, the system utilizes a procedural generation pipeline. Instead of loading static image files, it calculates colors mathematically in real-time. The system is composed of three primary pillars:

1. **Pattern Configurations:** A centralized configuration file that stores the mathematical data of each pattern. It defines specific parameters such as frequency, amplitude, and scale.
2. **Pattern Generator Class:** The "engine" of the component. This class contains the core logic that evaluates coordinates (x, y) and time (t) through mathematical functions. It transforms these raw inputs into dynamic, high-fidelity color states on the fly.
3. **Pattern Selection Component:** The user-facing interface that allows for seamless switching between different algorithms at runtime. This allows users to instantly observe how different mathematical models affect the simulation's visual output.

By calculating colors "on the fly" according to the user's runtime choices, the system provides key advantages such as **Memory Efficiency**, **Resolution Independence**, and **Temporal Smoothness** so the patterns exhibit fluid, organic animations that never repeat or "loop" in a noticeable way.



Figure 6.16: Generated example Star Pattern using the mathematical function at time ‘t’

6.5.4 Synchronizer

Acts as the critical link between the human’s position data and the final visual output. It serves as a "translator" that converts spatial data (where a human is) into aesthetic data (what color that human should be).

The primary workhorse of this component is the `update()` method. Its job is to take a snapshot of the current states and map them to the procedural pattern. The Synchronizer queries the `PatternGenerator` for what color each human should be at that position and time. It packages these results into a `HumanColorState`, creating a ready-to-use list for the renderer. The Synchronizer is modular; it only cares about the relationship between position and color. It uses almost no extra memory by "pointing" to existing pattern logic. Because it returns a `const&` (constant reference), it guarantees that the visual data cannot be accidentally altered, ensuring "**data integrity**" across the pipeline.

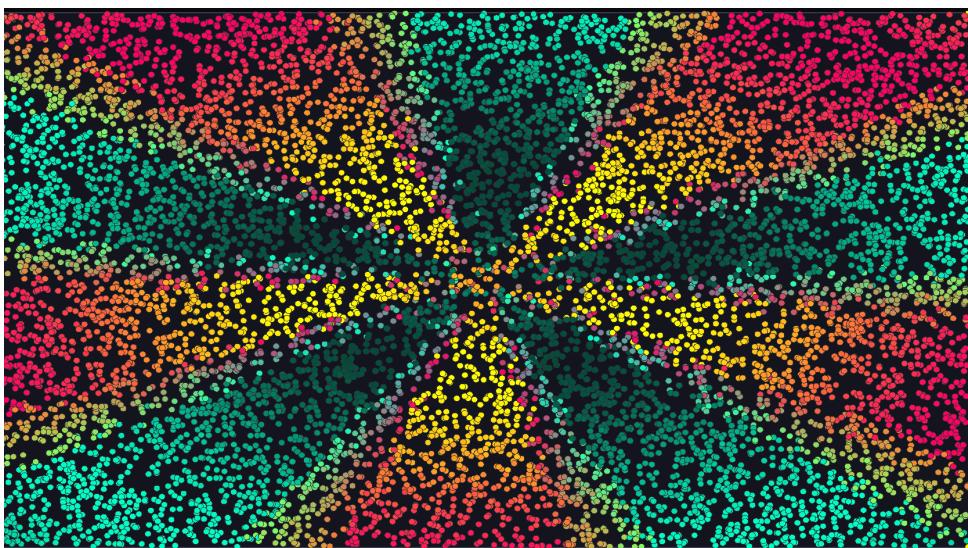


Figure 6.17: Synchronizer output for the Star pattern snapshot at time ‘t’

6.5.5 Visualizer

This is a high-performance rendering engine built on **SFML (Simple and Fast Multimedia Library)** that transforms abstract data positions into a fluid crowd simulation display. To handle thousands of humans at high frame rates, the Visualizer uses a sophisticated **Two-Phase approach** to solve the bottleneck of preparing and drawing data sequentially on a single thread.

- **Phase 1:** The SFML engine receives high-precision coordinates. Using **OpenMP**, the workload is parallelized across all available CPU cores. Each core processes a distinct subset of humans, performing coordinate transformation and styling pre-allocated shapes. Since each core operates on a unique `humanShape` index, there is no memory contention.
- **Phase 2:** Sequential Drawing of the "Final Output". Since SFML's `window.draw()` command is not thread-safe, this phase runs on a single thread, pushing the already-prepared shapes to the screen.

Instead of creating and destroying objects every frame, the Visualizer maintains a `vector` of `CircleShape` in memory, updating properties as needed for maximum efficiency.

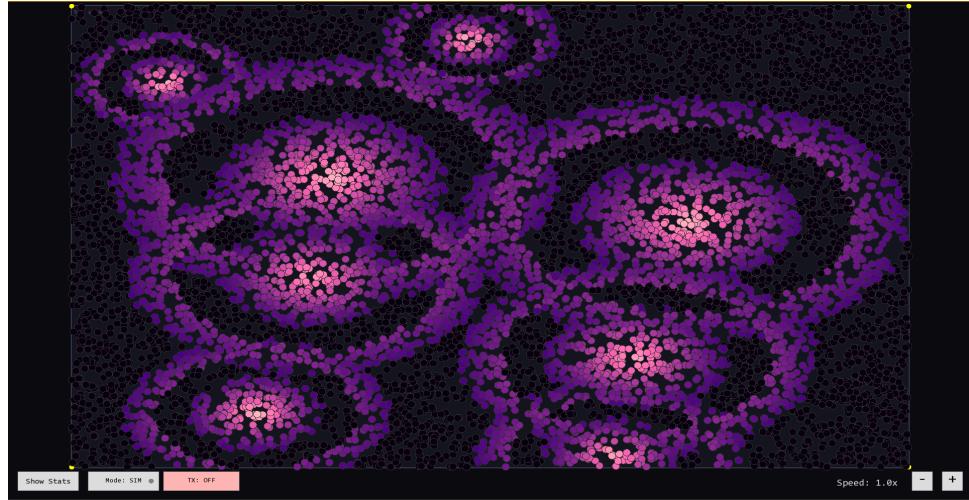


Figure 6.18: Snapshot of the visualizer with 10,000 human positions

6.5.6 Database & Pattern Player

The Recording & Playback System uses a **Hybrid Storage Architecture**. Instead of a single database, it splits the data to handle 10,000+ humans efficiently.

1. **SQLite for organization:** Acts as the Librarian. It stores "metadata" like simulation date and an index of every frame, knowing exactly where each frame starts in the binary file.

<u>frame_id</u>	<u>run_id</u>	timestamp_ms	human_gen_delay_ms	sync_delay_ms	total_humans	bin_offset
Filter	Filter	Filter	Filter	Filter	Filter	Filter
159	2	6257.5741	0.3458	3.5946	9000	0
160	2	6296.0519	0.4133	3.5946	9000	135000
161	2	6334.3366	0.4039	3.5946	9000	270000
162	2	6372.7136	0.3891	3.5946	9000	405000
163	2	6410.3001	0.3008	3.6505	9000	540000
164	2	6451.5836	0.2819	3.6505	9000	675000
165	2	6488.9164	0.3357	3.6505	9000	810000
166	2	6525.7607	0.2771	3.6505	9000	945000
167	2	6563.3291	0.3786	3.6505	9000	1080000
168	2	6600.4849	0.4162	3.6505	9000	1215000

Figure 6.19: SQLite database snapshot used to store offsets of the binary file and metadata

2. **Binary Files for raw speed:** Acts as the Warehouse. It stores the raw "Human State" (position and color) in a compressed, 15-byte packet.

To record 10,000 humans at 30 FPS, the required memory write rate is:

$$10,000 \text{ humans} \times 15 \text{ bytes} \times 30 \text{ FPS} \approx 4.5 \text{ MB/s}$$

For a standard SSD (approx. 500 MB/s), this uses only about 1% of the write bandwidth.

Also, to ensure the simulation stays at a smooth 30 FPS, the system never waits for the hard drive to finish writing. The system uses a **Background Worker Thread** (Producer-Consumer model).

1. The Main Thread (Producer) packages data into a **Thread-Safe Queue**
2. The Worker Thread (Consumer) writes to disk without blocking the simulation.

The playback engine is built for instant navigation. Because every frame has a fixed size and its location is indexed in SQLite, the system can perform **O(1) Random Access** (No Scanning). If the user moves the playback slider to the middle of a 2-hour recording, the system doesn't "fast-forward" through the file. It queries SQLite for the bin offset, "jumps" directly to that byte in the binary file, and reads the frame instantly.

6.6 RGB LED Data Reception and Control

After processing in the synchronizer, the corresponding tag ID and color information are transmitted to the Gateway, which forwards the data to the wearable tags via the 2.4GHz transmitter. This unit is responsible for receiving the color data from the Gateway and driving the lighting system on each tag.

6.6.1 Wireless Color Data Transmission Architecture

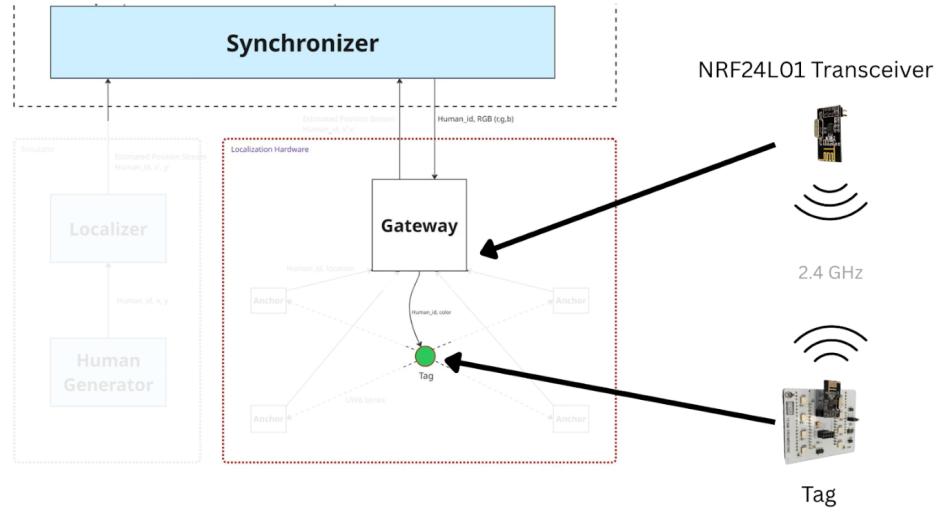


Figure 6.20: Transmission between Gateway and Tag

Gateway-to-Tag Communication Protocol

Wireless reception and transmission are handled by a **2.4 GHz NRF24L01** module. Addressable LEDs **WS2812B** are used for emitting the light. The received packet contains the **Tag_ID** and the assigned color value. Upon successful reception and validation of the **Tag_ID**, the color data is applied to the LEDs integrated on the tag, enabling real-time visual identification.

NRF24L01 Configuration and RF Channel Allocation

The wireless communication between the Gateway and the wearable tags is implemented using the **NRF24L01 2.4 GHz transceiver module**. This module operates in the **2.4–2.5 GHz ISM band** and supports configurable data rates of **250 kbps, 1 Mbps, and 2 Mbps**, enabling low-latency and reliable communication.

Each NRF24L01 module supports **125 selectable RF channels**, allowing multiple independent wireless networks to coexist in the same environment. Additionally, each channel supports up to **six logical addresses (pipes)**, enabling communication with multiple devices simultaneously. In this system, a unique **Tag_ID** is used to ensure that only the intended tag responds to the received color data.

Packet Structure and Tag Identification Mechanism

The NRF24L01 communicates with the controller using the **SPI protocol**, while the CE and CSN pins are used to control transmission and reception modes. Payloads of up to **32 bytes** can be transmitted in a single packet, which is sufficient for sending compact data such as **Tag_ID** and RGB color values.

Power Optimization and Transmission Range Analysis

To optimize power consumption and communication reliability, the **Power Amplifier (PA)** level is configured based on the required operating range. Lower PA levels are used during short-range operation to reduce power usage, while higher levels can be enabled for extended

range when necessary. In open-space conditions, the communication range can reach up to **25 m**, and even higher when PA/LNA variants with external antennas are used.

The NRF24L01 operates at **3.3 V**, with a typical transmission current of approximately **12 mA**. Due to its sensitivity to power supply noise, **local decoupling capacitors (10 μ F–100 μ F)** are placed close to the module's VCC and GND pins to ensure stable operation. In the final tag design, power is derived from a **7.4 V Li-ion battery pack** and regulated on-board to meet the module's voltage requirements.

Upon receiving a valid packet containing the matching Tag_ID and color information, the tag controller decodes the data and updates the **LED color output in real time**, providing immediate visual feedback for identification and tracking purposes.

Multi-Channel Round Robin Transmission Strategy

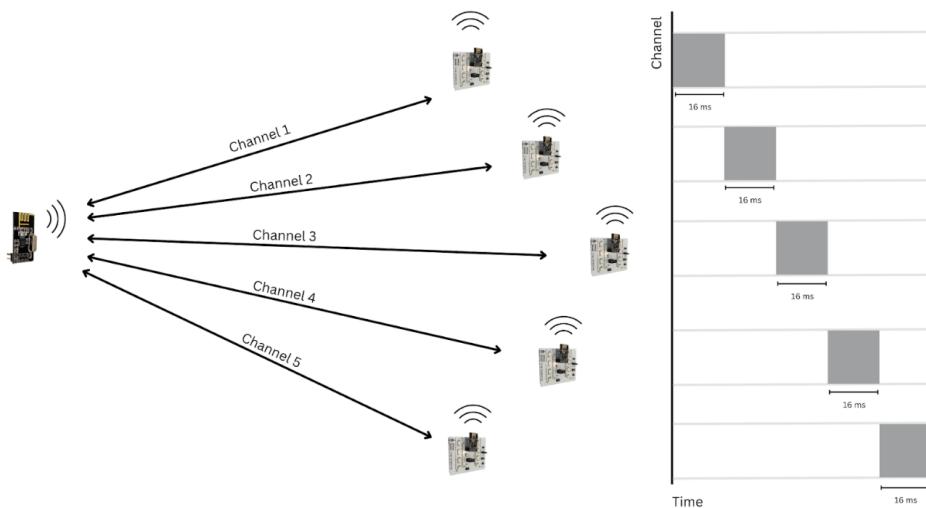


Figure 6.21: Color Data transmission channels

Transmission of color data is transmitted via 5 channels. They were sent in a round robin manner to the tags where each transmission took 16ms for transmission. So the entire cycle takes 80ms to transmit.

6.6.2 Receiver Circuit

Receiver Block Diagram Overview

The receiver circuit consists of 4 key blocks:

- Microcontroller
- NRF24L01 Transceiver
- Addressable LEDs
- Power regulation

The tag receiver unit is designed using the following block diagram.

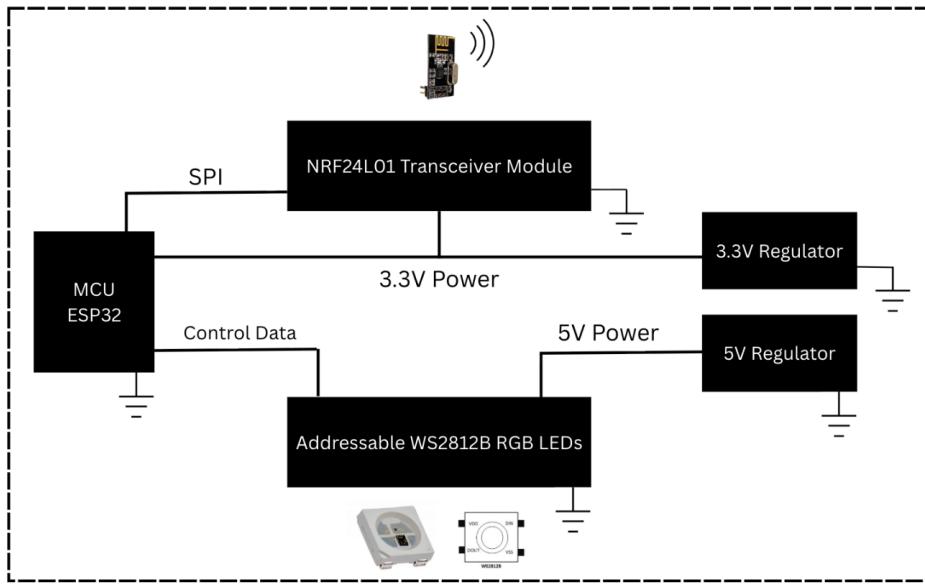


Figure 6.22: Block Diagram of Tag receiver circuit

Power Regulation

As the microcontroller, we used an **ESP32**. Both MCU and Transceiver modules are operating at 3.3V and LEDs are powering at 5V. So we used 2 voltage regulators to regulate the voltage levels, and coupling capacitors are used to stabilise the voltage.

PCB Design, Fabrication, and Hardware Implementation

The schematic diagram and PCB layout of the tag receiver module are shown below.

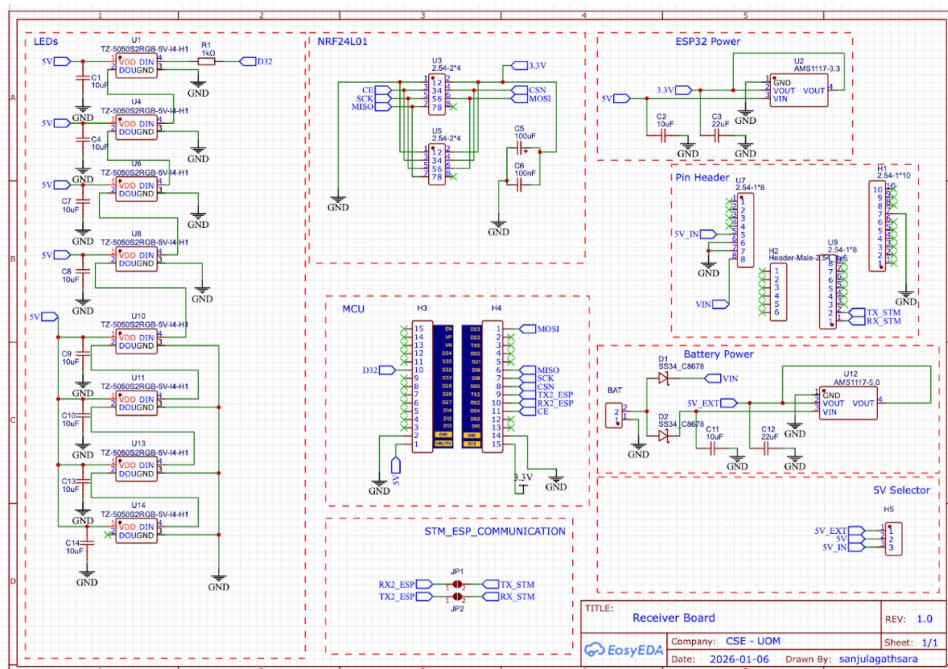


Figure 6.23: TDOA Schematic Diagram of the tag receiver circuit

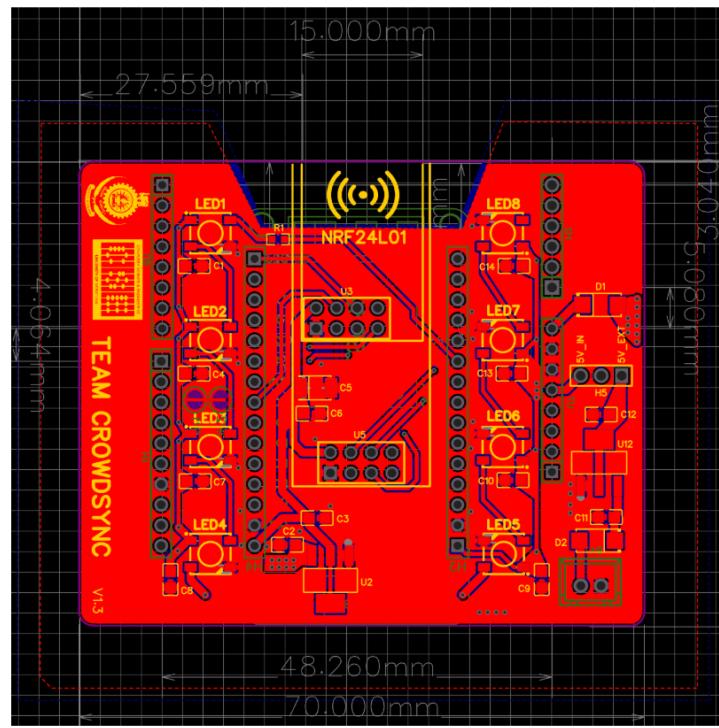


Figure 6.24: Layout diagram of tag receiver circuit

Power is supplied by an **external 7.4 V Li-ion battery pack**, which is regulated on-board to meet the operating voltage requirements of the NRF24L01 module and the LED circuitry. After fabrication, the finalized PCB appears as shown below.

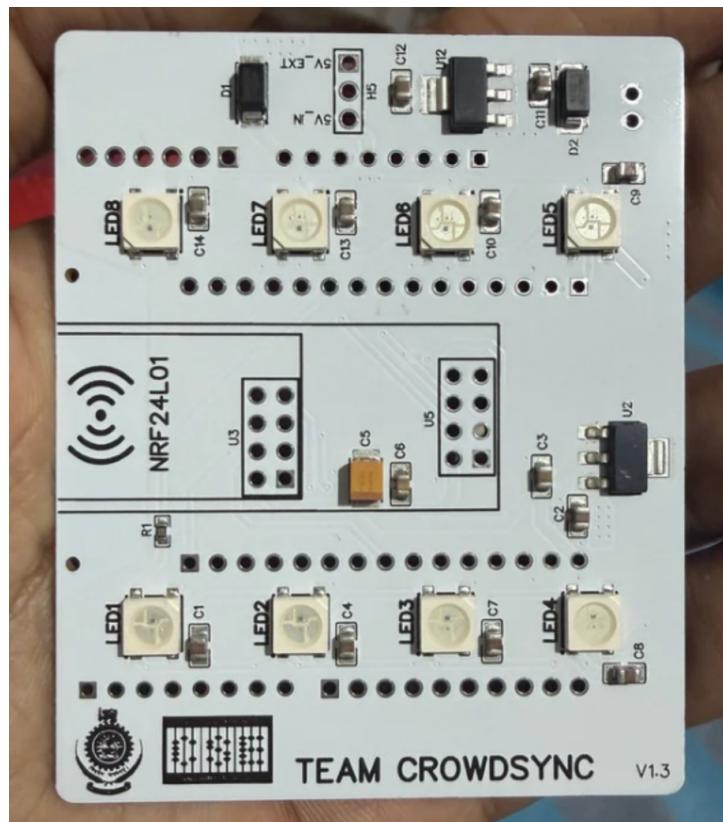


Figure 6.25: TDOA Manufactured Tag receiver Printed Circuit Board

A total of **five tag receiver boards** were fabricated for testing and validation purposes. In the completed tag assembly, this receiver PCB is **stacked with the UWB module and STM32**, enabling both **localization and color reception** functionalities within a single wearable unit.

6.6.3 Controlling LEDs

WS2812B LED Architecture and Daisy-Chain Configuration

The LEDs used are RGB WS2812B LEDs where they are connected in a **daisy chain** method. The DOUTs of LEDs are connected with DINs.

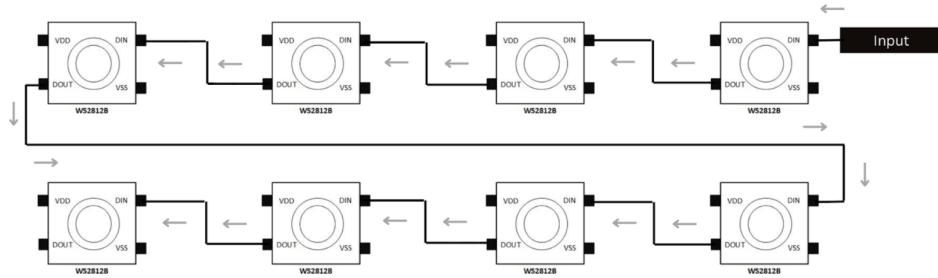


Figure 6.26: LEDs Connected as a Daisy Chain

Timing Requirements for Data Transmission

In order to transmit a bit we need to follow the following steps:

- **Transmitting a 1:**
 - Time for the signal to remain high ($T1H$): $0.8 \mu s$
 - Time for the signal to remain low ($T1L$): $0.45 \mu s$
- **Transmitting a 0:**
 - Time for the signal to remain high ($T0H$): $0.4 \mu s$
 - Time for the signal to remain low ($T0L$): $0.85 \mu s$
- **Latching the data:** After sending all the bits corresponding to the intensity values of all the LEDs that we want to control, we need to simply hold the value of the pulse wave at its minimum value for at least $50 \mu s$.

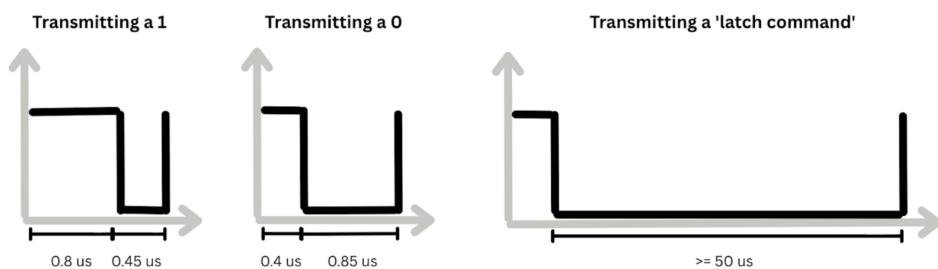


Figure 6.27: LEDs Control Timing Diagrams

Bit Encoding and Color Data Formatting (24-bit GRB Protocol)

The signal needed for 1 LED is 24 bits, where 8 bits each are allocated for color Green, Red, and Blue. So for controlling 8 LEDs we need to send $24 \times 8 = 192$ bits in sequence and Save by Latch command. Figure 6.28 shows the bit sequence diagram for controlling 8 LEDs.

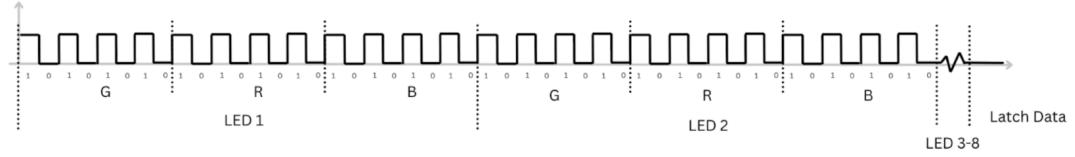


Figure 6.28: LEDs Bit Sequence Diagram

Conclusion

The color data coming from the synchronizer is successfully transmitted to tags and lights up the color successfully.

7. Research Timeline

7.1 Project Milestones

- **Sep 15, 2025** – Proposal Completion
- **Sep 18, 2025** – Proposal Presentation
- **Dec 25, 2025** – Mapping Algorithm & Control Message Generator due
- **Dec 31, 2025** – Scalable Transmission Framework due
- **Jan 4, 2026** – Scalable UWB-based Localization Subsystem due
- **Jan 31, 2026** – Novel Approach Finalized
- **Feb 16, 2026** – Initial Draft Completed
- **Mar 20, 2026** – Final Documentation and Presentation

7.2 Gantt Chart

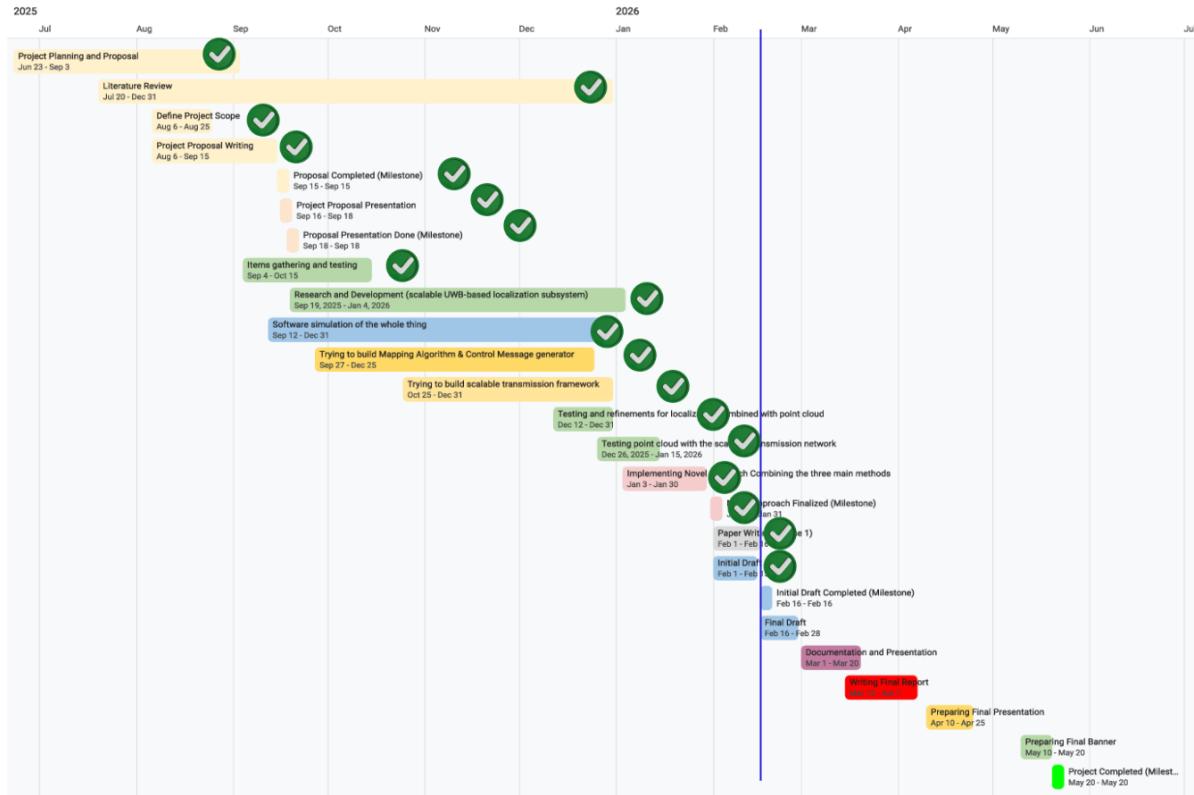


Figure 7.1

References

- [1] (2025) Coldplay - a sky full of stars (live at river plate). Accessed: 2025-08-29. [Online]. Available: <https://www.youtube.com/watch?v=Fpn1imb9qZg>
- [2] (2025) Pixmob. PixMob. Accessed: 2025-08-29. [Online]. Available: <https://pixmob.com/>
- [3] (2025) CrowdSync. CrowdSync. Accessed: 2025-08-29. [Online]. Available: <https://www.crowdsynctechnology.com/>
- [4] (2025) Xylobands usa. Xylobands. Accessed: 2025-09-13. [Online]. Available: <https://xylobandsusa.com/>
- [5] Hackaday. (2023, Jul) Exploring the tech behind concert led wristbands. Accessed: 2025-09-13. [Online]. Available: <https://hackaday.com/2023/07/01/exploring-the-tech-behind-concert-led-wristbands/>
- [6] (2025) Pixmob mvt. PixMob. Accessed: 2025-09-13. [Online]. Available: <https://pixmob.com/mvt>
- [7] W. S. Journal. (2023) How concert led wristbands work | wsj tech behind. Accessed: 2025-09-13. [Online]. Available: <https://www.youtube.com/watch?v=15>
- [8] T. Aziz and I. Koo, “A comprehensive review of indoor localization techniques and applications in various sectors,” vol. 15, no. 3, p. 1544, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2076-3417/15/3/1544>
- [9] M. Ridolfi, S. Van De Velde, H. Steendam, and E. De Poorter, “Analysis of the scalability of UWB indoor localization solutions for high user densities,” vol. 18, no. 6, p. 1875. [Online]. Available: <https://www.mdpi.com/1424-8220/18/6/1875>
- [10] A. De-La-Llana-Calvo, J.-L. Lázaro-Galilea, A. Gardel-Vicente, D. Rodríguez-Navarro, I. Bravo-Muñoz, and F. Espinosa-Zapata, “Characterization of multipath effects in indoor positioning systems by aoa and poa based on optical signals,” *Sensors*, vol. 19, no. 4, p. 917, 2019.
- [11] FiRa Consortium, “Impact of imt on uwb: Fira™ imt impact study white paper,” FiRa Consortium, White Paper, July 2025, monte Carlo simulations assessing interference of IMT systems (outdoor macro urban and indoor small-cell) on UWB deployments. [Online]. Available: <https://www.firaconsortium.org/sites/default/files/2025-07/FiRa-IMT-impact-Study-White-Paper.pdf>
- [12] F. Zafari, A. Gkelias, and K. Leung, “A survey of indoor localization systems and technologies.” [Online]. Available: <http://arxiv.org/abs/1709.01015>
- [13] I. Oppermann, Ed., *UWB Theory and Applications*, reprinted ed. Chichester: Wiley, 2006.

- [14] L. Tian, S. Santi, A. Seferagić, J. Lan, and J. Famaey, “Wi-fi HaLow for the internet of things: An up-to-date survey on IEEE 802.11ah research,” vol. 182, p. 103036. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S108480452100062X>
- [15] J. Hopper. Wi-fi HaLow. [Online]. Available: <https://nexcommsys.com/wi-fi-halow-long-range-wi-fi>
- [16] I.-G. Lee, D. B. Kim, J. Choi, H. Park, S.-K. Lee, J. Cho, and H. Yu, “WiFi HaLow for long-range and low-power internet of things: System on chip development and performance evaluation,” vol. 59, no. 7, pp. 101–107. [Online]. Available: <https://ieeexplore.ieee.org/document/9502660/>
- [17] M. Nikodem and P. Szeliński, “Channel diversity for indoor localization using bluetooth low energy and extended advertisements,” vol. 9, pp. 169 261–169 269. [Online]. Available: <https://ieeexplore.ieee.org/document/9661373>
- [18] “Direction-of-arrival based indoor localization systems for massive IoT networks.”
- [19] M. Jurgens, D. Meis, D. Mollers, F. Nolte, E. Stork, G. Vossen, C. Werner, and H. Winkelmann, “Bluetooth mesh networks for indoor localization,” in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, pp. 397–402. [Online]. Available: <https://ieeexplore.ieee.org/document/8788826/>
- [20] C.-C. Chong, F. Watanabe, and H. Inamura, “Potential of UWB technology for the next generation wireless communications.”
- [21] D. Coppens, A. Shahid, S. Lemey, B. Van Herbruggen, C. Marshall, and E. De Poorter, “An overview of UWB standards and organizations (IEEE 802.15.4, FiRa, apple): Interoperability aspects and future research directions,” vol. 10, pp. 70 219–70 241. [Online]. Available: <https://ieeexplore.ieee.org/document/9810941>
- [22] (2018) openrtls. [Online]. Available: <https://openrtls.com/Archive/index.html>
- [23] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter, “Analysis of the scalability of UWB indoor localization solutions for high user densities,” vol. 18, no. 6, p. 1875, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1424-8220/18/6/1875>
- [24] B. Großwindhager, M. Stocker, M. Rath, C. A. Boano, and K. Römer, “SnapLoc: an ultra-fast UWB-based indoor localization system for an unlimited number of tags,” in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. ACM, pp. 61–72. [Online]. Available: <https://dl.acm.org/doi/10.1145/3302506.3310389>
- [25] “(PDF) MULoc: Towards millimeter-accurate localization for unlimited UWB tags via anchor overhearing,” in *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/387772183_MULoc_Towards_Millimeter-Accurate_Localization_for_Unlimited_UWB_Tags_via_Anchor_Overhearing
- [26] (2024) Muloc github repository. [Online]. Available: <https://github.com/MULoc>
- [27] S. Krebs and T. Herter, “Ultra-wideband positioning system based on esp32 and dwm3000 modules,” *arXiv*, Mar 2024. [Online]. Available: <https://doi.org/10.48550/arxiv.2403.10194>
- [28] A. Censi, J. Strubel, C. Brandli, T. Delbrück, and D. Scaramuzza, “Low-latency localization by active LED markers tracking using a dynamic vision sensor,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 891–898. [Online]. Available: <https://ieeexplore.ieee.org/document/6696456/>
- [29] Qorvo. (2025) About us - qorvo. [Online]. Available: <https://www.qorvo.com/about-us>

- [30] Everything RF. (2020, Dec.) Qorvo's full uwb technology product portfolio is now available from mouser. [Online]. Available: <https://www.everythingrf.com/News/details/11459-qorvo-s-full-uwb-technology-product-portfolio-is-now-available-from-mouser>
- [31] NXP Semiconductors. (2020) Trimension® ultra-wideband (uwb). [Online]. Available: <https://www.nxp.com/products/wireless-connectivity/trimension-uwb:UWB-TRIMENSION>
- [32] FiRa Consortium. (2025) Fira consortium | fira consortium. [Online]. Available: <https://www.firaconsortium.org/about/consortium>
- [33] (2025) Qorvo dw1000 product page. Qorvo, Inc. Accessed: 2025-08-29. [Online]. Available: <https://www.qorvo.com/products/d/da007946>
- [34] Makerfabs. (2025) Esp32 uwb (ultra wideband). [Online]. Available: <https://www.makerfabs.com/esp32-uwb-ultra-wideband.html>
- [35] (2025) Qorvo dw3000 datasheet. Qorvo, Inc. Accessed: 2025-08-29. [Online]. Available: https://www.mouser.com/datasheet/2/412/Qorvo_DW3000-2934245.pdf
- [36] T. Polonelli, S. Schlapfer, and M. Magno, "Performance comparison between decawave DW1000 and DW3000 in low-power double side ranging applications," in *2022 IEEE Sensors Applications Symposium (SAS)*. IEEE, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9881375/>
- [37] M. Gately, Y. Zhai, M. Yeary, E. Petrich, and L. Sawalha, "A three-dimensional swept volume display based on led arrays," *Journal of Display Technology*, vol. 7, no. 10, pp. 503–514, 2011.
- [38] C.-W. Hung, W.-T. Hsu, and K.-H. Hsia, "Multiple frequency shift keying optimization of adaptive data rate for ultra-low power wireless sensor network," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, Oct 2019.
- [39] S. Devalal and A. Karthikeyan, "Lora technology – an overview," *IEEE Xplore*, Mar 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8474715>
- [40] R. Kshetrimayum, "An introduction to uwb communication systems," *IEEE Potentials*, vol. 28, no. 2, pp. 9–13, Mar 2009.
- [41] G. Koulouras, S. Katsoulis, and F. Zantalis, "Evolution of bluetooth technology: Ble in the iot ecosystem," *Sensors*, vol. 25, no. 4, p. 996, Feb 2025.
- [42] S. Kumar, G. Moloudian, D. R. Gawade, B. O'Flynn, and J. L. Buckley, "Sub-ghz wrist-worn antennas for wireless sensing applications: A review," *IEEE Open Journal of Antennas and Propagation*, vol. 5, no. 5, pp. 1258–1281, May 2024.
- [43] Texas Instruments, "Cc1310 data sheet, product information and support," <https://www.ti.com/product/CC1310>, 2015, accessed: 2025-09-17.
- [44] W. Zhao, A. Goudar, M. Tang, and A. P. Schoellig, "Ultra-wideband time difference of arrival indoor localization: From sensor placement to system evaluation," *arXiv preprint arXiv:2412.12427*, Dec. 2024. [Online]. Available: <https://arxiv.org/abs/2412.12427>
- [45] J. Nagdeo, "Synchronization of anchors in a tdoa based ultra-wide band localization system," n.d., unpublished manuscript.
- [46] nsnam.org. (2025) ns-3: a discrete-event network simulator for internet systems. Accessed: 2025-09-17. [Online]. Available: <https://www.nsnam.org/>