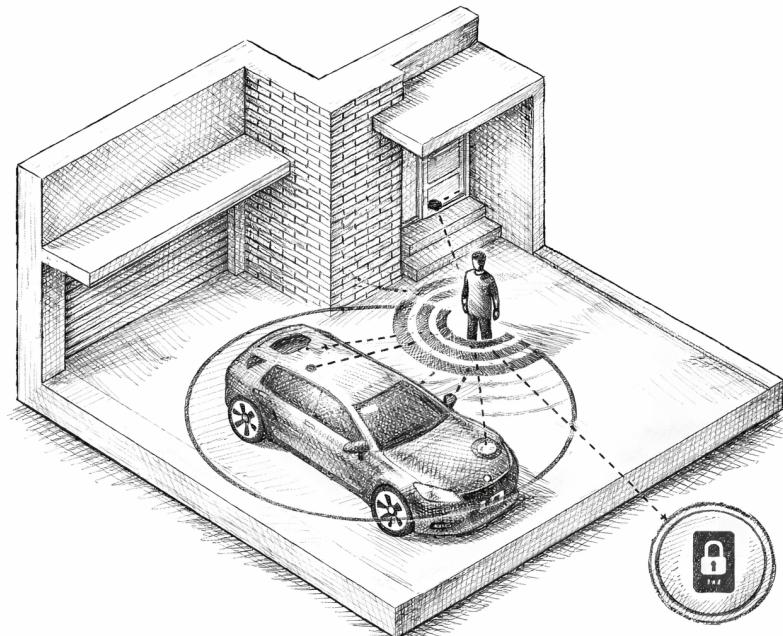


Keyless Entry in Automotive

A viable application solution

Amine Benchekroun
Badr Diani
Fatine Azzabi
João G. Buttow A.



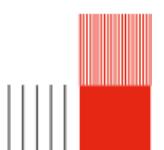
Institut National des Sciences Appliquées de Toulouse

January 28, 2026



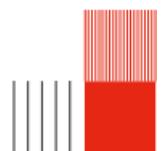
Abstract

Ultra-Wideband (UWB) has emerged as a key enabler for secure keyless vehicle access due to its ability to verify proximity with high spatial resolution. This work presents an end-to-end prototype for a BLE/UWB-based access system and focuses on three practical aspects required for real deployments: improving positioning accuracy from raw ranging outputs, enabling cloud-based logging for traceability, and quantifying the energy impact of an always-available in-vehicle infrastructure. Starting from a reference UWB ranging implementation, we enhance the localization pipeline by formulating position estimation as a nonlinear least-squares problem and by introducing robustness mechanisms suitable for noisy and imperfect measurements. The solver is warm-started with the previous state to exploit temporal continuity, optional weighted residuals reduce the influence of less reliable ranges, and invalid or incomplete samples are safely handled. A Kalman filter is then coupled to the optimization outputs to obtain smoother and physically plausible trajectories, especially in challenging phases such as approaching and entering the vehicle. To support monitoring and experiment reproducibility, we implement a secure cloud backend using AWS IoT Core (MQTT over TLS with X.509 authentication) and persist measurements in DynamoDB through IoT Rules, creating an auditable history of distance logs that can later be connected to live embedded streaming. Finally, energy consumption is assessed through oscilloscope-based current profiling with a shunt resistor and a system-level parked-autonomy model. Measurements show a clear increase in average current when UWB ranging is enabled, and the derived autonomy bounds emphasize that power-supply strategy and duty-cycling decisions are critical to balance user experience with parked-vehicle constraints.



Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 2 |
| 2 | System Overview | 3 |
| 3 | Presentation of the Partners | 5 |
| 4 | Background and Motivation: Why UWB for Keyless Car Access | 6 |
| 4.1 | UWB (Ultra Wideband) technology | 6 |
| 4.2 | UWB for Intelligent Vehicle Access | 7 |
| 4.3 | Key Advantages of UWB in Vehicle Access | 7 |
| 4.4 | Objectives of UWB-Enhanced Vehicle Systems | 7 |
| 5 | The project and its main objectives | 9 |
| 6 | Project Organization | 10 |
| 7 | Objective 01 - Improving Positioning Accuracy | 12 |
| 7.1 | Vehicle Configuration and Evaluated Scenario | 12 |
| 7.2 | Initial Positioning Approach | 13 |
| 7.3 | Proposed Solution | 14 |
| 7.3.1 | Nonlinear Least-Squares Position Estimation | 14 |
| 7.3.2 | Warm-Start from the Previous State | 14 |
| 7.3.3 | Weighted Residuals for Measurement Robustness | 15 |
| 7.3.4 | Solver Choice: Least-Squares Dedicated Optimisation | 15 |
| 7.3.5 | Handling Invalid and Incomplete Measurements | 15 |
| 7.3.6 | Tighter Coupling Between Optimisation and Kalman Filtering | 15 |
| 7.4 | Comparison of Results | 16 |
| 8 | Objective 02 - Cloud | 18 |
| 8.1 | Cloud provider selection | 18 |
| 8.2 | Cloud implementation | 19 |
| 9 | Objective 03 - Energy | 21 |
| 9.1 | Current measurement method (shunt resistor) | 21 |
| 9.2 | Results 1 – Initiator | 22 |
| 9.3 | Results 2 – Receptors (responders) | 22 |
| 9.4 | System-level power model and parked autonomy | 23 |
| 9.4.1 | Per-node current model | 23 |
| 9.4.2 | Vehicle-side power and DC–DC conversion | 23 |
| 9.4.3 | Starter-battery autonomy (parked vehicle) | 24 |
| 9.5 | Mitigation strategies for long parking | 24 |
| 9.5.1 | Dedicated auxiliary battery (LiFePO ₄ 12.8 V – 30 Ah example) | 24 |
| 9.5.2 | Ambient energy harvesting (rooftop photovoltaics) | 25 |
| 10 | Objective 04 – Real-Time Position Computation, Cloud Streaming, and Live Visualisation | 26 |
| 11 | Objective 05 - BLE/UWB Integration | 28 |
| 12 | Conclusion | 29 |



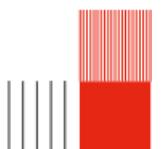
Chapter 1

Introduction

Keyless access systems are a strong illustration of how embedded sensing, secure wireless communication, and software robustness must converge to produce a reliable user experience. In an automotive context, the challenge is not limited to detecting a device nearby: the system must make correct proximity decisions under highly variable radio conditions, provide traceable evidence of its behaviour during tests, and remain compatible with strict parked-vehicle energy constraints. In practice, a prototype that “works once” is not enough; the system must behave consistently across trajectories, environments, and measurement imperfections, while remaining explainable and measurable.

This project builds on an existing BLE/UWB experimental baseline and focuses on turning it into an end-to-end, testable pipeline that can be evaluated under vehicle-level conditions. The work is intentionally engineering-oriented: we address concrete weaknesses observed in real measurements, rather than proposing purely theoretical improvements. First, we improve localisation robustness by reformulating position estimation and by introducing mechanisms that reduce sensitivity to initialization and measurement outliers. Second, we connect the experiment loop to a secure cloud backend in order to log and persist measurements for monitoring and traceability. Third, we quantify the energy impact of the in-vehicle infrastructure and derive order-of-magnitude parked autonomy bounds, which are essential to assess feasibility in real deployment scenarios. Finally, we operationalize the pipeline by computing the user state in real time and making it immediately interpretable through live visualisation, while preparing the BLE/UWB integration workflow for the next project iteration.

The remainder of this report is structured as follows. Chapter 2 introduces the system at a high level and the considered BLE-to-UWB workflow. Chapter 3 then presents the project context and partners, followed by the motivation for UWB-based access in automotive scenarios. Chapter 7 details the localisation improvements and compares results on representative datasets. Chapter 8 presents the secure cloud pipeline and its validation. Chapter 9 provides the current measurements, the system-level power model, and mitigation options for long parking periods. Chapter 11 discusses the BLE/UWB integration constraints and the delivered integration guide. The report concludes with a summary of contributions and a short outlook on the next technical steps.



Chapter 2

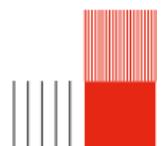
System Overview

Modern vehicles are increasingly moving toward digital key systems that allow drivers to use smartphones in place of traditional physical keys. A digital key solution enables an authorized user to be identified and to seamlessly lock, unlock, and start the car as long as the user's device (e.g. a mobile phone) is in close proximity. To support widespread adoption of such technology, the Car Connectivity Consortium (CCC) has standardized the digital key function to ensure interoperability across different automakers and mobile devices. In this context, and in collaboration with ACTIA, NXP Semiconductors, INSA Toulouse, and Linköping University, this project aims to design and implement a secure proximity-based car unlocking system using a smartphone as a digital key.

At the core of the proposed system is Ultra-Wideband (UWB) wireless technology, which provides highly precise distance measurement and real-time localization of the user in relation to the vehicle. UWB ranging allows the vehicle to determine the smartphone's distance with unprecedented accuracy – on the order of a few centimeters – by measuring the time-of-flight of radio signals. This fine-grained precision is crucial for both usability and security. Unlike conventional passive keyless entry systems that are vulnerable to relay attacks (where an attacker relays signals to spoof proximity), a UWB-based digital key can verify that the authorized device is truly nearby, effectively thwarting relay-based car theft attempts. In practice, the car remains locked until the verified user comes sufficiently close with their device, and it can automatically lock again when the user walks away, ensuring convenience without sacrificing safety.

To maximize reliability and user experience, the system employs a multi-technology approach combining Bluetooth Low Energy (BLE) and UWB. Initially, BLE is used for device discovery and coarse-range communication when the user is in the general vicinity of the car. Once the user's smartphone is detected and authenticated via BLE, a handover to UWB is performed for fine-range positioning and secure verification as the user approaches the vehicle. By using BLE for the initial wake-up and authentication and then UWB for precise distance measurement, the solution ensures robust operation across varying ranges while minimizing false unlock triggers. The hardware implementation leverages NXP's UWB components: UWB anchor modules are installed on the vehicle, and the user's mobile device is equipped with a UWB transceiver. This setup enables reliable real-time communication between the car and the smartphone, allowing the system to continuously measure distance and orientation with high accuracy.

As illustrated in Figure 2.1, the interaction between BLE and UWB is organized as a progressive range-based workflow. When the user is outside the system range, the BLE module remains in an idle scanning mode to detect known devices. Once the user enters the BLE coverage area, identification and access verification are performed, while coarse distance estimation can be initiated. Finally, when the user reaches the UWB coverage area, the system performs a handover from BLE to UWB, enabling fine-grained ranging and precise positioning close to the vehicle.



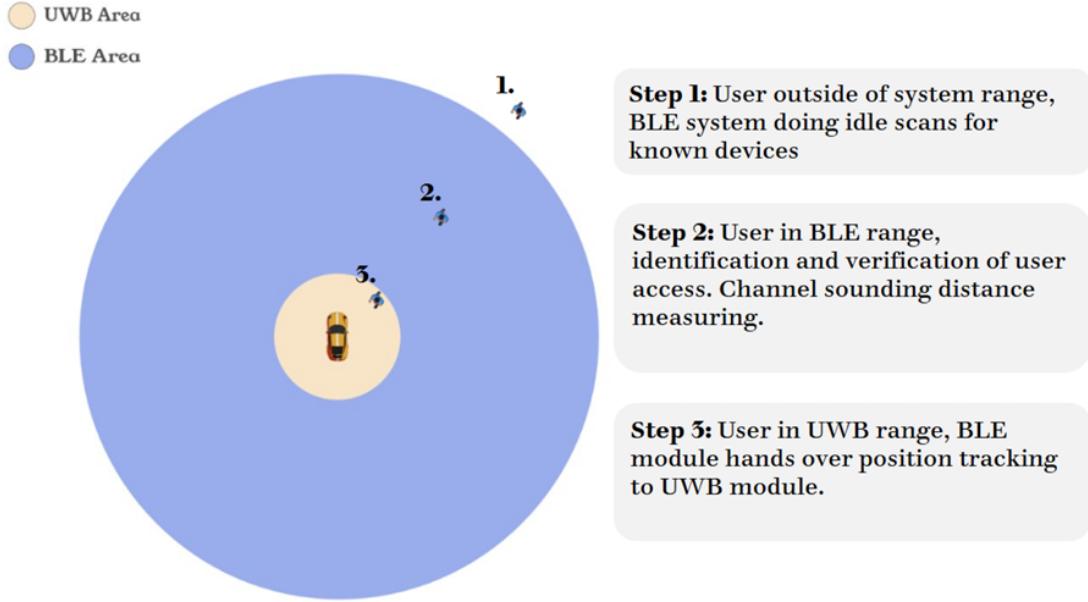
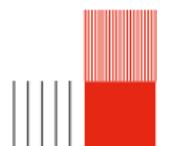


Figure 2.1: BLE-to-UWB operational workflow: BLE is used for discovery and authentication at longer range, then a handover to UWB is performed for accurate ranging and positioning close to the vehicle.

Furthermore, the project integrates a cloud-based backend for data monitoring and system management. Real-time distance measurements and status information are published from the vehicle to the cloud using the MQTT protocol via AWS IoT Core, and the data is stored in the AWS cloud (DynamoDB) for live monitoring and analysis. This architecture allows stakeholders to remotely observe the system's performance and verify the user's proximity data during experiments or real-world trials. The cloud infrastructure is also designed with security and data protection in mind, ensuring that all sensitive information (such as location data and digital key credentials) is handled in compliance with relevant regulations. In addition, this cloud-based approach supports advanced capabilities such as secure device tracking and controlled experimentation with different system configurations.

The motivation behind this work is grounded in the broader trend of replacing physical keys in vehicle fleets and shared mobility services with digital keys. Eliminating physical keys allows fleet operators and car-sharing services to manage vehicle access remotely and flexibly – improving operational efficiency and reducing the risk of lost or stolen keys. More importantly, a UWB-backed digital key provides enhanced protection against modern theft techniques: it neutralizes relay attacks by requiring the legitimate user's device to be physically near the car, and this shift to a “key in the cloud” paradigm can offer a level of vehicle security never before possible with traditional keys. Finally, the emphasis on robust, real-time positioning in this project is critical, as reliable instant user localization is essential for a smooth user experience and for safety in automatic locking/unlocking scenarios. The outcome is a cutting-edge car access system that combines convenience with high security, illustrating the potential of UWB and BLE technologies to transform how vehicles are accessed and shared.



Chapter 3

Presentation of the Partners

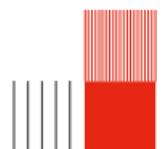
This project was carried out in collaboration with key industry partners, ACTIA and NXP Semiconductors, as well as with a team of students from Linköping University in Sweden, which we present below.

ACTIA, one of our key partners, is a global leader in automotive electronics, specializing in vehicle connectivity and embedded systems. They provided critical guidance on integrating Ultra-Wideband (UWB) technology into automotive environments. ACTIA ensured that our work aligned with industry standards and facilitated access to real-world scenarios for testing and validation. The commitment and support of Pierre Gruyer, Innovation Program Manager, played a vital role in helping us achieve the project's objectives.

NXP, another key partner in this project, is a leading semiconductor manufacturer renowned for its expertise in wireless communication and embedded systems. NXP provided us with state-of-the-art UWB development boards, which formed the foundation of our experimental work and prototype implementation. When needed, technical support from NXP was provided through a designated contact: Youssef Houiti, System Application Engineer.

In addition to ACTIA and NXP, we collaborated with a team of students from Linköping University in Sweden. This international partnership fostered an exchange of ideas, methodologies, and diverse perspectives, enriching the development process. It allowed us to address complex challenges with a multidisciplinary approach and highlighted the value of teamwork across borders in achieving innovative outcomes.

Throughout the project, we were also supported by Thierry Monteil, professor and researcher at INSA Toulouse, who provided guidance and academic supervision. His involvement played a key role in transforming our ideas into practical and innovative solutions.



Chapter 4

Background and Motivation: Why UWB for Keyless Car Access

This project builds on several years of ongoing research into keyless vehicle access. Rather than starting from scratch, we continued an existing development effort focused on a UWB/BLE-based digital key solution, supported by the NXP development ecosystem and a set of six NXP BLE/UWB boards provided for experimentation and validation. This naturally raises a broader set of questions: why is the automotive industry through companies such as ACTIA investing in this topic, and why are semiconductor vendors like NXP supporting it as an industrial solution? Most importantly, why choose Ultra-Wideband (UWB) over other wireless technologies? The answer lies in UWB's unique ability to deliver secure, precise ranging and reliable proximity verification, which are critical requirements for modern "keyless" access systems. At NXP, the long-term objective goes beyond simply knowing whether the user is "near" the vehicle: it is to achieve truly high-precision localization, accurate enough to determine where an object is located even within the cabin, for example, distinguishing whether a device is in the driver's seat area, in a passenger position, or elsewhere inside the car.

4.1 UWB (Ultra Wideband) technology

Ultra-Wideband (UWB) is a short-range wireless communication technology operating over a very wide frequency spectrum, typically from 3.1 GHz to 10.6 GHz. Unlike conventional wireless technologies such as Bluetooth or Wi-Fi, which rely on narrowband transmissions, UWB is based on the emission of very short-duration pulses spread over a wide bandwidth. This transmission principle provides specific advantages in terms of ranging accuracy, robustness, and security.

The distinctive characteristics of UWB include:

- **High Precision:** UWB enables distance estimation with centimeter-level precision through Time of Flight (ToF) measurements. This makes it particularly suitable for applications requiring accurate positioning and localisation.
- **Low Interference:** Due to its wide bandwidth and low spectral power density, UWB is less affected by interference from other wireless systems, allowing reliable operation in dense or electromagnetically congested environments.
- **High Data Rates:** UWB supports fast signal exchanges with very low latency, which is essential for real-time localisation and control applications.
- **Low Power Consumption:** The low transmission power of UWB makes it well suited for embedded and portable systems, where energy efficiency is a key constraint.

The table below presents a comparative overview of the main technologies used for distance estimation and localisation. Among these solutions, Ultra Wideband (UWB) demonstrates significantly higher localisation accuracy, reaching the centimeter scale.



Table 4.1: Comparison between technologies for distance estimation and localization.

| Technology | Accuracy | Latency |
|------------|----------|---------|
| UWB | 1 cm | < 1 ms |
| Bluetooth | 1–5 m | > 3 s |
| WiFi | 5–15 m | > 3 s |
| RFID | 1 m | 1 s |
| GPS | 5–20 m | 100 ms |
| 5G | 10 m | < 1 s |

Ultra-Wideband (UWB) technology has emerged as a game-changing solution across various industries due to its unmatched capabilities in precision, reliability, and security. Among its many applications, UWB has shown significant promise in revolutionizing vehicle access systems by enabling intelligent and secure localization.

4.2 UWB for Intelligent Vehicle Access

In automotive systems, UWB contributes to improving both the functionality and security of keyless entry solutions. Unlike conventional proximity-based technologies, UWB allows the vehicle to determine the position of a digital key accurately and to authorise access only when the user is located within a predefined and valid area. This functionality relies on the bidirectional exchange of short-duration radio signals between a connected device, acting as a digital key, and multiple UWB anchors integrated into the vehicle. By measuring the Time of Flight (ToF) of these signals with high temporal resolution, the system can estimate the precise location of the device, thereby enabling reliable differentiation between users located outside, near, and inside the vehicle.

4.3 Key Advantages of UWB in Vehicle Access

The use of UWB technology in vehicle access systems provides several major benefits:

1. Precision and Security: Centimeter-level distance estimation allows the system to verify the actual position of the digital key, significantly reducing the risk of relay attacks and other proximity-based security threats.
2. Responsiveness: The low latency of UWB enables rapid system response, allowing vehicle functions such as unlocking or driving authorisation to be triggered without perceptible delay.
3. Robustness: UWB remains effective in environments where satellite-based localisation systems, such as GPS, are unavailable or unreliable, including indoor and underground areas.

4.4 Objectives of UWB-Enhanced Vehicle Systems

The integration of Ultra-Wideband (UWB) technology into vehicle access systems enables several advanced functionalities aimed at improving localisation accuracy and system reliability. The main objectives include:

- User Position Triangulation: Accurate determination of the position of the digital key or connected device relative to the vehicle, enabling precise and context-aware access control.
- Obstacle Management: Ensuring stable and reliable system performance in the presence of obstacles or signal disturbances that may affect radio propagation.
- Dynamic Scenarios: Adapting vehicle behaviour to user movements in real time, such as automatic unlocking or activation of vehicle functions based on the user's position.

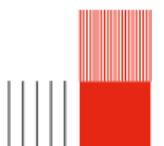
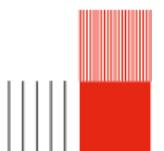




Figure 4.1: UWB technology for secure and intelligent vehicle access using a digital key.

In summary, UWB is particularly well suited for modern keyless access because it enables reliable proximity verification with a level of spatial resolution that other short-range technologies typically cannot achieve. This precision is not only relevant for deciding whether a user is authorized to unlock or start the vehicle, but also for finer-grained localization use cases, such as identifying where a device is located within the cabin.

Building on this motivation, the following chapters shift from why UWB is a strong enabler of secure keyless access to how we engineered a complete and measurable system: starting from NXP’s reference code originally limited to point-to-point ranging, we extend the pipeline with estimation and filtering steps to reduce noise, reject inconsistent measurements, and obtain more stable and accurate position estimates. We then present the cloud integration used to stream and persist distance logs for monitoring and traceability. Finally, we quantify the energy cost of an always-on UWB infrastructure through practical bench measurements and a system-level power model, leading to concrete conclusions on the power-supply and duty-cycling decisions that ultimately govern real deployments.

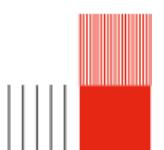


Chapter 5

The project and its main objectives

This project prototypes and evaluates a secure BLE/UWB-based keyless entry system for automotive access, using a smartphone as a digital key. It is structured around five concrete objectives:

1. **Improve positioning accuracy and stability in vehicle scenarios.** We studied the end-to-end localisation pipeline (ranging → position estimation → filtering) and identified limitations of the initial approach (sensitivity to initialization, convergence instabilities, and unrealistic jumps under noisy UWB measurements). We then proposed an improved estimator based on nonlinear least-squares with temporal warm-starting (using the previous state) and robustness refinements (measurement validation and optional weighting), leading to smoother and more physically consistent trajectories in the evaluated datasets.
2. **Implement a cloud pipeline for real-time experiment logging and traceability.** To enable remote monitoring and maintain an auditable history of experiments, we implemented a secure MQTT link to AWS IoT Core and a persistence layer in DynamoDB using IoT Rules. During development, a real-time stream was emulated by publishing records from a CSV-like source at a fixed rate; the architecture is designed to remain unchanged when replacing this source with live outputs from the embedded system.
3. **Quantify energy consumption and derive parked-vehicle autonomy bounds, then explore mitigations.** Because the UWB boards rely on a host workflow during ranging, direct standalone power profiling is not straightforward. We therefore combined oscilloscope-based current measurements (shunt resistor method) with a system-level consumption model. This allowed us to estimate order-of-magnitude autonomy for a parked vehicle and to evaluate mitigation strategies such as a dedicated auxiliary battery and rooftop photovoltaic (PV) harvesting for long parking durations.
4. **Compute the user position in real time and stream it to the cloud with a graphical interpretation of proximity.** Beyond offline evaluation, we aimed to operationalize the pipeline by producing real-time estimates of the user's state relative to the vehicle (position and/or distance-to-car) and publishing them to the cloud. This objective includes structuring the payload (timestamp, device identifiers, distances/position), streaming over MQTT, and providing a live visualization that highlights the evolution of proximity (e.g., distance vs. time while the user approaches, reaches the door area, and moves away), enabling immediate interpretation during experiments.
5. **Integrate BLE and UWB into a first end-to-end functional test (BLE discovery/authentication → UWB ranging).** To validate system-level feasibility, we integrated the BLE and UWB components into a single operational workflow. BLE is used as the low-power entry point for discovery and access verification at longer range, then a controlled handover triggers UWB ranging when proximity conditions are met. The goal of this first integration test is to make both technologies operate together in a consistent chain, producing coherent real-time measurements and laying the foundation for subsequent optimization (handover criteria, duty-cycling, robustness under obstacles, and vehicle-scale deployment).



Chapter 6

Project Organization

In week 40, the project officially started with the reception of the NXP Ultra-Wideband (UWB) development boards and the first team coordination meeting, during which the objectives and initial organisation of the project were defined. During this initial phase, a coordination meeting was also held with the ACTIA team in France and the Swedish student team involved in the same project. This meeting allowed all teams working on the project to introduce themselves, clarify their respective roles, and establish a common understanding of the project scope and objectives.

During the same week, we participated in a hackathon organised within the framework of an ECIU University challenge led by ACTIA Group. This hackathon focused on the integration of wireless technologies, namely Bluetooth Low Energy (BLE) and Ultra-Wideband (UWB), to accurately detect and localise users around a vehicle, with the objective of enabling keyless access solutions for connected vehicles. Students from electrical engineering, computer science, and international partner universities collaborated during this event. The hackathon provided an opportunity to discuss project objectives and technical strategies with the Swedish team, gain hands-on experience with the hardware, and validate the relevance of the proposed use case. A communication group was created at this stage in order to maintain regular contact between the teams and ensure coordination between the UWB and BLE parts of the project.

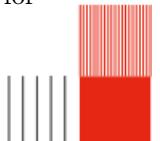
During week 41, the team focused on consolidating the outcomes of the kickoff phase. Internal meetings were organised to clarify the project scope, distribute tasks, and plan the next development steps. From this point onward, regular follow-up meetings were held with Mr. Pierre Gruyer throughout the project, generally every one to two weeks, in order to review progress, discuss encountered difficulties, and adjust the project orientation when necessary.

In week 42, we attended a technical presentation on UWB technology, which deepened our understanding of its principles, constraints, and potential applications in an automotive context. In parallel, development activities continued, and we familiarised ourselves with the UWB hardware and software environment. During the same period, a coordination meeting was held with the Swedish team to clarify the conditions required to trigger the transition from Bluetooth Low Energy (BLE) to Ultra-Wideband (UWB), and to identify open technical points related to this handover.

Week 43 was mainly dedicated to software development, with all team members working jointly on the localisation code. During this phase, we progressively realised that having the entire team focused on the same tasks was not the most efficient approach. Task overlap and limited testing efficiency highlighted the need for a reorganisation of the workflow in order to improve productivity and result quality.

In week 45, the project organisation was adapted in response to these observations. The team progressively moved towards a parallel development approach. Part of the effort was dedicated to improving the accuracy and stability of the UWB localisation code, while another part focused on initiating the cloud-related development. The objective of the cloud work at this stage was to validate a coherent and secure data transmission chain, from data generation to storage, without aiming for advanced cloud-side processing.

Week 46 was dedicated to stabilising the localisation algorithms and continuing the cloud development in parallel. On the localisation side, efforts focused on improving measurement repeatability and understanding the sources of inaccuracy. On the cloud side, a basic architecture was set up to ensure secure communication, data transmission, and persistent storage of measurements, providing a foundation for later analysis.



During week 47, the improved localisation code was further refined and evaluated using datasets from previous project iterations. This allowed us to validate the behaviour of the system in a controlled environment and identify remaining sources of error. In parallel, the cloud pipeline was validated end-to-end to ensure that transmitted data could be correctly received, processed, and stored.

In week 48, another coordination meeting with the Swedish team was organised in order to better understand the Bluetooth Low Energy (BLE) component and prepare for its integration with the UWB-based localisation system. This exchange was essential to align both parts of the project before system-level integration.

Week 49 focused on the preparation of new experimental datasets to validate the updated localisation code, as well as the study of the documentation related to the BLE development board and the work carried out by the Swedish team, in anticipation of the next project phase.

Finally, from week 50 to week 2, the project entered its final phase. Real-world vehicle-level tests were conducted in order to evaluate the system behaviour under realistic conditions and to assess its overall performance and robustness. In parallel, we consolidated the software pipeline by improving the localisation code and the data output workflow: we implemented real-time user position computation and produced a structured time-series output that can be continuously generated during experiments. This stream is formatted to be directly published to the cloud for ingestion and persistent storage of measurements, while also enabling live visualization of the trajectory evolution (and proximity trends) during the test session. During this final phase, additional work was also dedicated to the energy aspect of the system, including current-consumption measurements and system-level autonomy estimation, as well as the exploration of mitigation options for long parked-vehicle periods.



Chapter 7

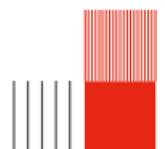
Objective 01 - Improving Positioning Accuracy

This chapter addresses the first objective of the project: improving the accuracy and robustness of UWB-based localisation in realistic keyless-entry vehicle conditions. We first define the evaluated vehicle configuration and scenario, then describe the baseline pipeline used in the previous implementation and its observed limitations on real measurements. Finally, we present the proposed improvements, nonlinear least-squares estimation, warm-start initialisation, measurement robustness mechanisms, and tighter coupling with Kalman filtering, and we conclude with a comparative analysis on two experimental datasets.

7.1 Vehicle Configuration and Evaluated Scenario

To evaluate and improve positioning accuracy under realistic conditions, we defined a vehicle-level configuration intended to replicate the constraints of a real keyless entry system. The objective was to configure the antennas in a way that enables reliable detection of the user's position and supports the key requirement of determining whether the user is *inside* or *outside* the car. After measuring the vehicle's main dimensions (height, length, and width) to establish a consistent geometric reference, we selected a single final scenario that corresponds to the target and desired configuration for the project.

More precisely, this final scenario places all five UWB receivers inside the vehicle: two near the windshield, two near the trunk, and one in the center of the cabin (Figure 7.1). This configuration was chosen because it is the intended final integration in a real vehicle, where keeping antennas inside the cabin is preferred for practical deployment and robustness. Although this setup is expected to be among the most challenging from a localization standpoint, due to attenuation, obstacles, and multipath effects caused by the vehicle structure, it is the most representative scenario for validating the improvements presented in the remainder of this section.



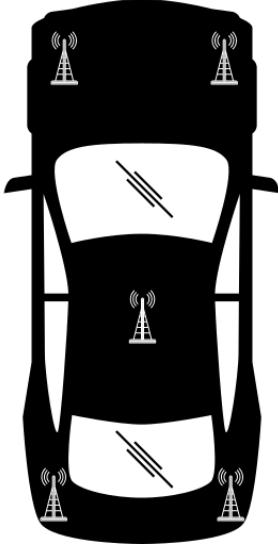


Figure 7.1: Final and desired configuration: all 5 receivers placed inside the vehicle (2 near the windshield, 2 near the trunk, and 1 centered).

7.2 Initial Positioning Approach

In the first implementation, the localisation pipeline was organised as a two-stage process. A ranging stage first provided distance measurements between the mobile node and each anchor, and a positioning stage then computed the user coordinates from these distances. Concretely, the ranging program running on the anchors periodically produced logs containing one measured distance per anchor, and the position was computed offline by a localisation script using these measured distances together with the known anchor coordinates.

Each distance measurement can be interpreted geometrically as a circle centred at anchor i with radius d_i . Under ideal conditions, the user position would correspond to a unique intersection point of all circles. In practice, UWB measurements are affected by noise and propagation effects, meaning that the circles rarely intersect at a single point. The localisation task therefore becomes an estimation problem rather than a direct geometric construction: the goal is to find the coordinates (x, y) that best satisfy all distance constraints simultaneously.

To compute this estimate, the initial solution relied on a Nelder–Mead minimisation scheme. At each time step, the algorithm iteratively adjusted the candidate position (x, y) to reduce an objective function built from distance errors, defined as the discrepancy between each measured distance d_i and the corresponding predicted distance.

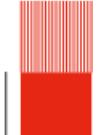
$$\sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (7.1)$$

The main advantage of Nelder–Mead in this context is its simplicity, as it does not require gradient information and can be implemented quickly for prototyping.

However, this approach also introduced major limitations when applied to real UWB data. Because distance measurements can be strongly impacted by multipath propagation and non-line-of-sight conditions, the resulting constraints may become inconsistent and lead the optimiser to unstable solutions. Moreover, the convergence of Nelder–Mead is highly dependent on the initial guess. In the first version, the minimisation was always initialised from a fixed starting point, typically $(0, 0)$, which often placed the solver far from the true solution. As a consequence, the optimisation could converge slowly, fall into local minima, and generate unrealistic position jumps when the measurement quality degraded.

Additional difficulties also appeared at the data level. The ranging logs were not always perfectly usable: some records were empty, and in rare cases fewer distance values were available when one anchor temporarily lost detection. Since the log format did not explicitly identify which anchor was missing, these samples could not be exploited reliably and were therefore ignored, which resulted in occasional gaps in the reconstructed trajectory.

To improve temporal consistency, a filtering stage was introduced on top of the raw optimisation outputs. A Kalman filter was used to smooth the sequence of estimated positions by exploiting the



fact that user motion is continuous over time. The filter maintains an internal state, typically including position and velocity, and alternates between a prediction step, which propagates the state using a simple kinematic model (for instance constant velocity), and an update step, which corrects the prediction using the position measurement provided by the optimiser. In this way, the Kalman filter behaves as a principled low-pass filter: it reduces high-frequency jitter, attenuates part of the measurement noise, and produces a more physically plausible trajectory when the optimisation results remain reasonably consistent.

Nevertheless, the filtering stage cannot fully compensate for severe optimisation failures. When the minimisation converges to an incorrect solution due to poor initialisation or strong non-line-of-sight bias, the resulting position measurement can be far from the real location. In such cases, the Kalman update may still introduce drifts or abrupt corrections, and the overall pipeline can exhibit instabilities, especially during approaching and entering phases where the radio channel is more challenging.

7.3 Proposed Solution

7.3.1 Nonlinear Least-Squares Position Estimation

To address the limitations observed with the initial Nelder–Mead approach (sensitivity to the initial guess, unstable convergence under noisy measurements, and inconsistent geometric constraints), the positioning problem was reformulated as a nonlinear least-squares estimation problem. Rather than relying on a generic minimisation process, this approach explicitly models localisation as the problem of finding the position that best fits all available distance measurements in a global and coherent manner.

For a candidate position (x, y) , each anchor i provides a measured distance d_i . The corresponding predicted distance to anchor i is given by $\sqrt{(x - x_i)^2 + (y - y_i)^2}$, where (x_i, y_i) denotes the known anchor coordinates. The mismatch between the predicted and measured distances is captured by the residual

$$e_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i. \quad (7.2)$$

If measurements were perfect, all residuals would be zero at the true position. In real conditions, residuals remain non-zero due to noise, multipath, and non-line-of-sight effects, which means that the circles defined by the anchors do not intersect at a unique point.

The least-squares formulation resolves this inconsistency by searching for the position that minimises the total squared residual error over all anchors:

$$J(x, y) = \sum_{i=1}^N e_i(x, y)^2. \quad (7.3)$$

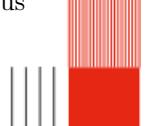
Minimising $J(x, y)$ yields the position that provides the best overall agreement with the full set of distances, instead of overreacting to a single biased measurement. In other words, the solution is a best-compromise estimate that distributes the error across anchors and reduces unrealistic jumps caused by local inconsistencies.

This reformulation is also well grounded from a statistical perspective. Under the common assumption that ranging errors are approximately Gaussian and independent, minimising the sum of squared residuals is equivalent to a maximum likelihood estimate of the user position. This gives the method a clear theoretical interpretation and makes it naturally robust for real-world UWB data where perfect geometric intersections cannot be expected.

7.3.2 Warm-Start from the Previous State

A second major improvement concerns the initialization of the optimisation process. Instead of restarting the solver from a fixed arbitrary point (e.g., the origin), the algorithm now uses the previous estimated position as the initial guess for the next estimation step. In practice, this initialisation is taken from the Kalman filter state at time $k - 1$, which provides a physically consistent estimate of the user position based on past measurements and the motion model.

This warm-start strategy directly exploits temporal continuity: in a real trajectory, the user cannot move arbitrarily far between two consecutive timestamps. Therefore, the solution at time k is expected to remain close to the solution at time $k - 1$. Initialising the least-squares solver with the previous



state places the optimiser near the true solution from the first iteration, which significantly improves the numerical behaviour of the estimation.

As a result, convergence becomes faster and more stable, because the algorithm spends less time exploring irrelevant regions of the search space and is less likely to be attracted by local minima created by noisy or biased distance measurements. This also reduces the occurrence of unrealistic jumps between successive estimates, since the optimisation naturally remains consistent with the recent motion history.

Overall, coupling the numerical optimisation with the Kalman filter provides a coherent temporal loop: the filter delivers a reliable state estimate that guides the next optimisation step, and the new optimisation output becomes the measurement used to update the filter. This mutual reinforcement improves robustness to noise and produces a smoother and more physically plausible trajectory.

7.3.3 Weighted Residuals for Measurement Robustness

Even with a least-squares formulation, not all ranging measurements have the same reliability. In practice, UWB distances tend to become noisier when the device is farther from an anchor or when the propagation conditions degrade (e.g., multipath or partial non-line-of-sight). To reduce the influence of such degraded measurements, the cost function can be extended to a weighted least-squares criterion in which each residual is scaled by a confidence weight:

$$J(x, y) = \sum_{i=1}^N w_i e_i(x, y)^2. \quad (7.4)$$

A simple and effective strategy is to assign smaller weights to larger ranges, which are more likely to be affected by attenuation and multipath. This prevents a single unreliable anchor from dominating the optimisation and improves stability when the geometry becomes inconsistent. As a result, the estimated position becomes less sensitive to outliers and better reflects the most informative anchors at each time step.

7.3.4 Solver Choice: Least-Squares Dedicated Optimisation

Beyond the mathematical reformulation, the numerical solver used to minimise the cost function is a major component of the overall accuracy and stability. General-purpose optimisers such as Nelder–Mead are convenient for prototyping but can be slow and sensitive to initialisation when the objective is noisy or ill-conditioned. In contrast, a dedicated nonlinear least-squares solver (e.g., a Levenberg–Marquardt-type method) is specifically designed for objectives expressed as sums of squared residuals.

This class of solvers exploits the structure of the problem to converge more efficiently, typically requiring fewer iterations and providing more reliable behaviour under small perturbations of the measurements. In a real-time localisation context, this translates into faster convergence, improved numerical stability, and more consistent estimates from one time step to the next.

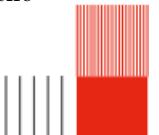
7.3.5 Handling Invalid and Incomplete Measurements

Real ranging logs may contain invalid samples (missing values, empty records, or partial anchor updates). If such measurements are fed directly into the estimator, they can either break the numerical pipeline or inject incoherent constraints that destabilise the solution. To avoid this, the localisation stage must include a pre-processing step that validates the measurement vector before optimisation.

Invalid samples are excluded from the computation, and only consistent anchor-distance pairs are retained. This guarantees that the least-squares solver operates on a well-defined residual vector and prevents artefacts in the estimated trajectory. More generally, this measurement validation step is necessary to ensure robustness to intermittent anchor detection and to maintain a stable processing chain even when the raw data stream is imperfect.

7.3.6 Tighter Coupling Between Optimisation and Kalman Filtering

Temporal filtering becomes significantly more effective when it is tightly coupled with the optimisation stage rather than applied as a loose post-processing step. In the proposed pipeline, the Kalman filter does not only smooth the trajectory; it also provides a physically consistent state estimate that guides the



next optimisation step. This creates a coherent loop in which the filter prediction supplies an informed initialisation, and the optimised position provides a refined measurement update for the filter.

This interaction improves global stability because the optimiser is continuously anchored to a plausible motion model, while the filter benefits from a measurement that is already geometrically consistent with the anchor constraints. In practice, this coupling reduces jitter, limits unrealistic jumps, and increases robustness during challenging phases such as approaching the vehicle or transitioning across regions with degraded radio propagation.

7.4 Comparison of Results

This section provides a direct comparison between the trajectories obtained in the previous project iteration and those produced by the improved method. In Figure 7.2, the previous approach shows a trajectory that is strongly affected by dispersion and abrupt changes of direction, especially around the vehicle area. The estimated track exhibits visible jitter and several unrealistic corrections, which indicates that the position estimates were highly sensitive to ranging noise and that the optimisation occasionally converged to inconsistent solutions. In contrast, the improved method produces a trajectory that is noticeably more stable and physically coherent: the path is smoother, the motion is more continuous, and the estimated positions remain concentrated around the expected user movement without excessive oscillations.

A similar behaviour is shown in Figure 7.3. With the previous approach, the estimated track contains large jumps and long segments that do not match a plausible user movement, suggesting convergence issues and local minima effects when measurements become degraded. The improved method significantly reduces these artefacts: the track follows a more consistent progression and avoids sudden discontinuities, which reflects a better numerical conditioning of the estimator and a stronger temporal coherence. Overall, across both datasets, the comparison highlights that the improved pipeline provides more reliable localisation under identical measurement conditions, with reduced sensitivity to outliers, fewer unrealistic position jumps, and a trajectory that better matches the expected physical motion during the approaching and entering phases.

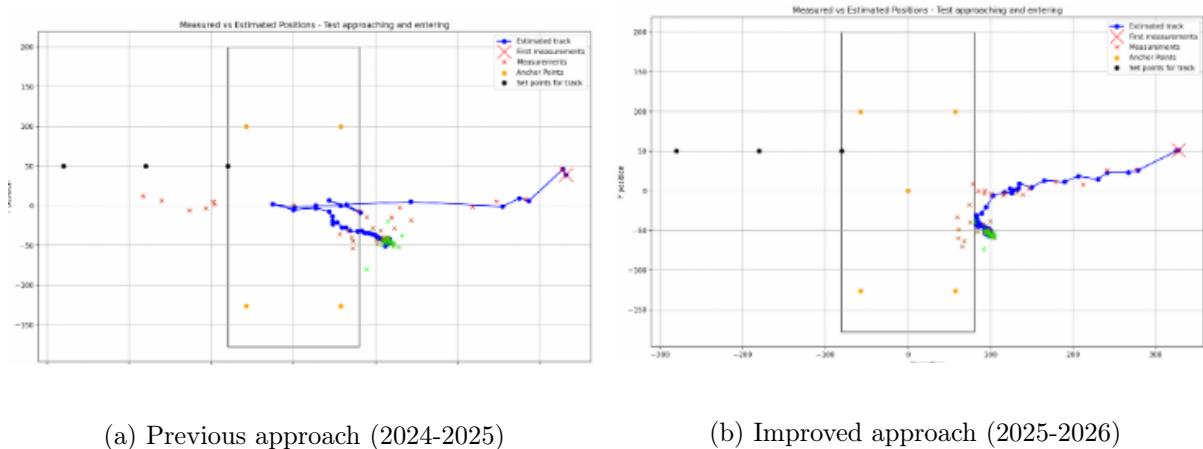
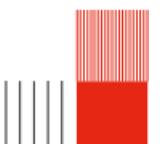
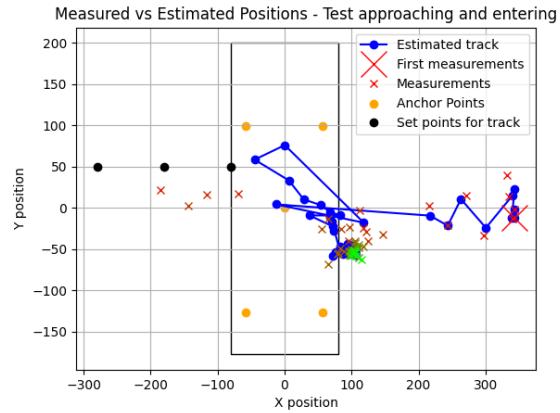
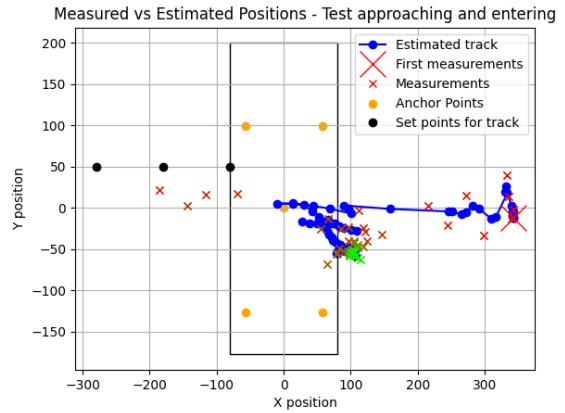


Figure 7.2: Measured vs estimated positions for the approaching and entering test (dataset 1): comparison between last year's results and the improved method.



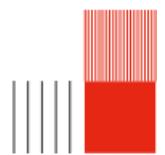


(a) Previous approach (2024-2025)



(b) Improved approach (2025-2026)

Figure 7.3: Measured vs estimated positions for the approaching and entering test (dataset 2): comparison between last year's results and the improved method.



Chapter 8

Objective 02 - Cloud

A key requirement of our project is integrating the system with a cloud backend to enable remote monitoring and persistent storage of data from the car sensors. This cloud connection is essential for both technical validation and security-oriented traceability. By centralizing measurements and system events, the project can maintain an auditable history of the vehicle's behavior over time, which is useful in situations where an investigation is required. In a company context, such records can support the analysis of potential misuse, for instance, when a vehicle is shared without authorization or when abnormal proximity patterns between the user and the car are observed. In addition to traceability, cloud connectivity significantly enhances demonstration capabilities, allowing us to display a real-time interface that shows the vehicle's state and sensor readings. This makes it easy to illustrate the system's behavior during scenarios, such as a user approaching the car, and the measured distances evolving accordingly.

8.1 Cloud provider selection

To design the cloud component of our project, we first evaluated what would be required to collect, store, and visualize distance-related measurements in a secure and reliable way. Our first reference came from the Machine-to-Machine course, where a typical approach combines Node-RED and an MQTT broker (e.g., Mosquitto) to build a publish/subscribe pipeline that is easy to deploy and extend. This setup is ideal during early development because it accelerates prototyping and testing. However, it remains fundamentally a self-hosted solution: it depends on a server that must run continuously and be accessible, and extending access beyond the local network usually demands extra infrastructure and security reinforcement. Considering that such constraints are less aligned with common industrial practices, we moved toward managed IoT cloud services and performed a comparative study of the main providers.

Given these requirements and practical constraints, we narrowed our comparison to the two most appropriate and widely used options for our embedded/IoT context: AWS and Microsoft Azure. Both provide mature IoT stacks (AWS IoT Core and Azure IoT Hub) with device authentication, encrypted communication, and straightforward integration with databases and monitoring/analytics tools, which are essential capabilities for securely collecting and storing BLE and UWB sensor measurements at scale.

Based on these criteria, and prioritizing a well-documented workflow and smooth prototyping, we selected AWS IoT Core as the most suitable option. Beyond the fact that AWS is one of the most widely adopted cloud ecosystems for IoT, it offered a very direct and structured path for our first integration steps: we could rely on official sample code and a well-defined setup procedure to configure the IoT endpoint, device authentication through X.509 certificates, and a TLS-protected MQTT link consistent with standard IoT security practices. This reduced uncertainty during early development and allowed us to validate the connectivity and message routing quickly using the built-in AWS console tooling (e.g., subscribing to the MQTT topic with the MQTT Test Client), before introducing project-specific payloads.

In addition, AWS provided a straightforward native integration between secure ingestion and managed storage services for experiment logging. In our implementation, this was critical because our objective was not only to stream data but also to persist an experiment history for later inspection and analysis. AWS IoT Core supports a rule-based routing mechanism (IoT Rules) that can subscribe to our MQTT topic and forward selected fields to storage services without adding complexity on the “device” side; paired with DynamoDB, this gave us a clean separation between data acquisition/publishing and cloud-side persistence. The approach also keeps the architecture extensible: the same pipeline can later be



connected directly to live ranging outputs (replacing the temporary file-based input used during early testing) while preserving the same topics, routing logic, and database-backed history. Finally, practical considerations also reinforced the choice, such as strong documentation and an accessible free trial period, which helped us implement a reliable baseline quickly and iterate with confidence.

8.2 Cloud implementation

With AWS IoT Core selected as our target platform, we began the implementation by establishing a secure MQTT communication channel between our computer (used as a prototype “device” during development) and the AWS IoT Core endpoint, following the recommended AWS workflow. To validate the environment end-to-end, we first ran a reference example that publishes simple test messages. This step confirmed that authentication, topic routing, and connection stability were correct, and it allowed us to observe incoming publications directly in the AWS console using the MQTT test client.

Once the connection was validated, we adapted the application to transmit data representative of our use case, namely distance measurements associated with the UWB sensors. At that stage, because real-time acquisition was not yet fully integrated into our software pipeline, we emulated a live flow by reading samples from a file exported from a spreadsheet and publishing them periodically. This approach allowed us to validate the cloud architecture independently of the final sensor acquisition chain. Messages were published to a dedicated MQTT topic and verified in real time within AWS, confirming that the system could stream measurements reliably.

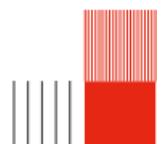
The next step was to store the data in a queryable form over time. To achieve this, we implemented a cloud pipeline in which the MQTT messages are ingested by AWS IoT Core and automatically forwarded to storage through an IoT Rule and appropriate access permissions, as illustrated in Figure 8.1. Each incoming message triggers a rule action that extracts the relevant fields from the payload and writes them into a database. This approach keeps the device side simple (it only needs to publish structured messages) while centralizing the storage logic in the cloud, which is easier to maintain and extend. For storage, we used DynamoDB and created a table tailored for time-based logging so that measurements can be efficiently retrieved as a history (e.g., grouped by device identifier and ordered by timestamp). To enforce proper access control, we created and attached an IAM role that allows the IoT Rules engine to perform the required write operations to DynamoDB, following a least-privilege approach. Finally, we validated the complete pipeline: each MQTT publication resulted in a corresponding entry in DynamoDB, confirming end-to-end operation from publisher to long-term storage; an example of this validation is shown in Figure 8.2.



Figure 8.1: Cloud pipeline

Even though our first integration tests did not rely on live UWB measurements, they were sufficient to validate the cloud pipeline end-to-end. At this stage, we used a simple Python script that periodically updated a .csv file with random (synthetic) distance values, allowing us to emulate a streaming behaviour and verify MQTT publication, IoT Core routing rules, and storage in DynamoDB.

For the final experiments, we transitioned to real measurements while keeping the same cloud architecture. We modified the NXP acquisition script so that, in addition to publishing the ranges, it also logs the measured distances into a .csv file using the same field structure. As a result, the integration script could be reused almost unchanged: we only adapted the file path and filename of the .csv source, and the rest of the pipeline remained identical. This confirmed a fully successful end-to-end operation, with consistent MQTT topics, unchanged IoT rules, and reliable persistence of the streamed payloads in the DynamoDB table.



Screenshot of the AWS DynamoDB console showing a table named "TestTable". The table has columns: center, datetime, front_left, front_right, rear_left, and rear_right. The data grid shows 116 items returned. A Windows PowerShell window is open in the foreground, displaying log messages related to publishing CSV rows to the topic "sdk/test/python".

```

Publishing CSV row #108 to topic 'sdk/test/python': {
    "datetime": "2025-12-15 16:48:13", "front_left": 10.7
    27919192927922122, "front_right": 4.5688029110227922122, "re
    ar_left": 10.3326048837709, "rear_right": 18.65809563
    78534114, "center": 16.3284842374989856}
PubAck received with <PubackReasonCode.SUCCESS: 0>

==== Received message from topic 'sdk/test/python': {
    "datetime": "2025-12-15 16:48:14", "front_left": 16.7
    27919192927922122, "front_right": 5.6923279292122, "re
    ar_left": 10.3326048837709, "rear_right": 18.65809563
    78534114, "center": 16.3284842374989856}
====

Publishing CSV row #109 to topic 'sdk/test/python': {
    "datetime": "2025-12-15 16:48:14", "front_left": 12.4
    56197355428706, "front_right": 2.6244843327452628, "re
    ar_left": 17.8946847175828, "rear_right": 6.655643
    99557466, "center": 15.997681880332871}
PubAck received with <PubackReasonCode.SUCCESS: 0>

Publishing CSV row #110 to topic 'sdk/test/python': {
    "datetime": "2025-12-15 16:48:14", "front_left": 10.8
    70951620828706, "front_right": 4.5688029110227922122, "re
    ar_left": 3.4780259612865998, "rear_right": 2.4931952
    094870824, "center": 15.71079192632856}
==== Received message from topic 'sdk/test/python': {
    "datetime": "2025-12-15 16:48:14", "front_left": 12.4
    56197355428706, "front_right": 2.6244843327452628, "re
    ar_left": 17.8946847175828, "rear_right": 6.655643
    99557466, "center": 15.997681880332871}

```

Figure 8.2: Screen during tests storing data in DynamoDB

Chapter 9

Objective 03 - Energy

This chapter studies the energy impact of the BLE/UWB keyless-access infrastructure and derives order-of-magnitude autonomy bounds for a parked vehicle. The main challenge is that the UWB boards cannot run fully standalone during ranging (they depend on a host workflow), which makes direct power-analyzer measurements difficult to isolate. We therefore combine (i) oscilloscope-based current profiling using a shunt resistor and (ii) a system-level consumption model to estimate parked autonomy and mitigation options.

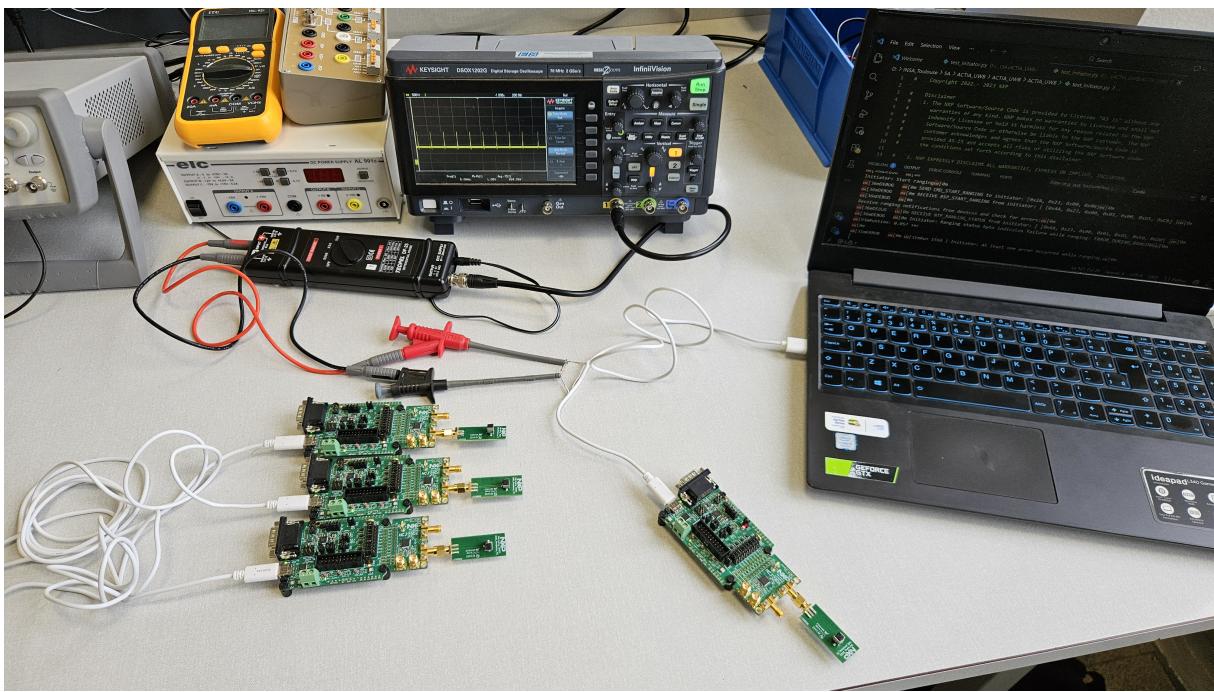


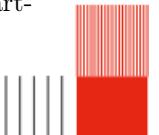
Figure 9.1: Experimental power-measurement setup: NXP UWB boards powered through a shunt-resistor cable and monitored with a Keysight oscilloscope while the ranging application runs on the host PC.

9.1 Current measurement method (shunt resistor)

To measure the board supply current without modifying the hardware, we inserted a shunt resistor in series with the USB 5 V supply line and measured the shunt voltage with a differential probe:

$$I(t) = \frac{V_{\text{shunt}}(t)}{R_{\text{shunt}}} \quad (9.1)$$

In our setup, we used $R_{\text{shunt}} = 1.3 \Omega$ to limit the voltage drop while keeping sufficient resolution. Two representative configurations were evaluated: (i) initiator-side consumption (board representing the smart-



phone side) and (ii) responder-side consumption (one responder powered through the shunt cable) at two distances (20 cm and 5 m).

9.2 Results 1 – Initiator

Figure 9.2 compares the initiator current consumption in two conditions: (i) idle mode (board powered, no application running) and (ii) active mode (UWB ranging enabled, code running). The measured average current increases from about 92.3 mA in idle mode to 109.3 mA during communication.

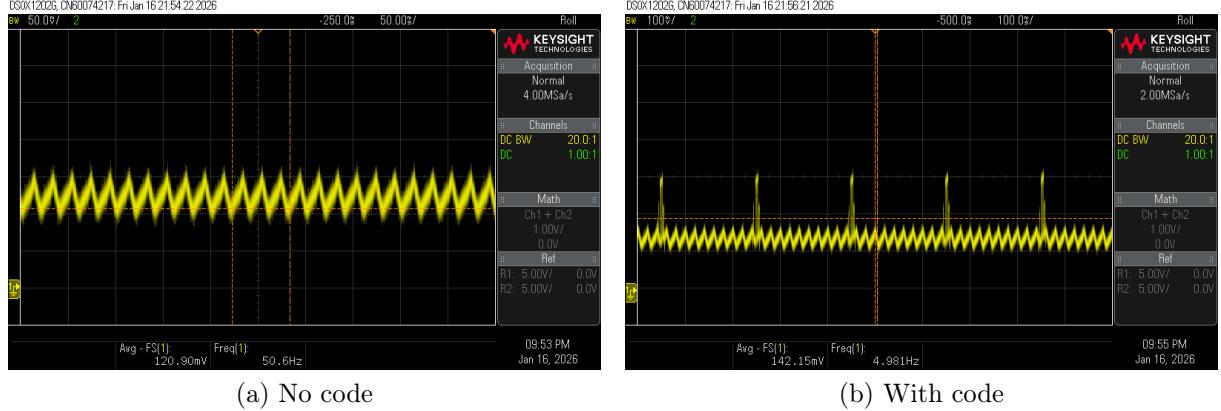


Figure 9.2: Initiator consumption: idle vs UWB communication.

The higher mean current in the communication mode is mainly explained by periodic UWB ranging transactions: RF transmission/reception bursts, digital processing, and SPI activity in the UWB transceiver. This duty-cycled behavior is visible in Figure 9.3, where three distinct peaks appear. They match our experimental setup with three responder boards: within one cycle, the initiator performs one ranging exchange per responder, generating one current burst per responder.

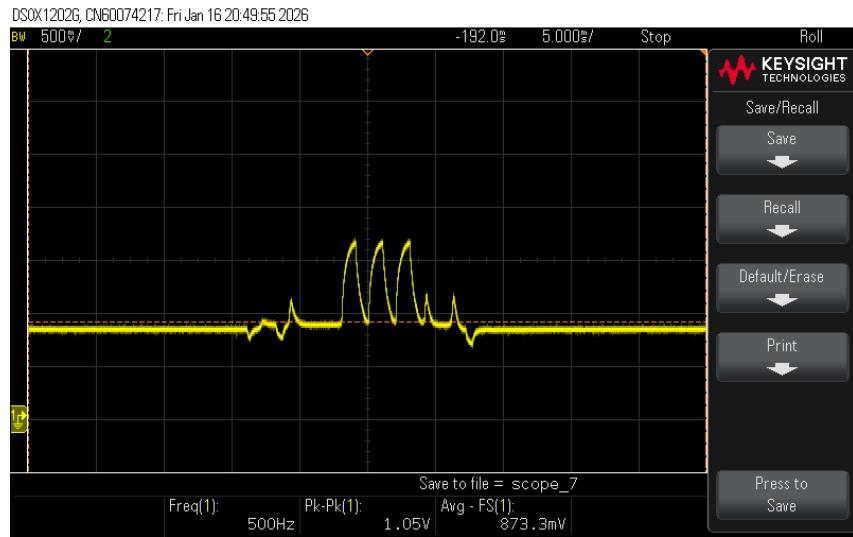
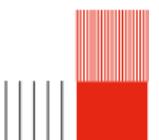


Figure 9.3: Initiator waveform showing three current peaks (one per responder within a cycle).

9.3 Results 2 – Receptors (responders)

Figure 9.4 shows the responder shunt-voltage waveforms for two distances from the initiator. From the oscilloscope average values, the mean shunt voltage increases from about 151 mV at 20 cm to about 210 mV at 5 m. Using $R_{\text{shunt}} = 1.3 \Omega$ (Eq. 9.1), this corresponds to an average current increase from ≈ 116 mA to ≈ 162 mA.



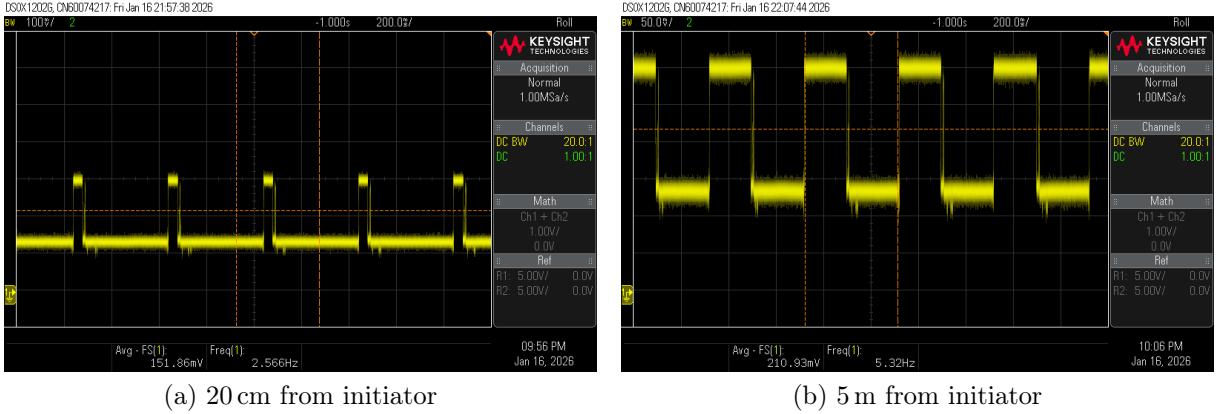


Figure 9.4: Responder consumption profile (via shunt) at two distances.

In both cases, the responder does not draw a constant current: the waveform alternates between a lower baseline and repeated higher-consumption phases. These active phases correspond to periodic wake-ups for UWB ranging (receive windows, response transmission, and processing). A noticeable difference is the activity rate: at 20 cm the repetition frequency is about 2.6 Hz, whereas at 5 m it increases to about 5.3 Hz, which increases the duty cycle of active phases and thus the average consumption. In practice, longer distances can lead to a more demanding link (longer listening time and/or retries), increasing the overall average current.

9.4 System-level power model and parked autonomy

9.4.1 Per-node current model

Because development boards include non-essential peripherals (debug interfaces, LEDs, extra regulators), we adopted a deployment-oriented model assuming a minimal custom node dedicated to BLE/UWB. From our measurements, enabling UWB increases the average current by about 17 mA; we conservatively round the incremental UWB contribution to 20 mA. We assume a minimal baseline of 40 mA and model BLE as 0.3 mA (100 ms connection interval, conservative worst-case). Two operating cases are considered:

$$I_{\text{node,A}} \approx I_{\text{base}} + I_{\text{BLE}} \approx 40 \text{ mA} + 0.3 \text{ mA} = 40.3 \text{ mA}, \quad (9.2)$$

$$I_{\text{node,B}} \approx I_{\text{base}} + I_{\text{UWB}} + I_{\text{BLE}} \approx 40 \text{ mA} + 20 \text{ mA} + 0.3 \text{ mA} = 60.3 \text{ mA}. \quad (9.3)$$

9.4.2 Vehicle-side power and DC–DC conversion

The in-vehicle infrastructure includes $N = 5$ nodes (4 anchors + 1 central unit), each supplied at $V_{\text{load}} = 5 \text{ V}$:

$$P_{\text{tot}} = N V_{\text{load}} I_{\text{node}}, \quad (9.4)$$

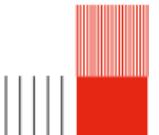
$$P_{\text{tot,A}} = 5 \cdot 5 \cdot 0.0403 = 1.0075 \text{ W}, \quad P_{\text{tot,B}} = 5 \cdot 5 \cdot 0.0603 = 1.5075 \text{ W}. \quad (9.5)$$

With a 12 V low-voltage battery and a buck converter efficiency $\eta \approx 0.90$, the equivalent battery current is approximated by:

$$I_{\text{bat}} \approx \frac{P_{\text{tot}}}{\eta V_{\text{bat}}}, \quad V_{\text{bat}} = 12 \text{ V}. \quad (9.6)$$

Thus,

$$I_{\text{bat,A}} = \frac{1.0075}{0.90 \cdot 12} = 0.0933 \text{ A}, \quad I_{\text{bat,B}} = \frac{1.5075}{0.90 \cdot 12} = 0.1396 \text{ A}.$$



9.4.3 Starter-battery autonomy (parked vehicle)

We consider a representative automotive lead-acid starter battery of $C_{\text{bat}} = 60 \text{ Ah}$. To preserve cold-start capability and include margin for aging/temperature, we apply a conservative constraint $\text{SoC}_{\min} = 80\%$ (i.e., $\text{DoD}_{\max} = 20\%$), yielding a usable charge $C_{\text{usable}} = 12 \text{ Ah}$. Assuming a constant average load and neglecting second-order effects, the operating time is:

$$t \approx \frac{C_{\text{bat}} \cdot \text{DoD}}{I_{\text{bat}}}. \quad (9.7)$$

Therefore,

$$t_A = \frac{12}{0.0933} = 128.6 \text{ h} \approx 5.36 \text{ days}, \quad t_B = \frac{12}{0.1396} = 86.0 \text{ h} \approx 3.58 \text{ days}.$$

These bounds indicate that always-on access electronics must be carefully budgeted for long parking periods, especially in worst-case continuous-UWB operation.

9.5 Mitigation strategies for long parking

9.5.1 Dedicated auxiliary battery (LiFePO₄ 12.8 V – 30 Ah example)

A robust solution is to decouple access-system availability from the starter battery by adding a dedicated auxiliary battery recharged while driving. As an example, we considered the ECO-WORTHY LiFePO₄ battery (12.8 V, 30 Ah) shown in Figure 9.5.

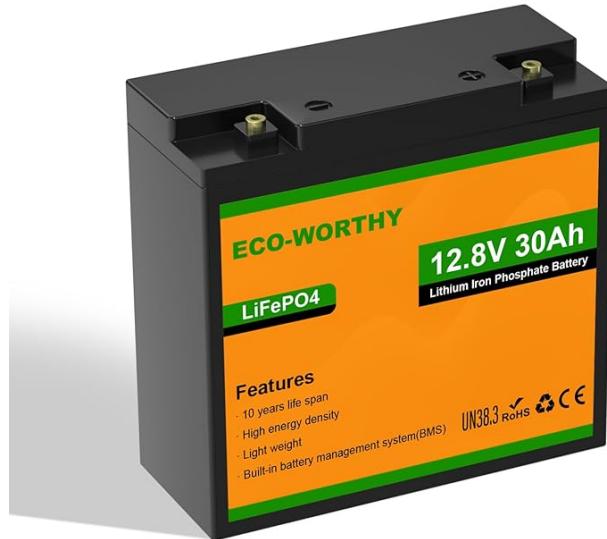


Figure 9.5: Example of a dedicated auxiliary battery: ECO-WORTHY LiFePO₄ 12.8 V, 30 Ah (integrated BMS 25 A).

Because the nominal voltage is $V_{\text{aux}} = 12.8 \text{ V}$, we recompute the battery-side currents:

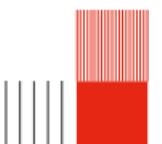
$$I_{\text{bat}} \approx \frac{P_{\text{tot}}}{\eta V_{\text{bat}}}, \quad V_{\text{bat}} = 12.8 \text{ V}. \quad (9.8)$$

$$I_{\text{bat,A}} = \frac{1.0075}{0.90 \cdot 12.8} = 0.08746 \text{ A}, \quad I_{\text{bat,B}} = \frac{1.5075}{0.90 \cdot 12.8} = 0.13086 \text{ A}.$$

Unlike the starter battery, the auxiliary battery is dedicated to standby electronics; we adopt a conservative usable fraction $\text{DoD}_{\text{aux}} = 80\%$, giving $C_{\text{usable}} = 30 \times 0.80 = 24 \text{ Ah}$. The resulting autonomy is:

$$t_{A,\text{aux}} = \frac{24}{0.08746} = 274.4 \text{ h} \approx 11.43 \text{ days}, \quad t_{B,\text{aux}} = \frac{24}{0.13086} = 183.4 \text{ h} \approx 7.64 \text{ days}.$$

This extends parked availability while keeping the starter battery protected.

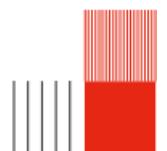


9.5.2 Ambient energy harvesting (rooftop photovoltaics)

A complementary option is rooftop photovoltaic (PV) harvesting to reduce net battery discharge during daylight. In automotive conditions, PV output depends strongly on roof area, irradiance, curvature/orientation, and conversion losses. A practical budgeting approach is to estimate a harvested power profile

$$P_{\text{PV}}(t) \approx \eta_{\text{PV}} \cdot \lambda(t) \cdot A \cdot \overline{A_{\text{eff}}} \cdot \eta_{\text{chg}} \quad (9.9)$$

then compare daily harvested energy $E_{\text{PV}} = \int_{\text{day}} P_{\text{PV}}(t) dt$ to the daily electronics demand $E_{\text{load}} \approx P_{\text{bat}} \cdot 24 \text{ h}$. If $E_{\text{PV}} < E_{\text{load}}$, PV still increases autonomy by lowering the effective discharge rate during parked periods.



Chapter 10

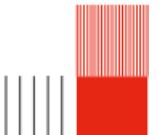
Objective 04 – Real-Time Position Computation, Cloud Streaming, and Live Visualisation

This chapter addresses the fourth objective of the project: moving from an offline evaluation of localisation performance to an operational pipeline capable of producing real-time estimates of the user state relative to the vehicle and making these estimates immediately interpretable during experiments. In practice, a key requirement for ACTIA is not only to compute positions “after the fact”, but to observe the system behaviour live, as a user approaches the car, reaches a door area, and then moves away, so that ranging quality, estimator stability, and proximity decisions can be validated under realistic conditions. This objective therefore combines three elements: (i) online position estimation, (ii) structured streaming of the resulting states to the cloud, and (iii) a live visual interpretation of proximity evolution.

In our initial workflow (Objective 01), the localisation algorithm was primarily evaluated in post-processing: the ranging application produced distance logs, and a separate script reconstructed the user trajectory afterwards by triangulation on the 2D vehicle plane. While this was appropriate for algorithmic development and comparison on datasets, it did not provide immediate feedback during a vehicle test. To operationalise the pipeline, we merged the two Python executables used in our experiments: the acquisition program responsible for collecting UWB ranges from the NXP boards, and the localisation program implementing the improved estimator (nonlinear least-squares with temporal continuity and filtering). Concretely, we developed a lightweight “wrapper” script that imports (or calls) both modules and executes them within a single real-time loop, ensuring that each newly acquired set of distances is directly fed to the position solver without intermediate manual steps.

At runtime, the unified script continuously retrieves the latest available ranges between the initiator (user side) and the set of in-vehicle anchors. Each measurement frame is time-stamped and validated (e.g., handling missing or invalid distances) before being passed to the estimator. The position is then computed online using the same principles described previously: the least-squares solver outputs an instantaneous coordinate estimate, and the temporal filtering stage provides a smoother and physically plausible trajectory update. In addition to the 2D position, we compute a proximity-oriented indicator that can be interpreted at a glance during experiments (for example, a scalar distance-to-vehicle reference point such as the driver-door area, or a condensed metric derived from the set of ranges). This allows the experimenter to relate the continuous trajectory to the expected keyless-entry states (approaching, near-door, moving away) in real time.

To connect this live state estimation to the cloud backend (Objective 02), we structured the output as a consistent payload that can be streamed over MQTT. Each published message is designed to be self-contained and directly usable for logging and visualisation. The payload includes a timestamp, device identifiers (initiator and/or vehicle-side identifiers), the set of raw distances (with explicit anchor labels when available), and the computed localisation outputs (estimated position and proximity indicator). These records are published continuously to the AWS IoT Core endpoint using the same secure MQTT link (TLS with X.509 authentication) already validated in the cloud chapter. By keeping the message format compatible with the previously defined ingestion rules, the pipeline remains extensible: the cloud side can ingest and persist the same fields in DynamoDB, while also enabling future dashboards that display proximity trends live.



Finally, beyond the cloud stream, we implemented a direct real-time visualisation on the experiment computer to provide an immediate interpretation of proximity and trajectory evolution during tests. Using the *matplotlib.animation* library, the live interface updates at each ranging cycle and displays the user's estimated position on the 2D vehicle plane, together with the anchor locations and the recent trajectory history. This real-time plot provides a clear, visual confirmation that the full chain is functioning end-to-end: acquisition of distances from the UWB boards, online position computation, cloud publication, and live interpretation. A representative capture of this demonstration is shown in Figure 10.1, and the corresponding video further illustrates the system's behaviour as the user moves around the vehicle.

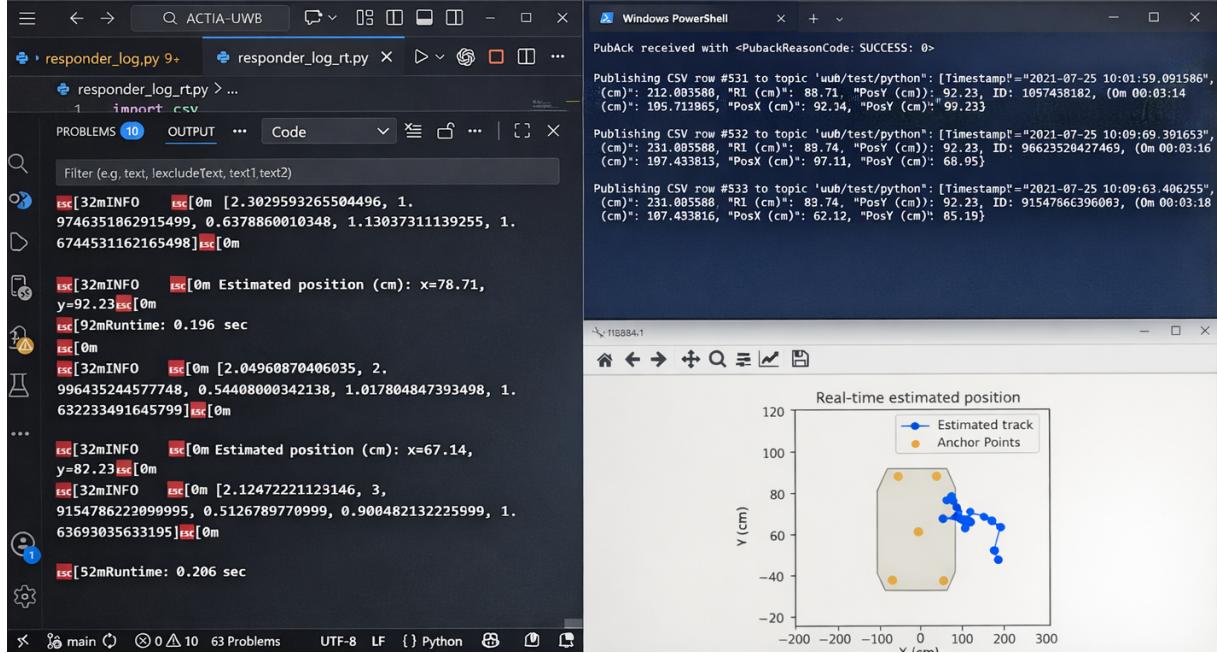
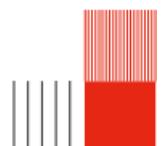


Figure 10.1: Real-time localisation and cloud streaming demonstration.



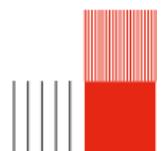
Chapter 11

Objective 05 - BLE/UWB Integration

This final objective aimed at validating a first end-to-end integration between the BLE and UWB components, in order to obtain a coherent operational chain suitable for a digital key workflow: BLE as a low-power discovery and authentication entry point, followed by a controlled handover to UWB for fine-grained ranging and localisation close to the vehicle.

However, the BLE development boards were received only one week before the end of the project. In addition, the technical alignment meeting with NXP to clarify the recommended integration procedure and the steps required to make both boards operate together was only possible two days before the final deadline. This late availability significantly constrained the time window for implementation, debugging, and validation under realistic conditions, which made the full completion of this objective challenging within the project schedule.

Despite these constraints, the work was not left incomplete: we consolidated the integration knowledge into a structured and actionable guide intended for the next team continuing the project. This guide focuses on reducing the initial ramp-up time by covering the key elements needed to start quickly: hardware setup and flashing prerequisites, expected BLE–UWB workflow and handover logic, recommended software structure, common pitfalls encountered during first bring-up, and a suggested test procedure to validate connectivity before moving to full localisation and cloud streaming. As a result, the next project iteration can avoid losing time on environment comprehension and can directly focus on implementation and system-level validation.



Chapter 12

Conclusion

This project prototyped and evaluated a secure BLE/UWB keyless-entry solution using a smartphone as a digital key, with an engineering focus that goes beyond basic ranging: robust localisation in realistic vehicle conditions, secure cloud-based monitoring and traceability, real-time operation with live interpretation, and an energy assessment for parked-vehicle constraints.

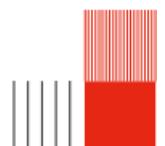
On the localisation side, we improved an initial pipeline that was sensitive to initialization and could become unstable under noisy UWB measurements. Position estimation was reformulated as a nonlinear least-squares problem, solved with a dedicated least-squares strategy and warm-started from the previous state to exploit motion continuity. Robustness was reinforced through measurement validation (handling invalid/incomplete samples) and optional weighting to reduce the influence of less reliable ranges. Compared to the previous approach, the resulting trajectories are smoother and more physically consistent on representative datasets, with fewer unrealistic jumps during challenging phases such as approaching and entering the vehicle.

To support experimentation and traceability, we implemented a secure cloud backend using AWS IoT Core (MQTT over TLS with X.509 authentication) and DynamoDB persistence via IoT Rules. The architecture was validated end-to-end and designed to remain unchanged when replacing development-time data sources with live embedded ranging outputs (same topics, routing logic, and database schema). Beyond offline processing, we operationalized the pipeline by enabling real-time position computation and producing a continuous time-series output that can be streamed to the cloud while also supporting live visualisation of proximity and trajectory evolution during test sessions.

Energy was addressed as a key deployment constraint. Using an oscilloscope-based shunt-resistor method, we measured higher average current when UWB ranging is enabled (initiator: ~ 92.3 mA idle vs. ~ 109.3 mA active; responder: ~ 116 mA at 20 cm vs. ~ 162 mA at 5 m). A system-level model (five vehicle-side nodes) was then used to derive order-of-magnitude parked-autonomy bounds for a 12 V starter battery under conservative usable-charge assumptions, highlighting that continuous operation must be carefully budgeted (about 5.36 days in the lower-power case vs. about 3.58 days in the continuous-UWB case). These results motivate mitigation strategies such as duty-cycling, BLE-triggered activation, adaptive ranging rate, and power-supply options including an auxiliary battery and complementary rooftop PV harvesting.

Finally, full BLE/UWB end-to-end integration was constrained by late reception of the BLE boards and a late technical alignment meeting with NXP. While a complete validated BLE discovery/authentication \rightarrow UWB ranging chain could not be finalized within the available time, we consolidated the integration knowledge into a structured guide (setup prerequisites, workflow and handover logic, common pitfalls, and test procedure) to accelerate the next project iteration.

Overall, the project delivers a measurable end-to-end prototype connecting robust localisation, secure cloud logging, real-time interpretation, and energy-aware deployment reasoning. Future work should prioritize full embedded integration without host-PC dependencies, systematic ground-truth validation across diverse vehicle/NLoS conditions, and rigorous power-management policies to reconcile user experience with long parked-vehicle autonomy.



Bibliography

- [1] Jane Smith. *Une étude sur quelque chose*. Journal of Something, 29(1):10–20, 2023.
- [2] John Johnson. *Tout sur rien*. Editeur XYZ, 2024.
- [3] Jane Doe. *Comment faire des liens dans LaTeX*. Online, 2025. Disponible sur Overleaf.
- [4] Eduardo Garcia-Espinosa, Omar Longoria-Gandara, Ivan Pegueros-Lepe, and Alfredo Veloz-Guerrero. Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications. *Electronics*, 7(12):386, 2018. DOI: 10.3390/electronics7120386.
- [5] Sophia Bauknecht, Florian Wätzold, Anton Schlösser, and Julia Kowal. Comparing the Cold-Cranking Performance of Lead-Acid and Lithium Iron Phosphate Batteries at Temperatures below 0 °C. *Batteries*, 9(3):176, 2023. DOI: 10.3390/batteries9030176.
- [6] ZVEI – German Electrical and Electronic Manufacturers’ Association. *Lead-acid starter batteries in systems: Usage information*. Information leaflet 28e, ZVEI Batteries Association (Working Group Starter Batteries), Oct. 2017.
- [7] U.S. Department of Energy. *DOE-HDBK-1084-95: Primer on Lead-Acid Storage Batteries*. 1995.
- [8] Miguel Franco Rius. *Influence of Shallow Cycling on the Ageing of SLI Batteries*. Master’s thesis, Universitat Politècnica de Catalunya (UPC), 2010.
- [9] VARTA Automotive. *Start-stop myths*. Web page. Accessed: 2026-01-16.
- [10] BioLogic Learning Center. *What are SoC and SoH of a battery, how to measure them?* Web page, 2024. Accessed: 2026-01-16.
- [11] Texas Instruments. *TPSM33620-Q1: 3 V to 36 V Input, 1 V to 7 V Output Automotive Synchronous Buck Converter Power Module (Product Page)*. Web page. Accessed: 2026-01-16.
- [12] Hendrik Schub, Patrick Plötz, and Frances Sprei. Electrifying company cars? The effects of incentives and tax benefits on electric vehicle sales in 31 European countries. *Energy Research & Social Science*, 120:103914, 2025. DOI: 10.1016/j.erss.2024.103914.
- [13] Christian Schuss, Tapio Fabritius, Bernd Eichberger, and Timo Rahkonen. Calculating the Output Power of Photovoltaic Cells on Top of Electric and Hybrid Electric Vehicles. In *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, Auckland, New Zealand, 2019. IEEE. DOI: 10.1109/I2MTC.2019.8826890.
- [14] Sajib Chakraborty, Hai-Nam Vu, Mohammed Mahedi Hasan, Dai-Duong Tran, Mohamed El Baghdadi, and Omar Hegazy. DC-DC Converter Topologies for Electric Vehicles, Plug-in Hybrid Electric Vehicles and Fast Charging Stations: State of the Art and Future Trends. *Energies*, 12(8):1569, 2019. DOI: 10.3390/en12081569.
- [15] Ömer Tan, Daniel Jerouschek, Ralph Kennel, and Ahmet Taskiran. Energy Management Strategy in 12-Volt Electrical System Based on Deep Reinforcement Learning. *Vehicles*, 4(2):621–638, 2022. DOI: 10.3390/vehicles4020036.
- [16] GS Yuasa. *Auxiliary Batteries: GS Yuasa E-Learning Support Documentation*. PDF, 2019. Accessed: 2026-01-16.

