



Research article

Design and engineering validation of a Bluetooth low energy indoor asset tracking system with ESP32-C6 wireless gateways for multi-story buildings

Xudong Gao*

Faculty of Information Technology, Monash University, Melbourne, Australia

* **Correspondence:** Email: gaoxudong61@gmail.com.

Abstract: Indoor asset tracking in multi-story buildings is difficult because low-cost radio signals are affected by attenuation, multipath propagation, human blockage, dense 2.4 GHz interference, and floor-to-floor ambiguity. This paper presents the design and engineering validation of a low-cost tracking system based on Bluetooth Low Energy asset tags and ESP32-C6 wireless gateways. In the prototype, beacons attached to assets broadcast identifiers, fixed transponders scan advertisements and record received signal strength indicator (RSSI) values, and a wireless/cloud backhaul stores observations for a web dashboard and historical CSV logging. The evaluation focuses on floor-level and area-level asset awareness rather than statistically verified coordinate-level positioning. Experimental observations show strong adjacent detection (−20 dBm), a reduced signal strength at 1 m (−68 dBm), packet loss around 10 m under noisy conditions (−94 dBm), and a weak maximum observed detection at approximately 28 m under favorable line-of-sight conditions. Vertical testing showed no signal through a single floor slab at approximately 3.9 m separation and only occasional weak detection around −92 dBm at approximately 2.34 m separation. Configuration tests show differences in the observed RSSI values across the transmit power, strength code, and internal capacitor code settings, with Config 2 and Config 4 providing the strongest observed RSSI in several scenarios. These observations do not establish detection stability or isolate the physical effects of the device-specific codes. The results support a preliminary deployment strategy based on per-floor gateways, conservative receiver spacing, and site-specific beacon configuration.

Keywords: indoor asset tracking; Bluetooth low energy; ESP32-C6 gateway; RSSI; sensor system

1. Introduction

Indoor asset tracking is an increasingly important function in smart buildings, healthcare facilities, warehouses, laboratories, universities, and industrial workplaces. Shared mobile assets such as carts, projectors, tools, test instruments, mobile medical devices, and inventory containers are frequently moved between rooms and floors by different users. When the asset locations are not visible, the cost is not only the replacement value of the asset but also the time spent searching, the delay in operations, and the uncertainty created for facility managers. A practical tracking system does not always need centimeter-level accuracy. In many building-management cases, users first need to know the correct floor and approximate area. Hightower and Borriello [1] discussed location systems for ubiquitous computing, while Zafari et al. [2] surveyed indoor localization technologies. These studies motivate interest in low-cost wireless systems, especially where dense ultra-wideband anchors, cameras, or expensive custom infrastructures cannot be justified.

Outdoor localization benefits from satellite navigation, but indoor environments remain difficult because building materials attenuate signals and create non-line-of-sight propagation. Moreover, radio-frequency signals reflect from walls, metal objects, furniture, and people, producing multipath behaviors and temporal fluctuations. These difficulties are amplified in multi-story buildings. A radio observation from the wrong floor may occasionally appear stronger than an observation from the correct floor because of antenna orientation, temporary blockage, stairwell propagation, reflections, or device placement. For asset tracking, cross-floor ambiguity is particularly problematic. A system that displays an asset on the wrong floor can be worse than no system at all because it directs the user to the wrong search area.

Bluetooth low energy (BLE) is a strong candidate for low-cost asset tracking because small battery-powered beacons can periodically broadcast identifiers with limited energy use. BLE hardware is inexpensive, widely available, and supported by microcontrollers that can also provide wireless connectivity [3]. However, a BLE received signal strength indicator (RSSI) measurements are not stable distance estimates. Even at similar physical distances, an RSSI can vary because of the antenna orientation, casing material, human movement, obstruction, and interference from other 2.4 GHz devices. Therefore, a deployable BLE system must be evaluated as a practical communication and sensing system rather than as a perfect geometric measurement device.

Wi-Fi is also widely used for indoor localization, especially in fingerprinting systems [4]. However, in the system studied in this paper, Wi-Fi primarily acts as the backhaul from fixed ESP32-C6 transponders to the cloud/backend. This distinction is important. The system should not be described as a full BLE-Wi-Fi positioning-fusion algorithm unless a Wi-Fi RSSI, Wi-Fi fingerprints, channel state information, or access-point features directly participate in the location estimate (see the fusion studies discussed in Section 2). Therefore, the system is described as a BLE-based indoor asset tracking system with ESP32 Wi-Fi gateways. This terminology is more accurate and better aligned with the experimental evidence, where the validated sensing data are BLE RSSI measurements and the validated infrastructure role of Wi-Fi is data upload and backend connectivity.

The contribution of this paper is an engineering validation of a low-cost asset tracking prototype. The paper presents the system architecture, hardware and software implementation, and practical test results for the detection range, signal behavior under interference, vertical penetration through floors, beacon configuration effects on the RSSI, and cloud/dashboard integration. The evaluation is intended for applied sensor-system development: it explains what the prototype can reliably support,

what environmental conditions limit deployment, and what data are still required before coordinate-level accuracy or classification percentages can be formally claimed.

The remainder of this paper is organized as follows: Section 2 reviews related work and explains the position of the prototype among BLE, Wi-Fi, and hybrid indoor localization methods; Section 3 describes the system architecture and design rationale; Section 4 presents the hardware and software implementation, including the model information drawn from the project documentation; Section 5 describes the experimental methodology and evaluation scope; Section 6 presents the available results; Section 7 discusses deployment implications, limitations, and future improvements; and Section 8 concludes the paper.

2. Related work and technical background

Indoor localization has been studied for more than two decades. Early work emphasized the general challenge of locating users and devices in ubiquitous computing environments, while later research developed radio fingerprinting, proximity detection, inertial sensing, ultrasound, infrared, ultra-wideband, magnetic field sensing, visual methods, and multi-sensor fusion. Each technology has different advantages. Ultra-wideband can achieve a high accuracy, but its anchor and deployment procedures may be unsuitable for small or cost-sensitive projects. Vision and light detection and ranging (LiDAR) can provide rich environmental information but may be expensive and may raise privacy or maintenance concerns. Wi-Fi is widely available, but its RSSI is coarse and varies strongly with the building layout. BLE offers a useful compromise for asset tracking because beacons are small, inexpensive, and energy efficient.

BLE indoor localization is commonly based on proximity, trilateration, fingerprinting, or filtering. In a proximity approach, the location is associated with the nearest or strongest receiver. This is simple and robust for area-level asset tracking, but it does not provide precise coordinates. Trilateration attempts to infer position from distances estimated from the RSSI, but RSSI-to-distance conversion is unreliable indoors because the path-loss exponent changes across spaces and because small body movements or antenna rotations can change the observed signal. Fingerprinting can improve the accuracy by collecting reference RSSI patterns throughout the building, but it requires a site survey and maintenance when the environment changes. Faragher and Harle [5] studied location fingerprinting using BLE beacons. Filtering approaches, such as Kalman or particle filters, can smooth observations, but they require assumptions about motion, noise, and update frequency. Azaddel et al. [6] proposed an asynchronous Wi-Fi/BLE fusion method based on particle filtering.

Wi-Fi fingerprinting has been widely used because many buildings already contain access points. He and Chan [4] reviewed advances in Wi-Fi fingerprint-based indoor positioning. A typical Wi-Fi fingerprinting system collects RSSI patterns at known reference points and later compares online measurements with the stored radio map. More advanced approaches use channel state information, probabilistic models, or learning-based classifiers. However, Wi-Fi fingerprinting requires a radio map and may be affected by infrastructure changes, access point replacement, user-device heterogeneity, and crowd density. Kanaris et al. [7] fused Bluetooth beacon observations with Wi-Fi radio maps. Leitch et al. [8] examined indoor localization using Wi-Fi, BLE, UWB, and IMU technologies. Zhang et al. [9] proposed privacy-preserving localization using fused Wi-Fi and Bluetooth fingerprints. Hu et al. [10] reported practical indoor localization experience in shopping malls. Jin et al. [11] investigated BLE-beacon positioning together with pedestrian dead reckoning for

smartphones. These studies address complementary indoor localization problems; combining multiple modalities can improve the robustness, but it may increase the calibration and algorithmic complexity.

Multi-story localization introduces a special challenge because floor-level correctness is a categorical decision. The vertical position cannot be reliably inferred from a BLE RSSI alone. A beacon may weakly leak through a floor slab or be detected through stairwells and open spaces, but dense concrete can block or strongly attenuate the signal. Fetzer et al. [12] studied barometer-based floor assignment, while Marjasz et al. [13] addressed vertical positioning in multi-building environments. Zhou et al. [14] investigated multi-floor localization based on multiple sensor modalities. In an asset tracking system where the asset does not necessarily carry a smartphone, barometers and inertial sensors may not be available. A practical alternative is to deploy fixed receivers per floor and use conservative floor assignment rules that prefer strong, recent, and locally consistent observations.

The present prototype belongs to the engineering-oriented part of this research space. It does not attempt to compete with advanced full-building fingerprinting or Ultra-Wideband (UWB) systems. Instead, it asks whether a small number of BLE tags and ESP32-C6 transponders can support useful indoor asset awareness when connected to a cloud dashboard. The system-level contribution is the integration of inexpensive beacons, receiver placement, Wi-Fi data upload, historical logging, and user interface design. The experimental contribution is a set of range, RSSI, configuration, and vertical-penetration measurements that can guide deployment decisions.

A key lesson from related work is that terminology must be used carefully. A system that uses BLE RSSI for sensing and Wi-Fi only for network communication should not be presented as a Wi-Fi/BLE localization-fusion system in the same sense as systems that fuse BLE RSSI with Wi-Fi RSSI, Wi-Fi fingerprints, or Wi-Fi channel state information. In this paper, the terms Wi-Fi gateway and Wi-Fi backhaul are preferred. The contribution is a BLE-based asset tracking architecture whose gateways use Wi-Fi for data transmission and cloud integration.

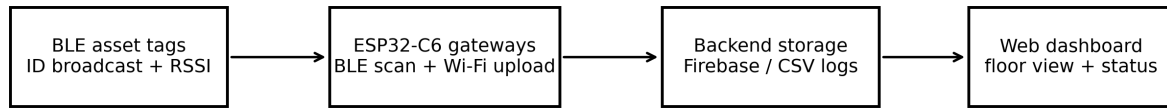
3. System architecture and design rationale

The system is organized as a four-layer architecture: a sensing layer, a gateway layer, a data layer, and an application layer. The sensing layer consists of battery-powered BLE beacons attached to assets. Each beacon broadcasts an identifier that can be used to distinguish assets. The gateway layer consists of ESP32-C6 devices placed at fixed locations in the building. These devices scan BLE advertisements, record RSSI, attach a gateway identifier and timestamp, and upload the observation through Wi-Fi. The data layer stores the latest observations and historical records. The application layer displays assets in a web interface, allowing users to view the floor-level status and last-seen information.

The design goal is not to create a dense high-precision tracking network. Instead, the system targets low-cost deployment in environments where the approximate location and floor-level awareness are sufficient. This goal affects several design decisions. First, the asset tags are intentionally simple. They broadcast identifiers and do not perform expensive local computation. Second, fixed gateways carry the responsibility for scanning and communication. This allows the asset tag to remain low power while the gateway uses Wi-Fi to connect to the backend. Third, the user interface emphasizes floor-level grouping and asset status rather than displaying precise coordinates that have not been validated by a coordinate-level ground-truth experiment.

Figure 1 summarizes the end-to-end architecture. BLE asset tags broadcast identifiers, ESP32-C6

transponders detect the signals and read the RSSI, the Wi-Fi/cloud backhaul stores the observations, and the web dashboard presents current status. The system supports the operational question that is most important for many asset management tasks: which floor or approximate area is the asset currently associated with, and how recent is that observation?



End-to-end BLE sensing and Wi-Fi/cloud data pipeline

Figure 1. End-to-end architecture of the BLE-based asset tracking system with ESP32-C6 Wi-Fi gateways.

Table 1. Main system components and functions.

Layer	Component	Role in the prototype
Sensing layer	BLE asset tag	Broadcasts an asset identifier; RSSI is read by fixed transponders
Gateway layer	ESP32-C6 transponder	Scans BLE advertisements and uploads observations through Wi-Fi
Data layer	Firebase/CSV logging	Stores latest asset observations and history records
Application layer	Web dashboard	Shows floor-level asset association and last-seen status

3.1. Floor-aware deployment strategy

A multi-story building requires explicit floor-aware deployment. The vertical tests presented in Section 6 show that full floor slabs can strongly block the BLE signals. Therefore, the design assumes that each relevant floor should contain at least one receiver, and larger floors may require multiple receivers depending on the coverage, room density, and user expectations. This receiver-per-floor strategy is not only about improving the signal strength; it also reduces ambiguity by associating each observation with a known physical floor.

In the documented prototype, observations are associated with each transponder's known floor and area, and recent per-transponder records support the dashboard display and historical logging. The available implementation description does not specify a unique asset-level rule for simultaneous detections or conflicting floor observations. Therefore, most-recent selection, strongest-RSSI selection, and confidence scoring are possible extensions, not verified implemented rules. The displayed association should be interpreted as receiver-linked observation information rather than a validated resolution of competing floor assignments.

The interface includes an active/inactive status concept, but the available records do not specify its timeout or the handling of stale records. A last-seen location is historical evidence, not confirmation of current presence; additionally, a missed scan does not establish that an asset has left. Explicit expiration thresholds and conflict handling require implementation-level confirmation before deployment.

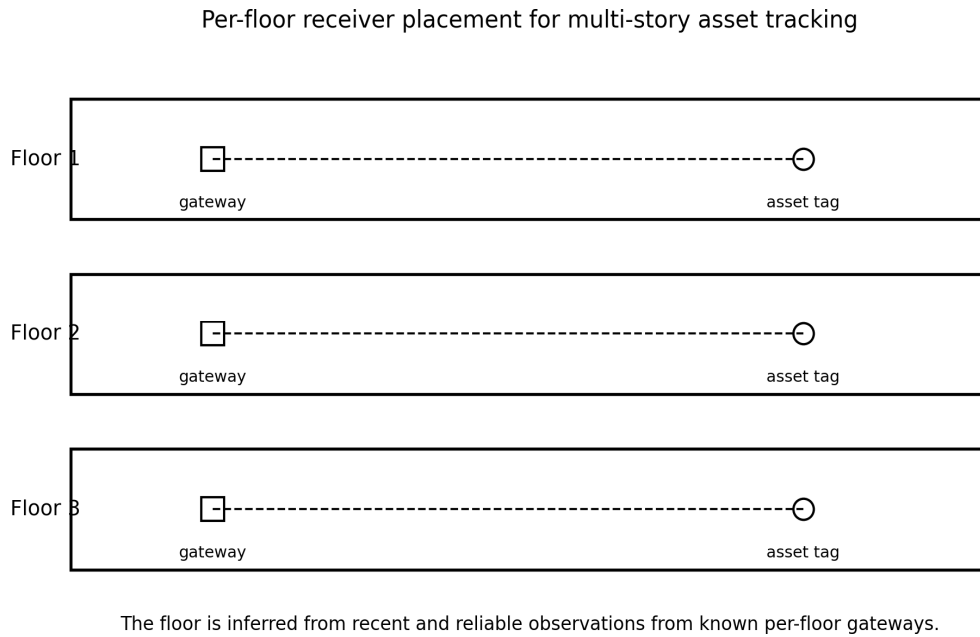


Figure 2. Floor-aware deployment concept using at least one receiver on each monitored floor.

3.2. Data model and communication pipeline

Each BLE observation can be represented by the tuple: asset identifier, transponder identifier, RSSI, timestamp, and optional metadata such as the floor number or transponder location. The transponder sends this observation through Wi-Fi to the backend. The backend can maintain two categories of data: the latest state for dashboard display and the historical log for a later analysis. In the project implementation, Firebase was used for real-time cloud storage, and CSV export/history functionality was developed so that existing Firebase data could be converted into tabular records and later observations could be retrieved into the CSV format.

This data pipeline is deliberately simple. It avoids requiring each browser client to directly access low-level device messages. Instead, transponders push observations to the backend, and the dashboard reads cleaned asset states. Such separation improves the maintainability because the device firmware, backend processing, and user interface behavior can be modified independently. Moreover, it allows the system to support alternative logging strategies, such as a direct local-network upload to a server when the cloud latency or Wi-Fi power consumption needs to be reduced.

The available project records include aggregate test values but not a complete public raw log for all experiments. Therefore, the manuscript uses data that can be verified from the project reports: range observations, RSSI values at different distances, vertical-penetration outcomes, and configuration comparison tables. Claims requiring sample-level error distributions or ground-truth coordinate logs are reserved for future controlled validation.

4. Hardware and software implementation

The transponder hardware is based on the ESP32-C6 platform. The project documentation identifies the ESP32-C6 DevKitC as the fixed-location receiver used for the prototype. The ESP32-C6 integrates 2.4 GHz Wi-Fi 6, Bluetooth 5.3, IEEE 802.15.4 support, radio-frequency components,

processors, and power-management functions [3]. The project transponder research lists key specifications including Wi-Fi operation in the 2412–2484 MHz band, a high-performance RISC-V processor up to 160 MHz, a low-power processor up to 20 MHz, 320 KB ROM, 512 KB high-performance SRAM, 16 KB low-power SRAM, Bluetooth LE receiver sensitivity of -106 dBm, and deep-sleep mode current of approximately 7 microamps. These specifications support the selection of the platform for a low-cost gateway that must scan the BLE advertisements and transmit data through Wi-Fi.

The use of ESP32-C6 simplifies integration because a single microcontroller platform can receive BLE advertisements and send data over Wi-Fi. This reduces the need for separate BLE gateway and Wi-Fi modules. The project documentation describes an initial transponder arrangement using the ESP32-C6 DevKitC and a 3AA battery pack, and the project documentation reports a $3 \times$ AA supply of approximately 4.5 V for the prototype. The same documentation also notes that wiring the supply to the correct board input is important: a power connection error to the regulated 3.3 V pin caused unstable behavior until the supply was connected to the 5 V input. This type of implementation detail is important in low-cost wireless systems because an apparent radio instability can sometimes originate from power or wiring faults rather than from the localization method itself.

BLE beacons are used as asset tags. Feasycom FSC-BP104D and MOKO products were examined during component selection. The FSC-BP104D was documented as supporting iBeacon, Eddystone, and AltBeacon formats, Bluetooth 5.1, and IP67 protection. Handover records identify TEL0149 Gravity Sensor BLE beacons, while testing descriptions mention Gravity sensor beacons and a Feasycom long-range beacon. These descriptions do not identify the model used in each reported test; the Feasycom test unit cannot be assumed to be the FSC-BP104D. Table 8 separates documented settings from an unresolved hardware attribution.

The beacon configuration comparisons in Section 6 compare default settings with nine modified configurations. The transmit power is reported in dBm. The terms “strength code” and “internal capacitor code” reproduce labels and numeric settings from the project configuration tables; the available documentation does not identify their manufacturer-defined functions, register mappings, configuration application, or firmware version. These codes are not calibrated physical quantities and must not be interpreted as dBm or capacitance values. Their physical mechanisms and transferability between beacon models remain unverified; the RSSI comparisons do not establish improved stability from either code.

The software implementation evolved through several stages. An early approach used local data posting and CSV logging. In this mode, the ESP32 sent records to a local server, and each record contained fields such as the beacon name, transponder identifier, date, time, and RSSI. Later versions integrated the Firebase Realtime Database for cloud storage and remote visualization [15]. A subsequent development path added the ability to export existing Firebase data to CSV and retrieve the latest data to CSV for history functions. Additionally, the project reports describe a local-network variant of the transponder software that sends data to a local server and writes a CSV file, which can reduce the dependence on the cloud response time in some scenarios.

Firmware power management is an important part of the implementation. The transponder software was redesigned to enter deep sleep between scan periods. The project review reports scanning every 15 minutes, with an adjustable interval. This is the application-level scan schedule, not the BLE scan window or controller scan interval. Firebase real-time synchronization concerns propagation of uploaded records to clients, not a continuous radio observation. The location

information can only change after a successful scan and upload; a movement just after a scan may remain unobserved for nearly 15 minutes, or longer if detection or upload fails. No measured end-to-end update latency is available.

The frontend is a web-based dashboard. Its role is not to perform complex localization calculations but to present operationally useful information: which assets have been observed, which floor or area they are associated with, and when they were last seen. This is appropriate for the current level of validation. Presenting a precise coordinate map would imply a spatial accuracy that the available experimental records do not fully support. Therefore, a dashboard organized by floors and asset status is more honest and more usable for the target prototype.

5. Experimental methodology and evaluation scope

The evaluation uses project experimental records collected during prototype testing. The available data include hardware quality criteria, beacon range testing, vertical-penetration observations, configuration consolidation tests, and software integration notes. These records are sufficient to evaluate the engineering feasibility and communication performance, but they are not sufficient to verify the coordinate-level localization accuracy, a full BLE-only baseline, or statistically robust floor-classification accuracy. Therefore, the evaluation scope is limited to claims supported by the available measurements.

The main experimental questions are as follows; first, what is the approximate communication behavior of the BLE beacon as distance increases; second, how does heavy radio interference affect usable range; third, can BLE signals sufficiently penetrate between floors to allow a single receiver to cover multiple floors; fourth, how do beacon configuration parameters affect RSSI in quiet and noisy conditions; and fifth, does the hardware/software architecture support an end-to-end asset tracking workflow with cloud and CSV logging? These questions match the engineering stage of the prototype.

The range test recorded the RSSI from adjacent placement to approximately 28 m (Table 3). Reports describe radio-noise cases with more than 200 or 500 nearby Bluetooth devices. “Quiet” and “high-interference” are qualitative labels; device counts do not measure the interference power or channel occupancy, and no quiet-test count is specified. The observation durations, repeated sample counts, BLE scan windows and intervals, active/passive mode, mounting heights, antenna orientations, and detailed obstruction geometry are unspecified. RSSI values are observations, not statistical means. The prototype’s 15-minute scan schedule does not establish these experiments’ acquisition settings.

The vertical test placed the beacon and transponder on different levels. One case involved approximately 3.9 m vertical separation through a floor slab, and another involved approximately 2.34 m vertical separation at an intermediate location. These tests are important because multi-story asset tracking depends on floor-level separation. If the BLE signals penetrated strongly between floors, then a naive strongest-RSSI rule could misassign assets. If the floor slabs block the signal, then each floor needs dedicated receiver infrastructure. Therefore, the results inform deployment planning and floor-aware gateway placement.

The configuration tests compared ten beacon settings: a default and Configurations 1 through 9, varying the transmit power, strength code, and internal capacitor code. Tests were performed at approximately 10 m in qualitatively quiet and high-interference environments, and approximately 5 m. The interference category of the 5 m test is not specified. Tables 4–6 reproduce the available

settings and RSSI values in dBm; the per-test beacon model and advertising interval are not identified. These comparisons describe observed RSSI differences, without demonstrating the detection-rate or temporal-stability improvements.

The paper does not report the mean coordinate error, root-mean-square error, median error, cumulative distribution functions, floor identification percentage, or confusion matrices. Such metrics require a controlled experiment that records the true asset coordinates, estimated coordinates, true floor labels, predicted floor labels, sample counts, and baseline outputs. The present evaluation is best understood as an engineering validation of hardware, signal behavior, configuration, and data-pipeline feasibility.

Table 2. Evaluation scope and evidence status.

Evaluation aspect	Available project evidence	Claim supported in this manuscript
Range behavior	RSSI values from adjacent placement to 28 m	Supported
Vertical penetration	One-floor and half-floor separation outcomes	Supported
Configuration impact	RSSI under ten beacon settings across three scenarios	Supported
Cloud/CSV pipeline	Project implementation records	Supported as system functionality
Coordinate-level accuracy	No complete raw ground-truth coordinate log	Not reported
Floor accuracy percentage	No confusion matrix or sample-level labels	Not reported
Improvement over BLE-only baseline	No baseline trial table	Not reported

6. Results

The results show that the prototype can support BLE-based asset detection and backend integration, but they also reveal important deployment constraints. The most important finding is that the RSSI is highly scenario dependent. Strong near-field readings do not guarantee reliable longer-range detection under interference, and vertical separation through a floor can entirely eliminate detection. These findings support the design decision to use per-floor transponders and conservative status presentation.

6.1. Range testing

The range test demonstrates the expected overall weakening of the BLE RSSI with increasing distance, but it also shows local non-monotonic behavior caused by indoor propagation and repositioning. Adjacent placement produced an RSSI of approximately -20 dBm, indicating a very strong signal. At 0.1 m, the RSSI was -48 dBm, at 0.5 m it was -57 dBm, and at 1 m it had fallen to approximately -68 dBm. The signal remained detectable at 2 m, 4 m, 6 m, and 8 m, with measured

RSSI values of -74 dBm, -70 dBm, -79 dBm, and -83 dBm, respectively. At approximately 10 m, packet loss was observed and the RSSI was around -94 dBm. At 11 m, after slight repositioning, a measurement of -86 dBm was observed. A maximum detection distance of approximately 28 m was recorded, but the signal was weak and should be interpreted as a maximum observed detection rather than a reliable operational range.

These values provide a preliminary deployment reference under the tested conditions. Approximately 8–10 m should not be described as a verified stable operating range: packet loss was observed at 10 m, and the detection-rate statistics and observation durations are unavailable. Receiver spacing should be confirmed by a site survey and repeated detection measurements, with additional coverage where needed. The 28 m detection is a best-case observation, not a spacing recommendation.

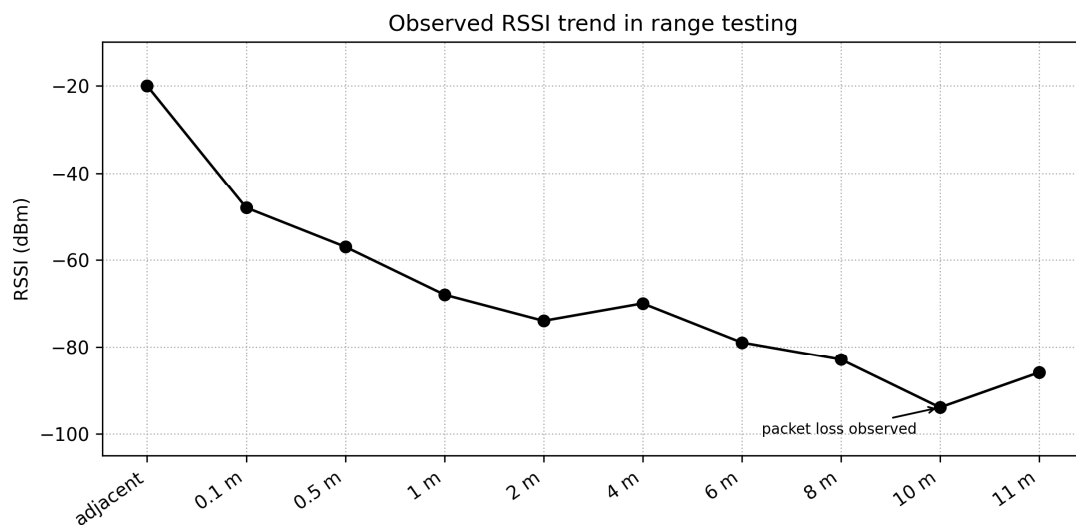


Figure 3. Observed RSSI trend from adjacent placement to 11 m; the 28 m maximum detection is discussed as a weak best-case observation.

Table 3. Range and vertical-separation observations.

Distance or scenario	Measured RSSI / outcome	Notes
Adjacent	-20 dBm	Very strong signal
0.1 m	-48 dBm	Short-range reference
0.5 m	-57 dBm	Short-range reference
1 m	-68 dBm	RSSI already reduced by short distance
2 m	-74 dBm	Detection reported; stability not quantified
4 m	-70 dBm	Detection reported; stability not quantified
6 m	-79 dBm	Detection reported; stability not quantified

Continued on next page

Distance or scenario	Measured RSSI / outcome	Notes
8 m	−83 dBm	Usable but weak
10 m	−94 dBm	Packet loss observed
11 m	−86 dBm	Detected after slight repositioning
28 m	−83 dBm	Maximum observed detection, weak/unstable
One floor below	No signal detected	Approximately 3.9 m vertical separation through a floor slab
Halfway to lower floor	Occasional weak signal around −92 dBm	Approximately 2.34 m vertical separation

6.2. Vertical penetration and multi-floor deployment

The vertical tests provide the strongest evidence for using per-floor transponders. When the beacon and transponder were separated by one floor, with approximately 3.9 m vertical distance through a floor slab, no signal was detected. At an intermediate position with approximately 2.34 m vertical separation, only occasional weak detection around −92 dBm was observed. This result indicates that the building floor structure created strong shielding. Additionally, it means that a single transponder should not be expected to reliably monitor multiple floors.

From a localization perspective, this finding has two implications. First, multi-floor asset tracking should be designed as a floor-specific receiver network. A receiver on each floor provides a physical anchor for floor-level association. Second, if a weak cross-floor signal is occasionally detected, the backend should treat it cautiously rather than immediately reassigning the asset. A robust floor decision rule should consider recency, RSSI strength, persistence over multiple scans, and known receiver placement.

6.3. Configuration testing

The configuration tests show that beacon parameters strongly affect the RSSI. At approximately 10 m in a high-interference environment, the default configuration produced −90 dBm. Config 1, with a transmit power of 0 dBm, a strength code of 24, and a capacitor code of 8, improved the RSSI to −85 dBm. Config 2, with a transmit power of +4 dBm, a strength code of 16, and a capacitor code of 8, reached −82 dBm. Config 4, with a transmit power of +4 dBm, a strength code of 24, and a capacitor code of 11, also reached −82 dBm. Low-power and weaker configurations performed poorly; Config 6, using a transmit power of −20 dBm, was only reported around −100 dBm at approximately 6 m, and Config 9 reached −95 dBm.

At approximately 10 m in a quieter environment, Config 2 had the strongest observed RSSI (−76 dBm), followed by Config 1 (−78 dBm) and Config 4 (−80 dBm), compared with −90 dBm for the default. Thus, Config 1, not Config 4, ranks second in this scenario. At approximately 5 m, Config 2 reached −67 dBm, Config 4 reached −70 dBm, Config 1 reached −75 dBm, and the default reached −80 dBm. These rankings only concern the observed RSSI, not demonstrated detection stability.

The results support scenario-specific RSSI comparisons rather than a universally best

configuration. Config 2 was strongest in the quiet 10 m and 5 m tests and tied with Config 4 in the high-interference 10 m test. Moreover, Config 1 exceeded the default without increasing the transmit power. At 5 m, Config 2 and the default share capacitor code 8; their difference is transmit power. Config 4 changes all three settings relative to the default, so its result does not isolate a capacitor-code effect. Neither a requirement for a higher capacitor code nor a robustness advantage is established. Configuration selection requires further detection-rate, temporal-variability, and battery measurements.

Table 4. Beacon configuration testing at approximately 10 m in high-interference conditions.

Config	Tx power (dBm)	Strength code	Capacitor code	RSSI at 10 m high-interference
Default	0	16	8	−90
1	0	24	8	−85
2	4	16	8	−82
3	0	16	11	−86
4	4	24	11	−82
5	0	12	8	−90
6	−20	16	8	−100 at approximately 6 m
7	−9	16	8	−93
8	0	16	6	−92
9	−9	12	6	−95

Table 5. Beacon configuration testing at approximately 10 m in quieter conditions.

Config	Tx power (dBm)	Strength code	Capacitor code	RSSI at 10 m quiet
Default	0	16	8	−90
1	0	24	8	−78
2	4	16	8	−76
3	0	16	11	−87
4	4	24	11	−80
5	0	12	8	−88
6	−20	16	8	−97
7	−9	16	8	−90
8	0	16	6	−84
9	−9	12	6	−93, inconsistent

Table 6. Beacon configuration testing at approximately 5 m.

Config	Tx power (dBm)	Strength code	Capacitor code	RSSI at 5 m
Default	0	16	8	−80
	0	24	8	−75
2	4	16	8	−67
3	0	16	11	−77
4	4	24	11	−70
5	0	12	8	−78
6	−20	16	8	−92
7	−9	16	8	−82
8	0	16	6	−80
9	−9	12	6	−84

Table 7. Summary of configuration implications.

Scenario	Strongest observed RSSI	Comparison supported	Deployment implication
Approximately 10 m, quiet	Config 2 (−76 dBm); Config 1 next (−78 dBm)	Config 4 is third (−80 dBm)	Prioritize site-specific RSSI testing; stability is unverified.
Approximately 10 m, high interference	Config 2 and Config 4 (−82 dBm)	Config 1 also exceeds default 82 dBm)	Candidates for further detection-rate and battery testing.
Approximately 5 m	Config 2 (−67 dBm); Config 4 next (−70 dBm)	No isolated capacitor-code benefit established	Compare detection rate and power use before selection.

Table 8. Hardware attribution and settings available for the reported tests.

Reported test	Beacon model attribution	Available settings
Range (Table 3)	Not identified per test	Tx power, codes and advertising interval unspecified
Vertical (Table 3)	Not identified per test	Tx power, codes and advertising interval unspecified
10 m quiet / high interference (Tables 4–5)	Not identified per test	Ten settings tabulated; advertising interval unspecified
5 m (Table 6)	Not identified per test	Ten settings tabulated; advertising interval unspecified

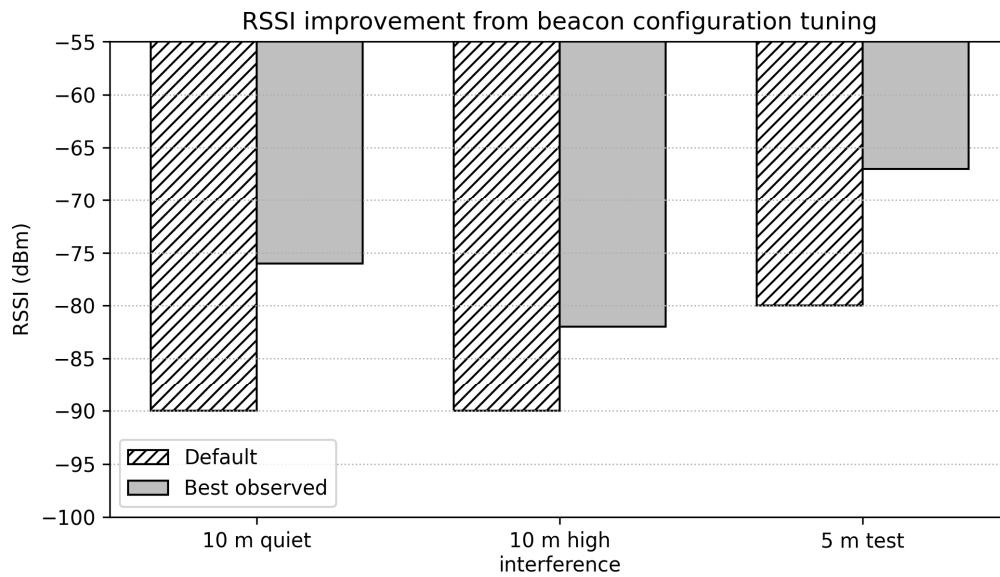


Figure 4. Representative RSSI improvement from beacon configuration tuning.

6.4. End-to-end system integration

The software and data pipeline were sufficiently integrated to support a functional prototype. The project records show that existing Firebase data were exported to CSV and that functionality was developed to retrieve the latest data to CSV. A local-network variant of the transponder software was also developed to send data to a local server and write to a CSV file, reducing dependence on the cloud response time in some scenarios. These implementation details matter because an asset tracking system is only useful if observations can be retained, displayed, and reviewed.

The web interface provides a practical visualization layer. Rather than overstating precision, the interface groups the assets by floor and shows the last-seen status. Additionally, the historical CSV mechanism creates a path for future analysis, such as calculating detection rates, asset dwell time, and movement patterns. Future work should ensure that CSV logs include all fields required for formal evaluation: asset identifier, transponder identifier, raw RSSI, filtered RSSI, timestamp, ground-truth floor, ground-truth coordinate, estimated floor, estimated coordinate, and algorithm version.

7. Discussion

The evidence supports a practical but limited conclusion. The prototype is a feasible low-cost system for BLE-based indoor asset monitoring with Wi-Fi/cloud backhaul. It provides a foundation for floor-level and area-level tracking, but the available project records do not support strong coordinate-level accuracy claims. This distinction is central to the academic reliability of the paper. A system can be useful without claiming a precise coordinate error when its evidence mainly supports the communication range, vertical attenuation, and deployment behavior.

The range results show why BLE tracking must be deployed conservatively. A maximum detection event at 28 m does not mean the system should be designed with 28 m spacing between receivers. Under heavy interference, packet loss appeared around 10 m, which suggests that reliable receiver spacing should be based on worst-case or normal operating conditions rather than best-case

detection. In real buildings, people, furniture, walls, and other devices will change the signal environment. Therefore, a production deployment should include a site survey before the receiver placement is finalized.

The vertical-penetration results are highly relevant to multi-story tracking. No signal was detected through a full floor slab in the tested scenario, while only occasional weak detection appeared at partial vertical separation. This suggests that floor slabs can help reduce cross-floor confusion, but it also means that each floor requires receiver coverage. The system should not rely on vertical leakage for coverage. A per-floor receiver model is more robust and easier for users to understand.

The configuration tests demonstrate that radio tuning is a major part of the system performance. In academic descriptions of indoor localization, the algorithm design often receives the most attention. In engineering prototypes, however, hardware settings can determine whether sufficient observations exist for any algorithm to work. A poor configuration may lead to a weak RSSI, unstable detection, or excessive packet loss. Conversely, stronger settings improve the signal quality but may reduce the battery life. This trade-off should be explicitly handled in the deployment documentation.

The strongest limitation of the current evidence is the absence of sample-level raw localization data. To claim a verified average horizontal error, the experiment must record a set of reference positions, the estimated position for each trial, the error for each trial, and an overall statistical summary. To claim the floor-identification accuracy, the experiment must record true floor labels, predicted floor labels, total sample count, errors, and preferably a confusion matrix. To claim percentage improvement over a baseline, the baseline method must be defined and evaluated on the same test cases. Those materials were not available in the project records used for this study, so such claims are excluded from the manuscript.

This limitation does not make the project unusable. It changes the paper type. Instead of a high-precision localization algorithm paper, the work is best positioned as an engineering validation paper. It describes a low-cost architecture, identifies deployment constraints, and reports practical signal behavior. Such papers can be valuable for applied sensor and Internet of Things (IoT) venues if the claims are calibrated to the evidence.

For future work, a controlled validation protocol should be added. A recommended protocol would place ground-truth points on each floor, collect at least 30 RSSI observations per point under multiple occupancy conditions, evaluate the chosen floor assignment rule, and report the mean error, median error, 90th percentile error, root-mean-square error, floor accuracy, confusion matrix, and stale-detection rate. The experiment should compare at least three baselines: nearest transponder by strongest RSSI, BLE-only area assignment, and the proposed floor-aware filtered method. If the Wi-Fi RSSI or Wi-Fi fingerprints are added as true localization features, then the paper could legitimately use the term BLE-Wi-Fi fusion.

The data pipeline should also be strengthened for reproducibility. Every observation should be logged with a unique experiment ID, asset ID, transponder ID, timestamp, raw RSSI, filtered RSSI, true floor, estimated floor, true coordinate, estimated coordinate, and device configuration. This would allow future reviewers to verify the claims and would enable publication of anonymized data with the manuscript. A small amount of well-structured data is more valuable than a large number of informal observations.

The power behavior requires direct measurement. The ESP32-C6 and BLE beacon specifications indicate low-power potential, but the system-level battery life depends on the scan

duration, upload frequency, Wi-Fi connection time, beacon advertising interval, transmit power, and sleep-mode implementation. A future paper should include current measurements for active scanning, Wi-Fi upload, sleep, and beacon advertising under each configuration. This would make it possible to compute the battery-life estimates and justify deployment recommendations.

Security and privacy also require more attention before production deployment. Asset tracking data can reveal building operations, equipment usage, or staff behavior. The backend should enforce authentication, role-based access, encrypted communication, secure device credentials, and data retention rules. The current prototype demonstrates functionality, but a production version should include a formal threat model and security controls.

8. Conclusions

This paper presented the design and engineering validation of a BLE-based indoor asset tracking system with ESP32-C6 Wi-Fi gateways for multi-story buildings. The system combines BLE asset tags, ESP32-C6 transponders, Wi-Fi/cloud data handling, CSV logging, and a web dashboard. The manuscript focused on evidence that can be supported by the available project records: range behavior, interference sensitivity, vertical signal penetration, beacon configuration effects, and end-to-end data-pipeline feasibility.

The experimental observations showed that the BLE RSSI strongly decreased with distance, packet loss appeared around 10 m in noisy conditions, and a maximum detection of approximately 28 m was only observed as a weak and potentially unstable signal. Vertical testing showed no signal through a one floor slab and only an occasional weak signal at partial vertical separation, supporting a per-floor transponder deployment strategy. Configuration testing showed differences in the observed RSSI across the tested settings, with Config 2 strongest in the quiet 10 m and 5 m tests and tied with Config 4 in the high-interference 10 m test. These readings do not demonstrate improved stability or robustness.

The resulting contribution is a practical low-cost system design and an evidence-based discussion of range, interference, floor separation, and configuration trade-offs. The system is better suited to an applied engineering or sensor-systems venue than to a high-precision localization algorithm venue. Future work should collect complete raw logs and ground-truth position data so that the coordinate-level accuracy, floor accuracy, and baseline improvements can be rigorously evaluated.

Author contributions

X.G. prepared the manuscript, integrated the available experimental data, organized the system design and test evidence, and wrote the engineering validation paper.

Use of Generative-AI tools declaration

The author declares the use of ChatGPT (OpenAI) to assist with manuscript formatting, English-language proofreading, and reference organization. AI tools used: ChatGPT (OpenAI). How was the AI tool used? Formatting and layout assistance, language proofreading, and citation-order checking. Where in the article is the information located? Language and formatting throughout the manuscript, particularly the literature discussion and references. No new experimental measurements or results were generated using AI tools.

Acknowledgments

The author acknowledges the Z1332 Asset Location System project team and project supervisors for their contributions to the project context, prototype implementation discussions, and testing records used in this study.

Conflict of interest

The author declares no conflict of interest.

References

1. Hightower J, Borriello G (2001) Location systems for ubiquitous computing. *Computer* 34: 57–66. <https://doi.org/10.1109/2.940014>
2. Zafari F, Gkelias A, Leung KK (2019) A survey of indoor localization systems and technologies. *IEEE Commun Surv Tutor* 21: 2568–2599. <https://doi.org/10.1109/COMST.2019.2911558>
3. Espressif Systems (2024) ESP32-C6 Datasheet. Available from: https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf.
4. He S, Chan SHG (2016) Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Commun Surv Tutor* 18: 466–490. <https://doi.org/10.1109/COMST.2015.2464084>
5. Faragher R, Harle R (2015) Location fingerprinting with Bluetooth Low Energy beacons. *IEEE J Sel Areas Commun* 33: 2418–2428. <https://doi.org/10.1109/JSAC.2015.2430281>
6. Azaddel MH, Nourian MA, ShahHosseini K, Junoh SA, Akbari A (2023) SPOTTER: A novel asynchronous and independent WiFi and BLE fusion method based on particle filter for indoor positioning. *Internet Things* 24: 100967. <https://doi.org/10.1016/j.iot.2023.100967>
7. Kanaris L, Kokkinis A, Liotta A, Stavrou S (2017) Fusing Bluetooth beacon data with Wi-Fi radiomaps for improved indoor localization. *Sensors* 17: 812. <https://doi.org/10.3390/s17040812>
8. Leitch SG, Ahmed QZ, Abbas WB, Hafeez M, Lazaridis PI, Sureephong P, et al. (2023) On indoor localization using WiFi, BLE, UWB, and IMU technologies. *Sensors* 23: 8598. <https://doi.org/10.3390/s23208598>
9. Zhang X, He F, Chen Q, Jiang X, Bao J, Ren T, et al. (2022) A differentially private indoor localization scheme with fusion of WiFi and Bluetooth fingerprints in edge computing. *Neural Comput Appl* 34: 4111–4132. <https://doi.org/10.1007/s00521-021-06815-9>
10. Hu Y, Qian F, Yin Z, Li Z, Ji Z, Han Y, et al. (2022) Experience: Practical indoor localization for malls. In: *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*, 82–93. <https://doi.org/10.1145/3495243.3517021>
11. Jin Z, Li Y, Yang Z, Y. Zhang, Z. Cheng (2023) Real-time indoor positioning based on BLE beacons and pedestrian dead reckoning for smartphones. *Appl Sci* 13: 4415. <https://doi.org/10.3390/app13074415>
12. Fetzer T, Ebner F, Deinzer F, Grzegorzec M (2023) Using barometer for floor assignment within statistical indoor localization. *Sensors* 23: 80. <https://doi.org/10.3390/s23010080>
13. Marjasz R, Grochla K, Polys K (2024) A comprehensive algorithm for vertical positioning in multi-building environments as an advancement in indoor floor-level detection. *Sci Rep* 14: 14034. <https://doi.org/10.1038/s41598-024-64824-9>

14. Zhou G, Xu S, Zhang S, Wang Y, Xiang C (2022) Multi-floor indoor localization based on multi-modal sensors. *Sensors* 22: 4162. <https://doi.org/10.3390/s22114162>
15. Google Firebase, Firebase Realtime Database Documentation. Available from: <https://firebase.google.com/docs/database>.



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)