ECE445 SENIOR DESIGN LABORATORY

Motion Analysis and Trajectory Reconstruction of Smart SoftBall with UWB Positioning

Final Report

By

Team #7

Chenhan Yang (chenhan7)

Yuxing Wu (yuxingw2)

Tianyang Sun (ts30)

Advisor: Timothy Lee

TA: Jianzu Zhang

May 16, 2026

Abstract

This report presents the design, implementation, and verification of a Smart SoftBall trajectory tracking system based on ultra-wideband (UWB) ranging. The final prototype integrates a battery-powered UWB tag inside a ball, four fixed UWB anchors, a sequential two-way ranging (TWR) communication flow, nonlinear three-dimensional position reconstruction, multi-stage filtering, and a Python graphical interface for live and offline trajectory visualization. The system was developed to provide a lower-cost and more portable alternative to camera-based sports tracking systems while preserving useful training information such as position history, trajectory continuity, and motion direction.

The final design differs from the original proposal in two important ways. First, the ranging and localization method was changed from TDoA to TWR to simplify synchronization and make the four-anchor prototype practical with the selected BU01/DW1000 modules. Second, the final validated scope focuses on robust trajectory collection, filtering, and visualization rather than formal landing-point or out-of-bounds judgement. Experimental verification was conducted in a 3.75 m by 3.9 m indoor test field using five reference positions and 75 total measurements. The system achieved X and Y axis RMSE values of 7.73 cm and 6.49 cm, with an average horizontal error of 8.44 cm. The Z axis error was larger (27.33 cm RMSE), which is consistent with the limited vertical separation of the anchor geometry. The prototype operated at a configured 20 Hz update rate, showed stable communication during testing, and provided both real-time delayed display and offline replay modes through the GUI.

Contents

1 Introduction

1.1 Motivation and Problem Statement

Traditional softball, tennis, and similar ball-sport training often depends on human observation or camera-based systems to determine the path and status of a ball. Human judgement is convenient but inconsistent, especially when the object moves quickly or when repeated training statistics are needed. Camera-assisted systems can be accurate, but systems in the style of Hawk-Eye require multiple high-speed cameras, venue calibration, 3D modeling, and fixed installation [1]. This makes them expensive and difficult to deploy for daily training, mobile experiments, or recreational matches.

The project addresses this gap by embedding a UWB tag into a physical ball and using four portable anchors to reconstruct the ball's position. Instead of relying on line-of-sight cameras, the system measures radio ranging data, reconstructs the trajectory in software, filters abnormal points, and presents the result in a GUI. The goal is not to replace professional broadcast systems, but to demonstrate a compact, low-cost tracking platform that can be moved between small fields and used for training analysis.

1.2 Project Objectives and Scope

The primary objective was to build a functional UWB-based tracking prototype. The final system includes a UWB tag embedded inside a test ball, four UWB anchors placed around a measured indoor field, embedded C firmware for UWB ranging and serial output, Python software for filtering and trajectory reconstruction, and a GUI for live acquisition, offline replay, and detailed point inspection.

The validated scope focuses on trajectory acquisition and reconstruction. The system supports a real-time display mode with an approximately 1 s delay buffer, and an offline mode that reprocesses saved raw RANGE4D CSV data with stronger filtering. Formal landing-point and out-of-bounds judgement were left outside the final verified scope; the GUI can still visualize projected positions against a configured field boundary, which provides the software foundation for future boundary logic.

1.3 High-Level Requirements

1. The embedded UWB tag and anchor hardware shall operate from battery power for at least 30 minutes and shall support stable UWB communication at the configured 20 Hz update rate.
2. The four-anchor positioning system shall reconstruct ball position in the 3.75 m by 3.9 m test field with an average horizontal error below 10 cm, while maintaining stable communication during repeated test runs.
3. The software shall import or acquire raw RANGE4D ranging data, apply distance and coordinate filtering, reconstruct a continuous trajectory, and display the trajectory with point-level details in a Python GUI.

1.4 Solution Overview

The final solution uses five C2764088 development boards with DW1000/BU01 UWB modules [2], [3]. One board is configured as the tag and is installed in the ball. Four boards are placed as anchors at known coordinates around the test field: A0 at (1, 0, 0.75), A1 at (3.75, 0, 0), A2 at (3.75, 3.9, 0), and A3 at (0, 3.9, 0), all in meters. The tag performs TWR sequentially with A0, A1, A2, and A3. After collecting the four distances, the tag sends the range vector to A0, and A0 forwards a RANGE4D record to the PC over a serial link.

On the PC, Python software applies a 20 cm range offset correction, filters abnormal distance values, solves the 3D position from the four ranges, and applies coordinate level filtering using RMS residual, local-neighborhood consistency, velocity, and acceleration constraints. The GUI then displays anchors, reconstructed points, velocity direction, point metadata, and trajectory playback. For ground-rolling tests, additional modes constrain the Z coordinate so that noisy height estimates do not dominate the visualized path.

The physical test environment and anchor placement are shown in Figure 1.1.

Figure 1.1: Indoor test field and UWB anchor layout used for prototype validation.

1.5 System Decomposition and Subsystem Roles

Table 1.1 summarizes the final subsystem decomposition and the role of each block.

Table 1.1: System Decomposition and Final Subsystem Roles.

Subsystem	Main Role	Final Implementation
Embedded tag	Generate UWB ranging exchanges while moving with the ball.	C2764088 development board with DW1000/BU01 module, battery, and foam-mounted installation inside the test ball.
Anchor network	Provide fixed reference points for range measurement and coordinate reconstruction.	Four anchors placed at measured coordinates around a 3.75 m by 3.9 m indoor field.
Communication	Move range data from the tag to the processing computer.	Sequential TWR with A0-A3, followed by tag-to-A0 range-vector transfer and A0-to-PC serial output.
Data processing	Convert four UWB ranges into filtered 3D coordinates.	Python pipeline for offset correction, abnormal range filtering, least-squares reconstruction, RMS checks, and motion constraints.
GUI and visualization	Provide live acquisition, replay, and inspection tools.	Python GUI with 3D trajectory display, anchor labels, selectable points, velocity arrows, field boundary projection, and offline/live filter modes.
Power management	Provide compact battery-powered operation.	1000 mAh cell per development board and a custom TP4056-based power management module with Type-C charging.

1.6 Design Evolution and Modifications

The original design document proposed a TDoA-based system. During implementation, the design was changed to TWR because the selected BU01/DW1000 hardware could be made more reliable without tight inter-anchor clock synchronization. This change simplified the embedded communication flow and made it practical to operate a portable four-anchor prototype in an indoor test space.

The prototype also moved from an idealized softball implementation to a practical test ball package. For easier assembly, repeated tests, and protection of the development board, the tag was installed inside a ball using a foam insert with a central recess. The software scope was expanded with a ground mode because indoor rolling tests showed that limited anchor height separation made the Z estimate unstable. The added ground_z_cap and ground_fixed_z modes keep ground trajectories physically plausible and make the displayed tracking result easier to interpret.

1.7 Key Performance Factors

- Ranging bias: BU01 distance measurements contained a systematic positive offset, corrected by subtracting approximately 20 cm before reconstruction.
- Anchor geometry: the test field had limited vertical separation between anchors, which made X/Y positioning stronger than Z estimation.

- Update rate: although firmware could be configured for a nominal 40 Hz output, full four-anchor TWR and serial diagnostics limited stable throughput to approximately 25.55 Hz; the final system used 20 Hz.
- Filtering latency: the live GUI uses an approximately 1 s delay buffer so that local-neighborhood checks can identify abnormal points using both earlier and later samples.
- Replay capability: raw CSV storage allows the same data to be reprocessed with different filter modes after collection.

2 Design

2.1 Design Procedure and Choices

2.1.1 Embedded Emitter System

The embedded emitter system uses the C2764088 development board and the BU01 module based on DW1000 UWB technology [2], [3]. The tag firmware is written in C. The board is mounted inside the ball using a foam support structure that centers and protects the electronics while leaving space for the battery and wiring. This packaging was selected because it is easy to assemble, inspect, and modify during repeated tests.

The development board used for both the mobile tag and anchor nodes is shown in Figure 2.1.

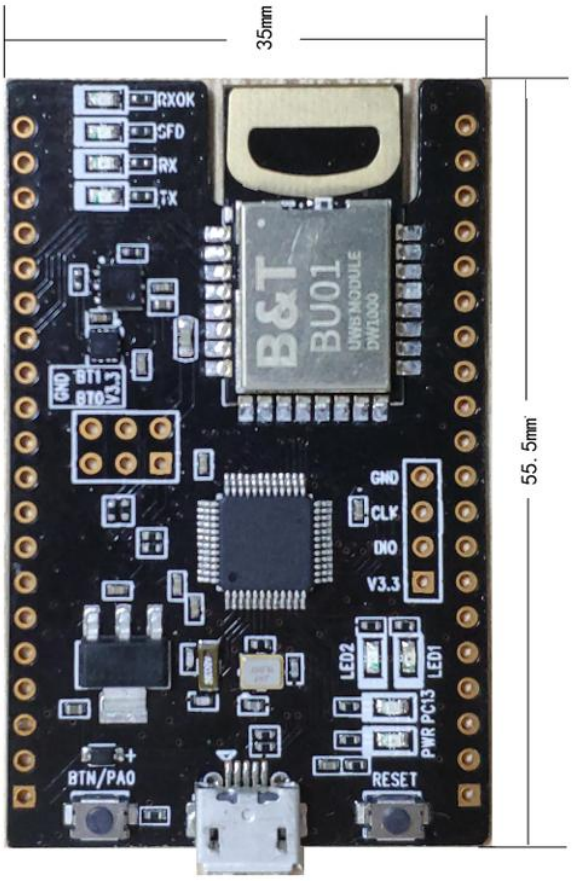

Figure 2.1: C2764088 development board used for the tag and anchors.

2.1.2 UWB Positioning System

The positioning subsystem uses four anchors at known coordinates. Instead of requiring synchronized anchors for TDoA, the final system uses TWR. For each frame, the tag ranges with A0, A1, A2, and A3 in sequence. The resulting distance vector is used to solve the nonlinear least-squares problem:

minimize over x,y,z: sum_i (sqrt((x - xi)^2 + (y - yi)^2 + (z - zi)^2) - di)^2

where (xi, yi, zi) is the known position of anchor i and di is the corrected UWB range from the tag to that anchor. The RMS residual of this fit is retained as a quality indicator for filtering and GUI inspection.

Figure 2.2 shows the BU01/DW1000 UWB module used as the ranging front end.

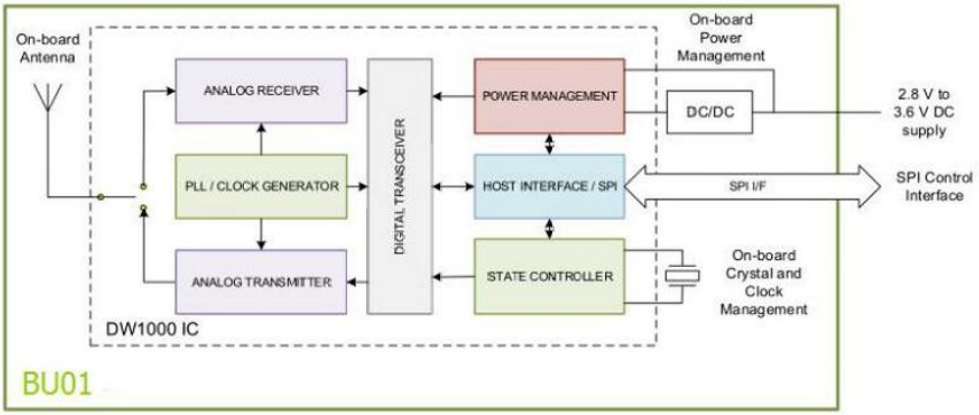

Figure 2.2: BU01/DW1000 UWB module and internal functional diagram.

2.1.3 Data Processing and Filtering

The filtering pipeline is layered deliberately. Raw distances first pass through a range layer that subtracts the measured 20 cm systematic offset and rejects obvious distance jumps. After the 3D coordinate is reconstructed, a coordinate layer evaluates RMS residual, local RMS consistency, velocity, and acceleration. Points that violate the motion model are not simply deleted; short abnormal segments are repaired by interpolation so that the time axis remains continuous for playback and velocity calculation.

The real-time display layer introduces an approximately 1000 ms buffer. This small delay allows the software to compare a point against both previous and following neighbors. The GUI also provides ground constraint modes. In free3d mode, the least-squares solution is shown directly. In ground_z_cap mode, Z is limited to a configurable maximum such as 0.20 to 0.30 m. In ground_fixed_z mode, Z is fixed near the ball radius or tag height, for example 0.08 to 0.10 m.

2.1.4 GUI and Visualization Subsystem

The GUI is implemented in Python [4]. It can open processed 3D coordinate CSV files, import raw RANGE4D ranging CSV files, or acquire a new trajectory from the A0 serial port. It then performs offset correction, filtering, reconstruction, and visualization. The GUI supports sequence/index playback, one-click full trajectory display, selectable points, anchor labels, velocity arrows, field boundary projection, and different motion filter modes such as slow walk, running, fast running, and ball motion.

The implemented GUI layout is shown in Figure 2.3.

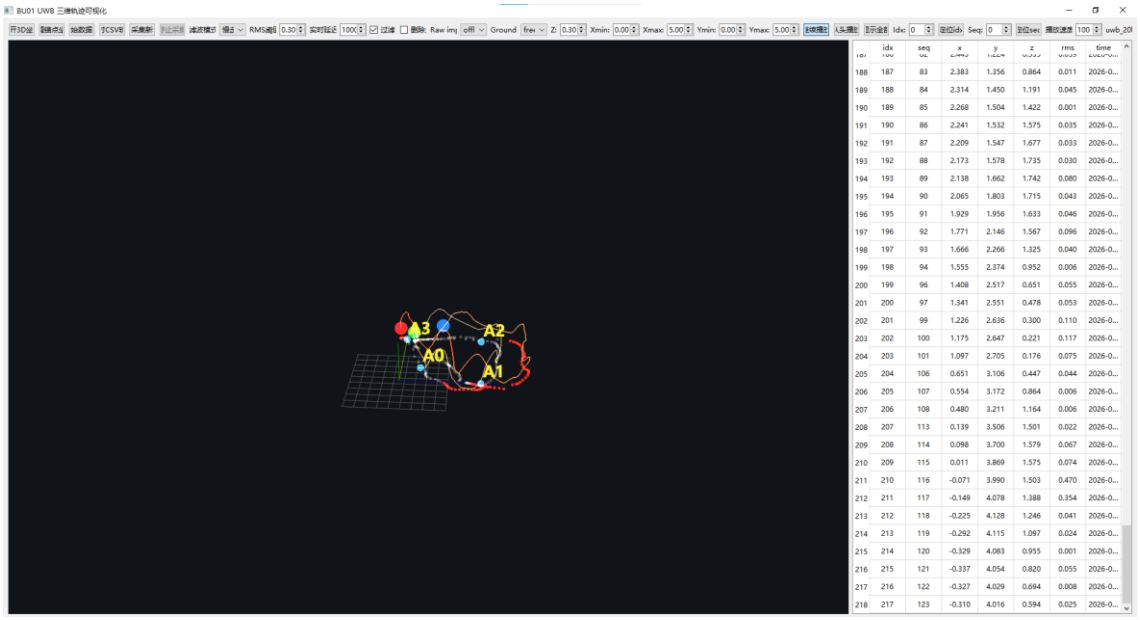

Figure 2.3: GUI for 3D trajectory display, point inspection, and replay.

2.2 Design Details

2.2.1 Embedded Tag Hardware and Mechanical Installation

Each development board is paired with a 1000 mAh battery. For the tag, the electronics are embedded in the test ball using a foam insert with a central groove. This method keeps the board from moving freely inside the ball, protects it from direct impact, and keeps the prototype easy to open for charging or firmware updates. The same development board hardware is used for the four anchors so that firmware and communication behavior remain consistent across nodes.

2.2.2 TWR Communication Flow

One complete frame contains several sequential actions: TAG to A0 ranging, TAG to A1 ranging, TAG to A2 ranging, TAG to A3 ranging, TAG to A0 range-vector transfer, and A0 to PC serial output. Each TWR exchange includes a poll packet, anchor reception, delayed response, tag reception, and timestamp calculation. This serialized four-anchor flow explains why practical throughput is lower than the nominal single-link firmware setting.

The system also outputs diagnostic fields such as receive power, first-path power, and power gap through a 115200 baud text CSV stream. These diagnostics were useful for development and debugging, but they increase serial bandwidth. At a nominal 40 Hz firmware setting, the measured average frame interval was about 39.14 ms, the median interval was about 39 ms, and the equivalent stable frequency was about 25.55 Hz. The final prototype therefore used 20 Hz for stable tracking.

2.2.3 Power Management System

The custom power management module is built around a USB Type-C input, a TP4056 lithium battery charging circuit [5], and a regulated supply path for the UWB board. The design allows each node to be charged conveniently and operated from a compact single-cell battery. Based on the measured current ranges, ordinary anchors draw approximately 120 to 180 mA, the gateway A0 draws approximately 150

to 220 mA, and the tag can draw higher current during active ranging. With a 1000 mAh battery, the worst-case operating time remains well above the 30 minute requirement.

The power-management schematic and PCB layout are shown in Figures 2.4 and 2.5.

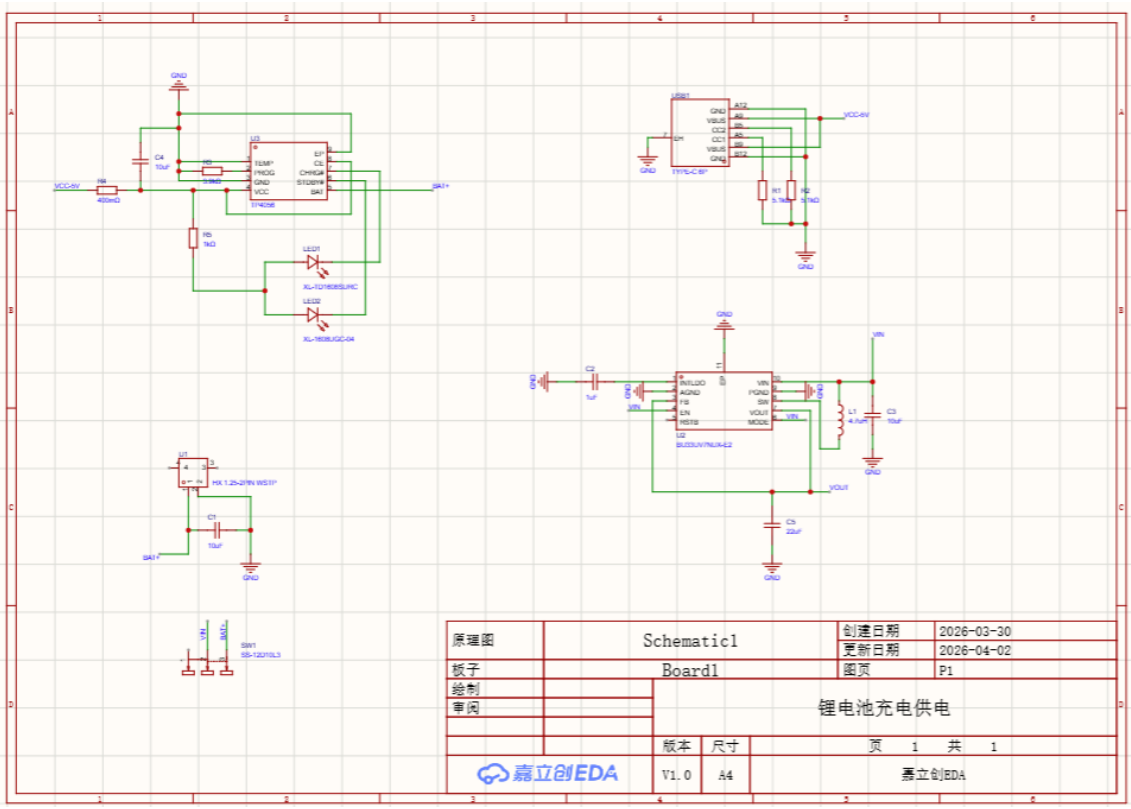

Figure 2.4: Power management module schematic.

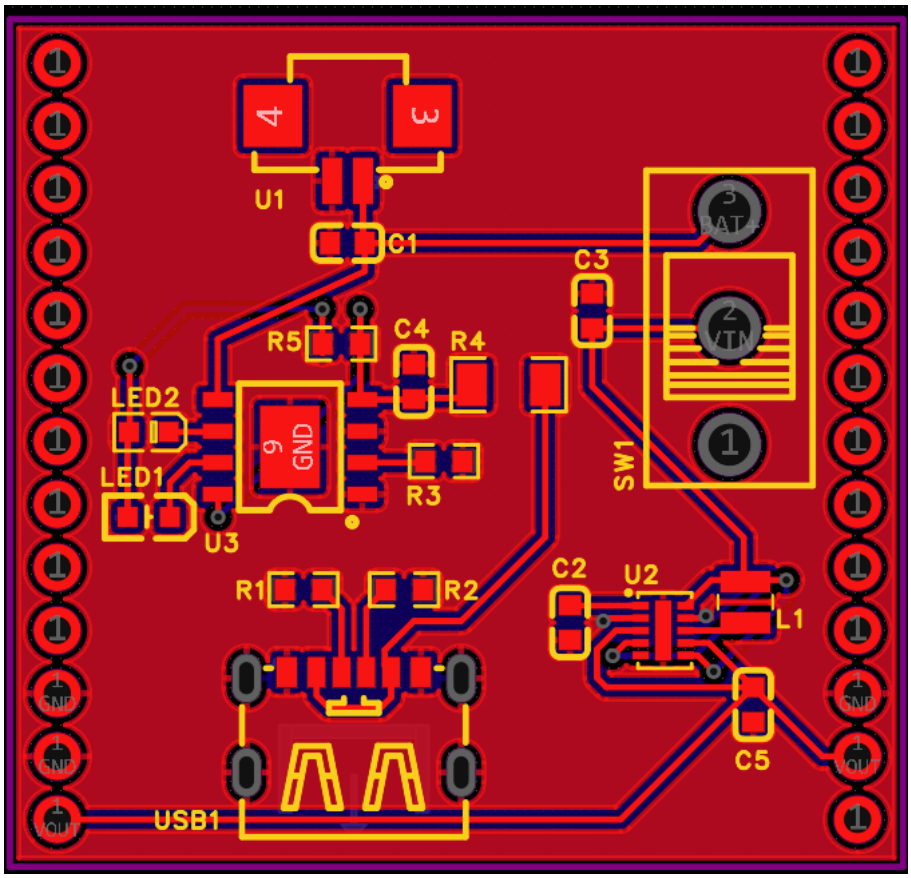

Figure 2.5: Power management module PCB layout.

2.2.4 Visualization Modes

The GUI includes both ordinary 3D trajectory visualization and a ground-oriented mode. Ordinary mode shows the raw three-dimensional reconstruction after filtering, which is useful for general motion and flight-like segments. Ground mode addresses the fact that the ball often rolls close to the floor while UWB geometry weakly constrains height. For these tests, constraining Z improves the practical readability of the trajectory without hiding the measured X/Y behavior.

Figures 2.6 and 2.7 compare the ordinary 3D display and the ground-constrained display.

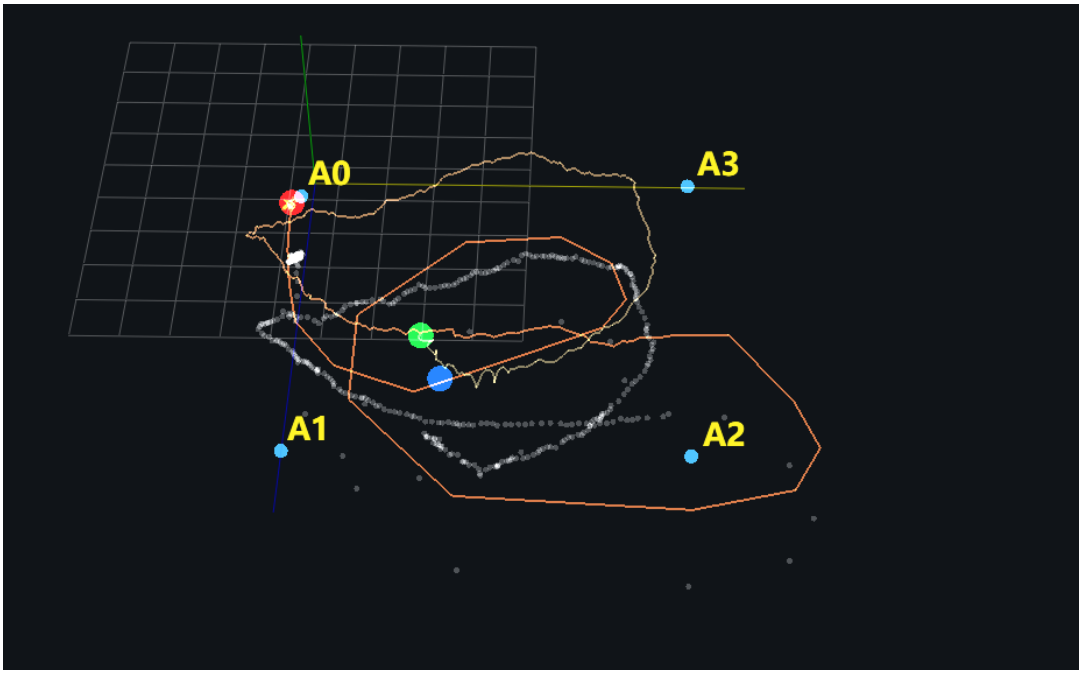

Figure 2.6: Reconstructed trajectory in ordinary 3D mode.

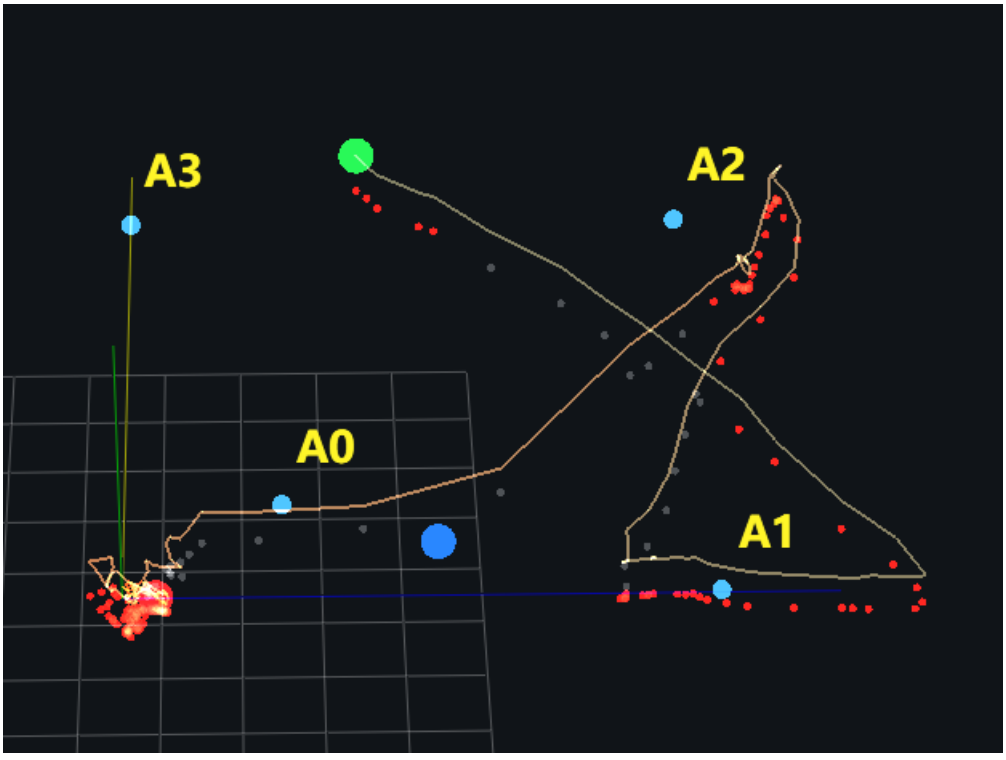

Figure 2.7: Ground-mode trajectory display with constrained height behavior.

3 Verification

3.1 Test Procedures and Environment

Verification was performed in an indoor field measuring approximately 3.75 m by 3.9 m. The four anchors were placed at the measured coordinates stated in Section 1.4. The tag was installed inside the test ball and configured to output at 20 Hz. For positioning accuracy tests, five reference coordinates were measured. Each reference point was sampled 15 times, producing 75 total position measurements. The raw Excel data were processed to compute axis-wise error, mean absolute error, RMSE, and 3D Euclidean error.

Software verification used both live acquisition and offline replay. Live display used the 1 s delay buffer, while offline replay reprocessed saved raw data with the selected filtering mode. Communication was observed during the test runs for packet loss and disconnects.

3.2 Subsystem Verification

3.2.1 Embedded Tag and Power Performance

The tag and anchors operated from battery power throughout testing. With a 1000 mAh cell, the measured current ranges imply several hours of operating time even under the higher-current node roles. This comfortably exceeds the 30 minute operation requirement. The tag could be installed inside the test ball and used for throw or rolling tests while the GUI continued to display the reconstructed trajectory.

3.2.2 UWB Positioning Accuracy

The positioning accuracy analysis was based on five known reference points and 15 measurements at each point. Overall, the system achieved an X-axis RMSE of 7.73 cm and a Y-axis RMSE of 6.49 cm. The average horizontal error was 8.44 cm, and the horizontal RMSE was 10.10 cm. The Z-axis RMSE was 27.33 cm, which is much larger than X/Y because the anchor layout provides weak geometric leverage in the vertical direction.

Table 3.1 summarizes the measured positioning errors at each reference point.

Table 3.1: Positioning Error Summary from Experimental Data.

Point	True coordinate (m)	N	X RMSE (cm)	Y RMSE (cm)	Z RMSE (cm)	Mean 3D error (cm)	Max 3D error (cm)
P1	(3, 3, 0)	15	13.68	11.31	39.85	43.60	46.14
P2	(3, 3, 1)	15	8.67	6.11	6.00	12.08	14.43
P3	(3.9, 0, 1)	15	1.78	5.65	10.71	11.27	26.72
P4	(3.9, 0, 0.55)	15	3.58	2.40	13.20	11.42	31.60
P5	(3.9, 0, 0)	15	4.56	2.84	42.67	42.93	48.33

Figures 3.1 and 3.2 visualize the signed axis errors and per-point RMSE results.

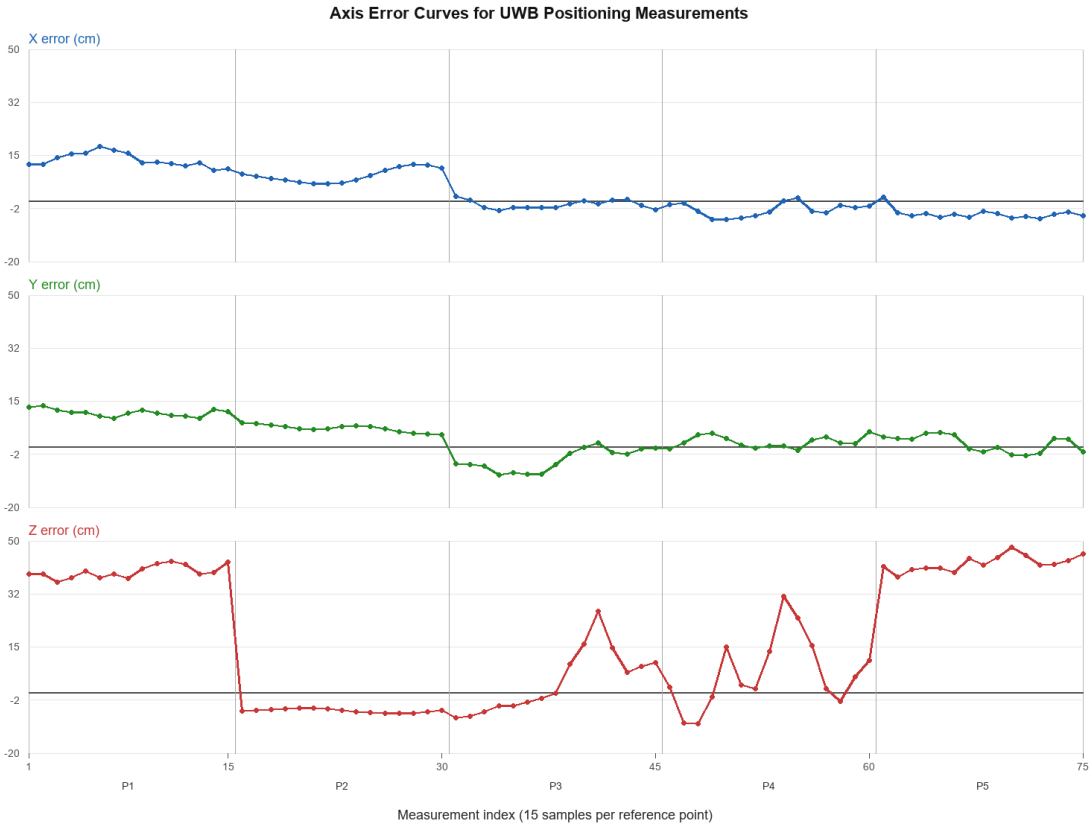

Figure 3.1: Axis-wise signed error curves across all 75 measurements.

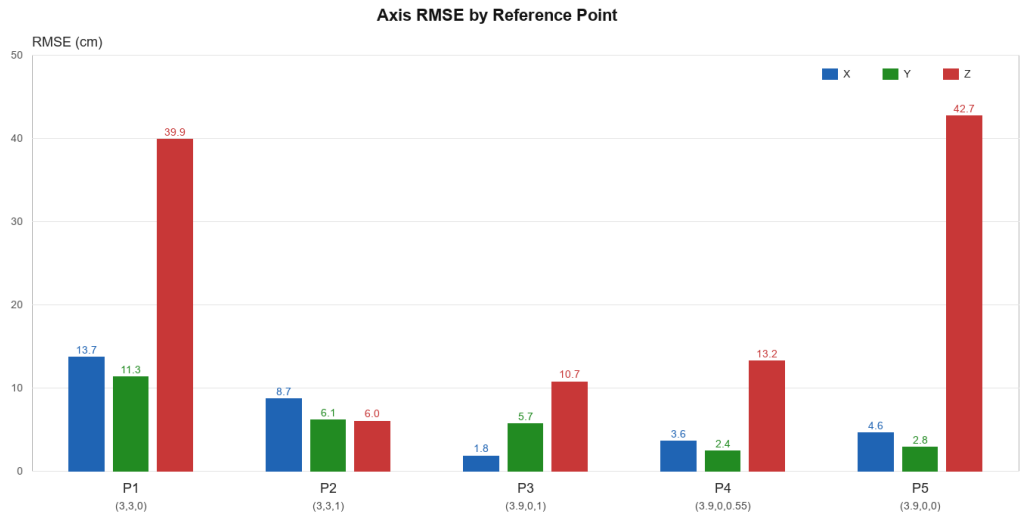

Figure 3.2: Axis RMSE by reference point.

3.2.3 Trajectory Reconstruction and GUI Functionality

The GUI successfully loaded processed coordinate CSV files, imported raw RANGE4D CSV files, acquired live serial data, and replayed trajectories by sequence or global index. Users could select points in the table or plot, inspect sequence, index, 3D coordinate, RMS residual, and velocity, and visualize velocity direction with arrows. The sequence wraparound problem from the 0 to 255 sequence range was handled by the global index.

3.2.4 Communication Stability and Update Rate

The final configured update rate was 20 Hz. Communication was stable during the reported tests, with no observed packet loss or disconnect events. A separate frequency stress test using a nominal 40 Hz firmware setting showed an average frame interval of approximately 39.14 ms, a median interval of approximately 39 ms, and an equivalent frequency of approximately 25.55 Hz. This result confirmed that the throughput bottleneck comes from serialized TWR timing, UWB waiting periods, and text serial output rather than from the GUI.

3.3 System-Level Verification (Integration Testing)

End-to-end tests exercised the full path from the embedded tag to the anchors, from A0 to the PC, through the Python reconstruction/filtering pipeline, and finally to the GUI. The system displayed trajectories in real time with the configured 1 s buffer and also allowed raw data to be saved and replayed offline. Ordinary 3D mode and ground-constrained mode both worked as intended, with ground mode improving visual stability for rolling segments where Z estimates were weak.

3.4 Requirement Verification Table

All final high-level requirements were met. The detailed requirement verification table is provided in Appendix A.

3.5 Tolerance Analysis Summary

The most important tolerance issue is UWB positioning error. The first contributor is a fixed ranging bias in the BU01 modules. Experimental calibration showed that subtracting approximately 20 cm from the raw distances improved consistency before the four-distance least-squares reconstruction. The second contributor is sporadic distance jumps caused by multipath, temporary obstruction, antenna attitude changes, or serial irregularities. The filtering pipeline addresses these jumps before and after coordinate reconstruction.

The third and most visible limitation is Z-axis sensitivity. UWB directly measures distance, not height. Because most anchors were near the same height and the horizontal distances were several meters, a large change in reconstructed Z could correspond to only a small change in range. For example, at a horizontal distance of about 3 m, changing Z from 0.10 m to 0.50 m can change the range by only several centimeters. This is on the same scale as practical UWB ranging noise during rolling tests, so the least-squares solution may lift the point vertically to satisfy inconsistent distances. This explains why the measured Z-axis RMSE (27.33 cm) is much larger than the X/Y RMSE values (7.73 cm and 6.49 cm).

The design mitigates this limitation through ground_z_cap and ground_fixed_z modes for ground rolling data. Long-term improvements include increasing the height separation of anchors, placing at least two anchors at elevated and spatially separated positions, separating flight and rolling segments in software, and using higher-bandwidth hardware or serial protocols to increase update rate.

4 Costs

4.1 Parts and Materials

The total parts and materials cost was 1204.28 RMB, approximately $167.26 USD using an exchange rate of 7.2 RMB/USD. This includes five C2764088 development boards for one tag and four anchors, PCB fabrication and SMT assembly for five power-management modules, and the component BOM for those modules.

Table 4.1 lists the parts-cost summary used for the total project cost calculation.

Table 4.1: Summary Bill of Materials and Costs.

Item	Qty	Unit (RMB)	Total (RMB)	Unit (USD)	Total (USD)
C2764088 development board	5	174.00	870.00	24.17	120.83
Power-management PCB fabrication	5 modules	4.00	20.00	0.56	2.78
Power-management SMT assembly	5 modules	55.60	278.00	7.72	38.61
Power-management electronic BOM	5 modules	7.26	36.28	1.01	5.04
Total			1204.28		167.26

4.2 Labor

The project involved three team members over a 14-week design period. Following the same labor-cost convention used in the template, labor is estimated using an ideal graduate engineering salary of $30 per hour and a 2.5 multiplier. Assuming 6 hours per person per week, the total labor time is 252 hours.

Labor Cost = $30/hour * 252 hours * 2.5 = $18,900.00

4.3 Total Cost

Total Project Cost = Parts Cost + Labor Cost = $167.26 + $18,900.00 = $19067.26. The monetary parts cost is modest because the system uses compact UWB development boards and a small custom power board, while the labor estimate reflects the time required for firmware, filtering, GUI design, integration, and experimental verification.

5 Conclusion

5.1 Project Summary and Accomplishments

This project produced a working UWB-based Smart SoftBall tracking prototype. The final system embedded a UWB tag inside a ball, deployed four anchors around an indoor field, implemented sequential TWR ranging, reconstructed 3D coordinates in Python, filtered abnormal points, and displayed live and offline trajectories in a GUI. The most important accomplishments were the successful TWR ranging and localization pipeline, the complete GUI workflow, and the layered filtering method that made noisy UWB trajectories usable for visualization.

Experimental data showed that the system achieved average horizontal error of 8.44 cm over 75 measurements. The GUI could replay trajectories, inspect individual points, display anchors and velocity direction, and support a ground mode for rolling tests. Together, these functions demonstrate the feasibility of a portable non-camera tracking approach for small-scale training and experimental use.

5.2 Challenges and Future Work

The first major challenge was the fixed ranging bias of the BU01 modules. The team had to identify and compensate an approximately 20 cm systematic error before the position solver produced stable results. The second challenge was update rate. Although the firmware could be configured to 40 Hz, the complete four-anchor TWR sequence, diagnostic output, and 115200 baud serial link limited stable throughput to about 25.55 Hz, so 20 Hz was selected for the final system. The third challenge was Z-axis accuracy, caused by limited height separation between anchors and weak vertical geometry.

Future work should use hardware with higher ranging rate and communication bandwidth, reduce diagnostic text output or switch to a binary serial protocol, increase serial baud rate to 460800 or 921600, and optimize DW1000 timeout and polling intervals. If the venue allows, at least two anchors should be raised to higher and more diverse positions to improve Z-axis observability. The software should also separate flight and rolling segments so that free 3D reconstruction can be used during flight while ground constraints are applied only during rolling segments.

5.3 Ethical Considerations

The system does not use cameras and does not collect images, faces, or personal identity data. It records only UWB ranging and reconstructed ball-motion data for training and technical analysis. This reduces privacy risk compared with camera-based alternatives. The GUI and report should present accuracy limitations transparently so that users do not interpret noisy UWB results, especially Z-axis estimates, as exact ground truth.

Electrical and physical safety were also considered. Batteries are enclosed with the electronics, the board is protected by foam inside the test ball, and the system uses low-power UWB modules. The final product should still be treated as a prototype: it is appropriate for controlled testing, but additional ruggedization, battery protection, and mechanical balancing would be needed before competitive sports use. These choices follow the broader engineering responsibility to communicate limitations and reduce foreseeable user risk [6].

5.4 Broader Impacts

A portable UWB tracking system can make motion analysis more accessible outside of professional venues. Economically, it offers a lower-cost alternative to camera arrays for small training spaces. Educationally, it combines embedded systems, RF ranging, signal filtering, GUI design, and experimental data analysis in one integrated project. Socially, the system can support recreational and amateur training without requiring video recording of participants. Technically, the project highlights both the promise and limitations of UWB tracking, especially the importance of anchor geometry and calibration.

References

[1] Hawk-Eye Innovations, "Hawk-Eye Technology and Tracking Systems," Hawk-Eye Innovations Ltd. [Online]. Available: https://www.hawkeyeinnovations.com/. Accessed: May 16, 2026.

[2] Decawave Ltd., "DW1000 User Manual," Decawave Ltd., 2020.

[3] Ai-Thinker Technology Co., Ltd., "BU01 UWB Module Product Specification," Ai-Thinker Technology Co., Ltd.

[4] Python Software Foundation, "Python 3 Documentation." [Online]. Available: https://docs.python.org/3/. Accessed: May 16, 2026.

[5] UMW, "TP4056 1 A Linear Li-Ion Battery Charger Datasheet," UMW.

[6] IEEE Standards Association, "Ethically Aligned Design," IEEE Standards Association, 2021.

Appendix A Requirement Verification Table

Table A.1 provides the detailed requirement verification matrix.

Table A.1: Requirement Verification Table.

ID	Description	Verification Method	Result	Met ?
HLR-1	Battery-powered tag and anchors support at least 30 minutes of operation and stable 20 Hz UWB communication.	Battery/current estimate and live communication test.	Worst-case estimated endurance exceeds 4 hours; final configured update rate is 20 Hz.	Yes
HLR-2	Four-anchor positioning reconstructs trajectory in the 3.75 m by 3.9 m field with average horizontal error below 10 cm.	Five reference positions, 15 measurements per position, error analysis from Excel data.	Average horizontal error 8.44 cm; X/Y RMSE 7.73/6.49 cm.	Yes
HLR-3	Software imports or acquires RANGE4D data, filters it, reconstructs trajectory, and displays point-level details.	GUI functional test using live serial input and offline CSV replay.	Live acquisition, raw CSV import, replay, point selection, RMS display, velocity arrows, and ground modes verified.	Yes
Sub-Req 1	Communication remains stable during tests.	Observe packet loss and disconnect behavior during test runs.	No packet loss or disconnect events observed during reported tests.	Yes
Sub-Req 2	Ground-mode visualization reduces unstable Z behavior during rolling tests.	Compare ordinary and ground-mode trajectory displays.	Ground_z_cap and ground_fixed_z modes produced more readable rolling trajectories.	Yes

Appendix B Bill of Materials

The following table summarizes the power-management component BOM from Power_management_BOM.xlsx. Prices are JLCPCB/LCSC component prices in RMB; Table B.1 lists the line items.

Table B.1: Power-Management Module Electronic BOM.

No.	Qty	Comment	Designator	Manufacturer Part	Unit RMB	Line RMB
1	3	10uF	C1,C3,C4	CL10A106MA8NRNC	0.1370	0.4110
2	1	1uF	C2	CL10A105KB8NNNC	0.0471	0.0471
3	1	22uF	C5	CL10A226MQ8NRNC	0.0620	0.0620
4	1	4.7uH	L1	FTC252010S4R7MBCA	0.2280	0.2280
5	1	XL-TD1608SURC	LED1	XL-TD1608SURC	0.2280	0.2280
6	1	XL-1608UGC-04	LED2	XL-1608UGC-04	0.0315	0.0315
7	2	5.1k ohm	R1,R2	0603WAF5101T5E	0.0096	0.0192
8	1	3.9k ohm	R3	0603WAF3901T5E	0.0098	0.0098
9	1	400m ohm	R4	1206W4F400LT5E	0.0320	0.0320
10	1	1k ohm	R5	0603WAF1001T5E	0.0096	0.0096
11	1	SS-12D10L3	SW1	SS-12D10L3	1.8250	1.8250
12	1	HX 1.25-2PIN WSTP	U1	HX 1.25-2PIN WSTP	0.2200	0.2200
13	1	BU33UV7NUX-E2	U2	BU33UV7NUX-E2	3.4400	3.4400
14	1	TP4056	U3	TP4056	0.4570	0.4570
15	1	TYPE-C 6P	USB1	TYPE-C 6P	0.2350	0.2350

The electronic BOM total is 7.2552 RMB per board, or 36.28 RMB for five boards before PCB fabrication and SMT assembly fees.

Appendix C Physical Prototype Photos

This appendix presents photographs and screenshots of the completed prototype, including the assembled UWB nodes, test field, GUI, and power-management board. Figures C.1-C.4 show these supporting artifacts.

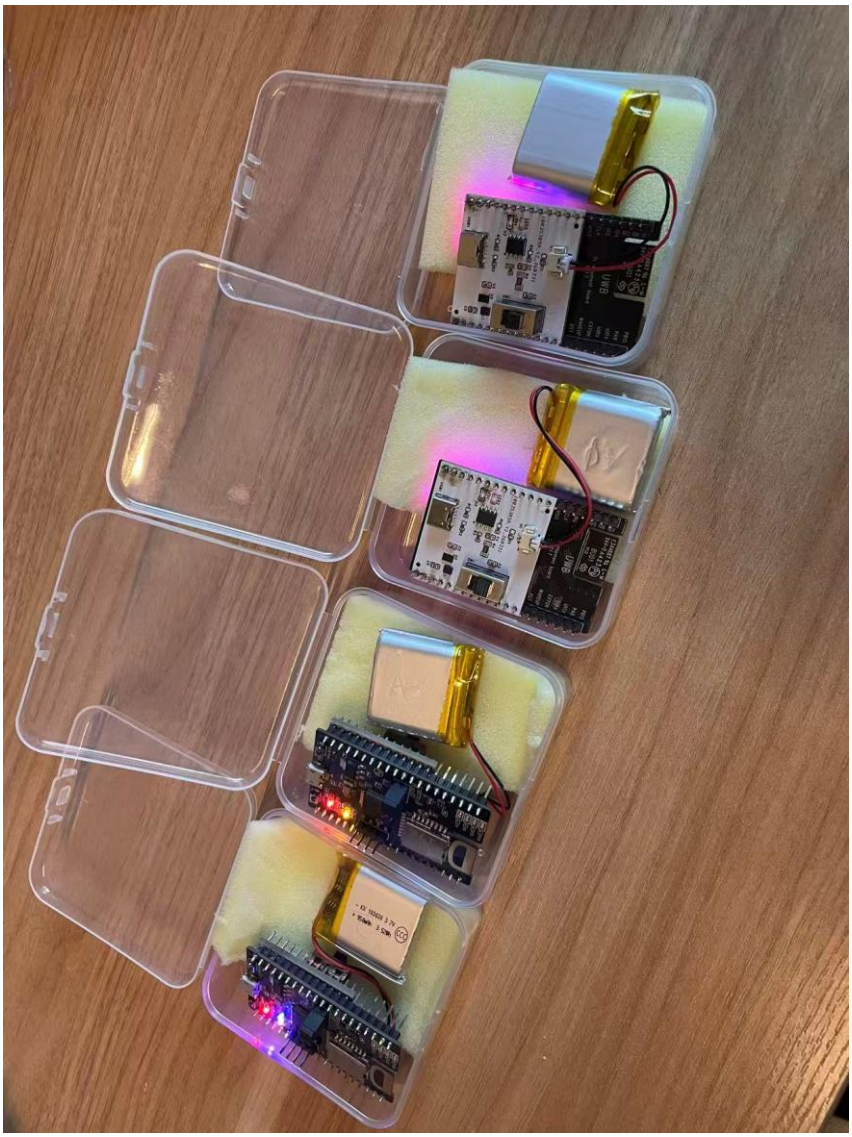

Figure C.1: Assembled UWB nodes and battery-powered prototype hardware.

Figure C.2: Test field used for trajectory collection.

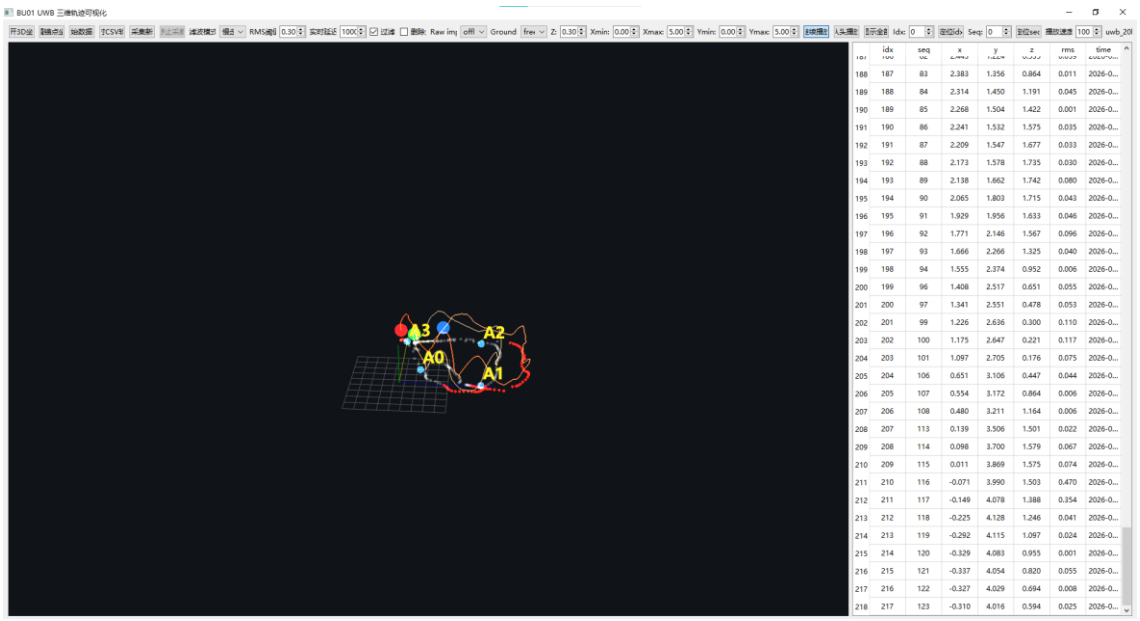

Figure C.3: GUI screenshot showing trajectory display and point data table.

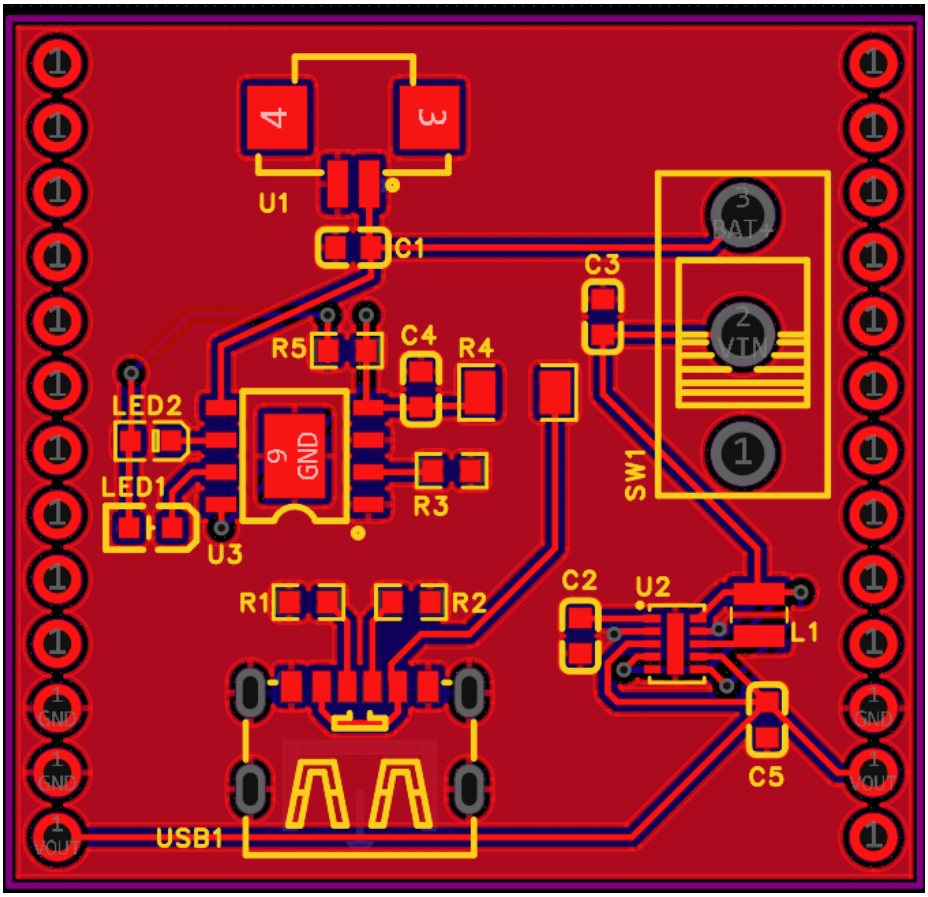

Figure C.4: Power-management PCB layout used in the prototype.