LoRa-Based Wireless Localization System

Group G24

November 26, 2025

Team Members

Name	Roll Number	Role
N V Poorna Vadhan	2024MCB1306	Hardware Assembly & Documentation
Mekala Sathwik Varma	2024MCB1305	Arduino Firmware Logic
Pola Sai Sujith	2024MCB1309	Data Collection & Calibration
Bukke Balaji Naik	2024MCB1290	Web Dashboard & Frontend
Jatin	2024MCB1296	Math Modelling & Backend

Project Links:

Live Dashboard (Local) | Detailed Write-Up PDF

Abstract

This project presents the design, implementation, and evaluation of a low-power, long-range wireless localization system capable of tracking assets in GPS-denied environments. The system estimates the 2D position of a mobile node in real-time by utilizing the **Received Signal Strength Indicator (RSSI)** from three fixed anchor nodes. Built upon the **LoRa SX1278** platform and **Arduino UNO**, the system overcomes the inherent instability of RF signals—caused by multipath fading and environmental noise—through a robust software pipeline. This pipeline integrates a **Log-Distance Path Loss model** for distance estimation, a **1D Kalman Filter** for noise reduction, and **2D Trilateration** for geometric coordinate determination. The final output is visualized on a modern web dashboard powered by Python and HTML5 Canvas.

Contents

1 Introduction

Agriculture, mining, and logistics are the backbone of many economies, yet these industries often struggle with asset tracking in vast or complex environments. Our project, the **LoRa-Based Wireless Localization System**, aims to solve this by integrating long-range radio modules with advanced signal processing algorithms.

1.1 Problem Statement

Tracking assets, livestock, or personnel in outdoor and complex environments is a critical challenge.

- **GPS Limitations:** GPS fails in indoor environments, underground tunnels, and dense forests due to lack of satellite line-of-sight.
- **Short-Range Limitations:** Wi-Fi and Bluetooth have poor range and high signal attenuation.
- **Cost:** Industrial UWB or RFID solutions are prohibitively expensive.

1.2 Proposed Solution

We propose a system operating at **433 MHz** using LoRa technology. By utilizing simple RSSI triangulation, we eliminate the need for expensive time-of-flight hardware. The system monitors real-time RSSI values, processes them through a Kalman Filter to remove noise, and calculates the precise location of a mobile node relative to three fixed anchors.

2 Hardware Components

The core of our localization device is the Arduino UNO microcontroller paired with the SX1278 LoRa transceiver.

2.1 Arduino UNO

The Arduino UNO is the processing unit for each node. It handles the SPI communication with the LoRa module and serial communication with the backend server.

2.2 LoRa SX1278 Module (Ra-02)

The Ai-Thinker Ra-02 module uses the SX1278 chip. It was chosen for its spread spectrum technology, which allows it to communicate over long distances (kilometers) even with very low power.

[Insert Arduino UNO Image Here]
Ensure the Arduino UNO.jpg file is in the directory.

Figure 1: Arduino UNO Microcontroller

[Insert LoRa Module Image Here]
Ensure the LoRa Module SX1278.jpg file is in the directory.

Figure 2: Ai-Thinker Ra-02 LoRa Module

2.3 Antenna Implementation

To ensure reliable communication at 433 MHz, we fabricated custom quarter-wavelength monopole antennas.

$$L = \frac{c}{4f} = \frac{3 \times 10^8}{4 \times 433 \times 10^6} \approx 0.173 \text{ meters} \quad (1)$$

A 17.3 cm solid core wire was soldered to the ANT pin of each module.

3 Connections & Wiring

Correct wiring is critical as the SX1278 is a 3.3V device. While the Arduino UNO uses 5V logic, the SPI lines are generally tolerant, but the VCC must strictly be 3.3V.

[Insert Wiring Diagram Here]
Ensure the Module.jpg file is in the directory.

Figure 3: Wiring Diagram: Arduino UNO to SX1278

LoRa Pin	**Arduino Pin**	**Function**
VCC	3.3V	Power (Warning: Do not use 5V)
GND	GND	Ground
MISO	D12	SPI Master In Slave Out
MOSI	D11	SPI Master Out Slave In
SCK	D13	SPI Clock
NSS	D10	Chip Select
DIO0	D2	Interrupt (RxDone)
RST	D9	Reset

Table 1: Detailed Pin Connections

4 Firmware Implementation

The firmware is written in C++ using the Arduino framework. It manages the LoRa radio state machine.

4.1 Core Logic Description

1. **Initialization (setup):** Configures SPI pins and initializes the LoRa radio at 433 MHz with a TxPower of 20dBm.
2. **Mobile Node Loop:**
   - Listens for incoming packets using `LoRa.parsePacket()`.
   - Identifies the sender ID (e.g., “ANCHOR_1”).
   - Reads the RSSI (Signal Strength).
   - Aggregates data from all three anchors before sending a JSON-formatted string to the serial port.

3. **Anchor Node Loop:** Simply broadcasts its ID every 200ms to act as a beacon.

5 Methodology & Mathematical Modelling

The raw data from the sensors is noisy. We apply a three-stage mathematical pipeline to ensure accuracy. Detailed mathematical derivations can be found in the attached **G24-WriteUp.pdf**.

5.1 Stage 1: Path Loss Model

We convert RSSI to distance using the Log-Distance Path Loss model:

$$d = 10^{\frac{A-RSSI}{10n}} \tag{2}$$

Where A is the reference RSSI at 1 meter, and n is the environmental path loss exponent.

5.2 Stage 2: Kalman Filter

To smooth the jittery RSSI values, we implemented a 1D Kalman Filter. It recursively estimates the true signal strength by balancing the "Process Noise" (R) and "Measurement Noise" (Q).

```
// Conceptual Logic for Kalman Update
K = P / (P + Q);
Estimate = Estimate + K * (Measurement - Estimate);
P = (1 - K) * P;
```

5.3 Stage 3: 2D Trilateration

The position (x, y) is determined by finding the intersection of three circles centred at the anchors (x_i, y_i) with radii d_i.

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2 \quad \text{for } i = 1, 2, 3 \tag{3}$$

Our system numerically solves these equations to pinpoint the user on the canvas.

6 Web Interface & Analysis

The system includes a Python Flask backend and a responsive HTML5/JavaScript Frontend. The dashboard serves as the central control unit for visualization, calibration, and data analysis.

6.1 Live Dashboard Features

The dashboard is divided into three functional zones designed for ease of use:

1. **Real-Time Triangulation Map (Canvas):**
    - Displays a dynamic 2D grid representing the physical space.
    - Visualizes the fixed positions of the three Anchor Nodes (Alpha, Beta, Gamma).
    - Renders the estimated position of the Mobile Node as a moving blue indicator.
    - Draws "shadow circles" representing the calculated signal radius from each anchor, allowing visual debugging of the intersection logic.
2. **Signal Monitoring Panels:**
    - **Live RSSI Cards:** Shows both the Raw (noisy) and Filtered (Kalman) signal strength for all three anchors simultaneously.
    - **Distance Estimation:** Displays the computed distance in meters derived from the Path Loss model.
    - **Latency Tracker:** Monitors the time-of-flight and processing delay in milliseconds.
3. **Analytical Graphs (Chart.js):**
    - **Signal Analysis Chart:** A real-time line graph comparing the Raw RSSI vs. Filtered RSSI. This provides immediate visual feedback on the effectiveness of the Kalman Filter.
    - **Link Latency Chart:** Tracks communication stability over time.

6.2 System Configuration & Control

To ensure adaptability to different environments (e.g., open fields vs. indoor corridors), the system includes a comprehensive configuration suite accessible via the Settings Modal.

- **Path Loss Calibration:**
    - **Reference Power (A):** Adjustable constant defining RSSI at 1 meter.
    - **Environmental Factor (n):** Sliders to tune the signal decay rate based on the environment (Free Space ≈ 2.0, Indoor ≈ 3.0).
- **Kalman Tuning:**
    - **Process Noise (R):** Controls how fast the system expects the mobile node to move.
    - **Measurement Noise (Q):** Controls how much the system trusts the noisy sensor data. Increasing Q yields smoother motion but higher latency.
- **Anchor Management:** Users can dynamically update the (x, y) coordinates of the anchors without restarting the server, allowing for flexible deployment geometries.

6.3 Real-Life Analysis

During field testing, we observed that raw RSSI values fluctuated by ± 6 dBm. After enabling the Kalman Filter with $Q = 2.0$, fluctuations reduced to ± 1.5 dBm, resulting in a stable position estimate on the map. For more detailed observations, please refer to the "Implementation & Results" section in the project write-up.

7 Conclusion

Our "LoRa-Based Wireless Localization System" successfully demonstrates the power of IoT in tracking applications. By providing real-time data and intelligent error reduction, we can empower industries to track assets without expensive infrastructure. The system is scalable and can be upgraded with GPS fusion or mesh networking in the future for even higher precision.