

Hardware Architecture

Smart Waste Bin Monitoring System

Introduction

The hardware architecture of the Smart Waste Bin Monitoring System is designed to enable reliable sensing, local processing, and long-range wireless communication from distributed waste bins. Each dustbin functions as an intelligent edge node that periodically measures its fill level and transmits essential data to a central gateway using LoRa communication. The hardware design prioritizes low power consumption, scalability, and independence from internet connectivity at the bin level.

Hardware Objectives

The main objectives of the hardware architecture are:

- To accurately measure waste fill levels inside bins
- To perform basic processing at the edge node
- To transmit structured data frames over long distances
- To support multiple bins using unique node identifiers
- To operate reliably in areas with limited or no internet connectivity

Hardware Components

Edge Node (Smart Bin Unit):

Each waste bin is equipped with the following components:

ESP32 Microcontroller

- Acts as the central controller of the edge node
- Interfaces with sensors using GPIO pins

- Communicates with the LoRa module using UART
- Handles data framing and transmission logic
- Selected for its low power consumption and multiple serial interfaces

Ultrasonic Sensor

- Measures the distance between the sensor and the waste surface
- Used to calculate bin fill level percentage
- Provides non-contact and reliable measurement
- Suitable for dusty and humid environments

LoRa Module (UART-Based)

- Enables long-range wireless communication
- Interfaced with ESP32 using UART protocol
- Operates in sub-GHz frequency band
- Supports transmission over several kilometers in urban environments

Power Supply Unit (Solar Powered)

The Smart Waste Bin edge node is designed to operate using a **solar-powered energy system**, enabling autonomous and long-term operation without dependency on grid power.

Components:

- **Solar Panel**
 - Harvests solar energy during daylight hours
 - Mounted on or near the dustbin for maximum exposure
- **Rechargeable Battery**
 - Stores energy collected from the solar panel
 - Supplies power during nighttime or low sunlight conditions
- **Charge Controller**
 - Regulates charging and discharging of the battery

- Protects against overcharging and deep discharge
- Ensures stable voltage supply to ESP32 and peripherals

Edge Node Hardware Logic

The following steps describe the hardware logic executed at each smart bin:

1. ESP32 triggers the ultrasonic sensor
2. Distance to waste surface is measured
3. Fill level percentage is calculated
4. Node ID and fill data are prepared
5. A structured data frame is created
6. Data frame is transmitted via UART to the LoRa module
7. LoRa module sends the data wirelessly to the gateway

➤ Communication Interface Design:

UART Interface (ESP32 ↔ LoRa)

- UART is used for reliable point-to-point communication
- Configured baud rate ensures error-free data transfer
- Frame-based transmission is implemented for data integrity

Node Identification and Scalability

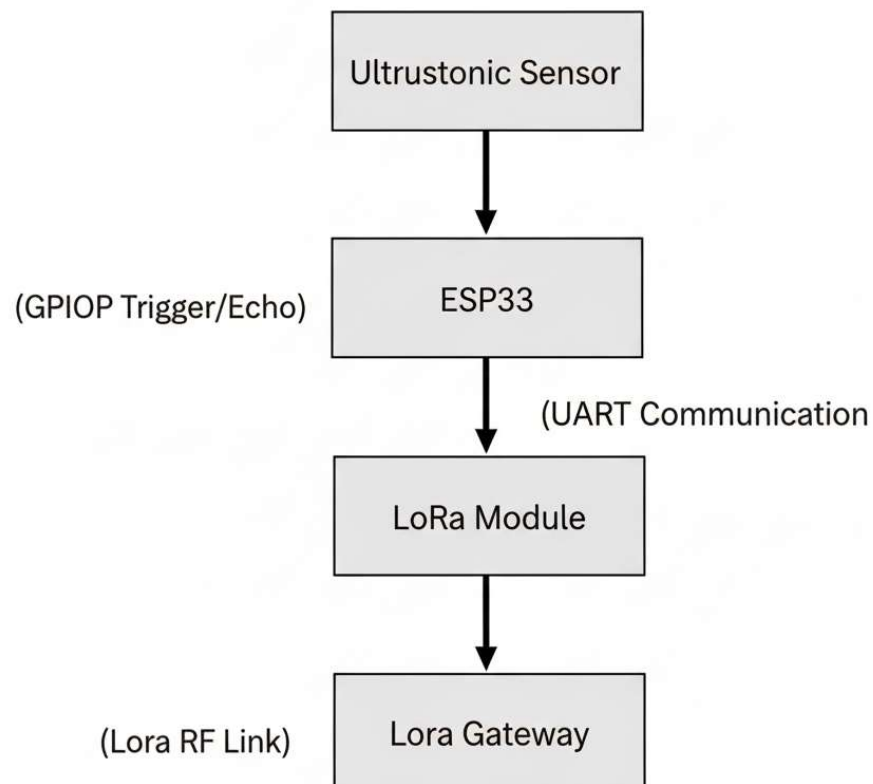
- Each bin is assigned a **unique Node ID**
- Node ID is embedded in every transmitted frame
- Allows the gateway and backend to differentiate between bins
- Supports large-scale deployment across multiple locations

Bin Location Handling

- Waste bins are **static assets**
- Geographic coordinates are pre-stored in the backend database
- Node ID is mapped to location information in software
- Eliminates need for GPS at the hardware level

Design Rationale

- LoRa eliminates dependency on internet connectivity
- ESP32 offers sufficient processing with low power usage
- Frame-based communication ensures data reliability
- Static location mapping reduces hardware complexity



The proposed hardware architecture enables efficient sensing, reliable long-range communication, and scalable deployment of smart waste bins. By keeping the edge node logic minimal and shifting intelligence to the backend, the system ensures robustness and ease of maintenance.

