EL-701

Module 7: Edge Computing

"Design and Simulation of Embedded Components"

Submitted by:

Naitik
202421023

Submitted to:

Dr. Tapas Maiti

Dhirubhai Ambani
University

DAU Gandhinagar, Gujarat

28 April 2025

Three-Layer Wireless Communication Using ESP32

Coursework Report – Edge Computing

1. Introduction

The rapid growth of the Internet of Things (IoT) and edge computing has led to the need for scalable, efficient, and robust wireless communication architectures. In this report, we present the design and implementation of a **three-layer wireless communication system** using the ESP32 microcontroller, a popular WiFi/Bluetooth-enabled chip. This system demonstrates how hierarchical networking, data aggregation, and edge processing can be achieved in a smart environment such as a building, a sensor network, or an industrial automation scenario.

2. System Architecture

2.1 Overview

The three-layer architecture improves scalability and reliability by dividing devices into three functional layers:

- **Layer 1 (Sensor/Client Layer):**
  Multiple ESP32-based nodes equipped with sensors (such as DHT11 for temperature/humidity or motion sensors) collect environmental data.

- **Layer 2 (Aggregator/Relay Layer):**
  Intermediate ESP32 devices act as aggregators, receiving sensor data from multiple clients, performing local processing or filtering, and forwarding summarized data to the top layer.

- **Layer 3 (Edge Server Layer):**
  A central node (ESP32 or a more powerful device like Arduino UNO R4 WiFi or Raspberry Pi) collects aggregated data, hosts a dashboard, and may interface with cloud services if required.

2.2 Layered Communication Flow

2.3 System Block Diagram

Block Diagram

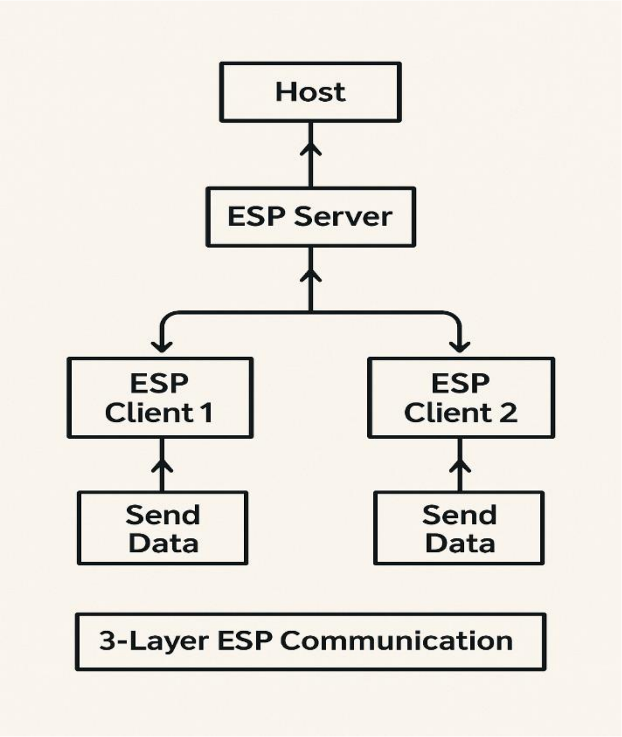


Figure: Three-layer communication topology with multiple ESP32 sensors, aggregator relays, and an edge server.

---

3. Implementation Details

3.1 Hardware Components

- **ESP32 Development Boards** (for all layers)
- **DHT11/DHT22 or other sensors** for environmental data
- **WiFi Router** or ESP32 SoftAP for Layer 1/2 connections
- **Optional**: Arduino UNO R4 WiFi, Raspberry Pi, or PC for Layer 3

3.2 Software Stack

- **ESP-IDF/Arduino Framework** for ESP32 programming
- **HTTP/REST or ESP-NOW** for wireless communication
- **JSON** for data serialization
- **HTML/CSS/JS** for dashboard (if using Layer 3 web server)

3.3 Layer Functions

Layer 1: Sensor/Client

- Periodically read sensor data.
- Transmit data via WiFi (HTTP POST) or ESP-NOW to aggregator.
- Example code snippet (Arduino-style):

```
// Pseudocode
loop() {
  readSensors();
  sendDataToAggregator();
  delay(5000);
}
```

Layer 2: Aggregator/Relay

- Receives data from several Layer 1 nodes.
- Optionally performs local computation (averaging, filtering).
- Forwards processed/aggregated data to Layer 3.
- Example:

```
// Pseudocode
onDataReceivedFromClient() {
  bufferData();
  if (timeToSend()) {
    aggregate();
    sendToEdgeServer();
  }
}
```

Layer 3: Edge Server

- Receives data from all aggregators.
- Stores, visualizes, and possibly analyzes the data.
- May host a web interface for real-time monitoring and configuration.
- Example:

```
// Pseudocode
onDataFromAggregator() {
  storeInDatabase();
  updateDashboard();
}
```

---

4. Communication Protocols

- **Layer 1 → Layer 2:**
  - Can use ESP-NOW (low-power, peer-to-peer) or WiFi TCP/UDP.
  - ESP-NOW is ideal for local, battery-powered sensors.
- **Layer 2 → Layer 3:**
  - Typically uses WiFi (HTTP/REST, MQTT, or raw TCP).
  - JSON is used for data interchange.

Sample JSON Payload:

```
{
  "node_id": "sensor_01",
  "temperature": 26.7,
  "humidity": 41.2,
  "timestamp": 1717564800
}
```

---

5. Edge Computing Advantages in Three-Layer Design

- **Scalability:**
  Aggregators reduce the load on the central server, allowing many more sensors to be deployed.

- **Reduced Latency:**
  Local aggregation and filtering mean less data is transmitted, and response times are faster.

- **Reliability:**
  If the central server is temporarily unreachable, aggregators can buffer data and forward it when possible.

- **Bandwidth Efficiency:**
  Only summarized data flows upward, conserving wireless bandwidth.

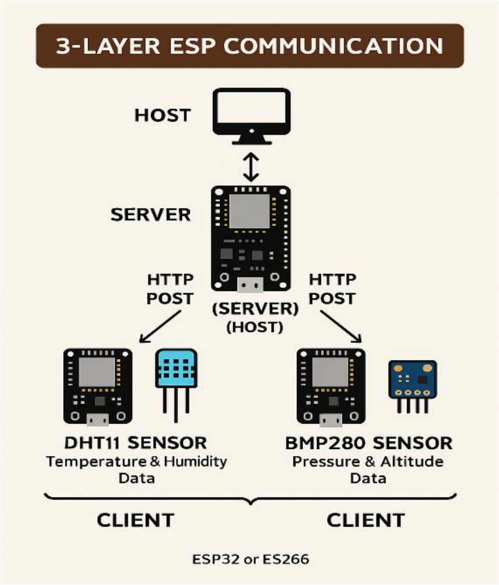
---

6. Results & Example Deployment

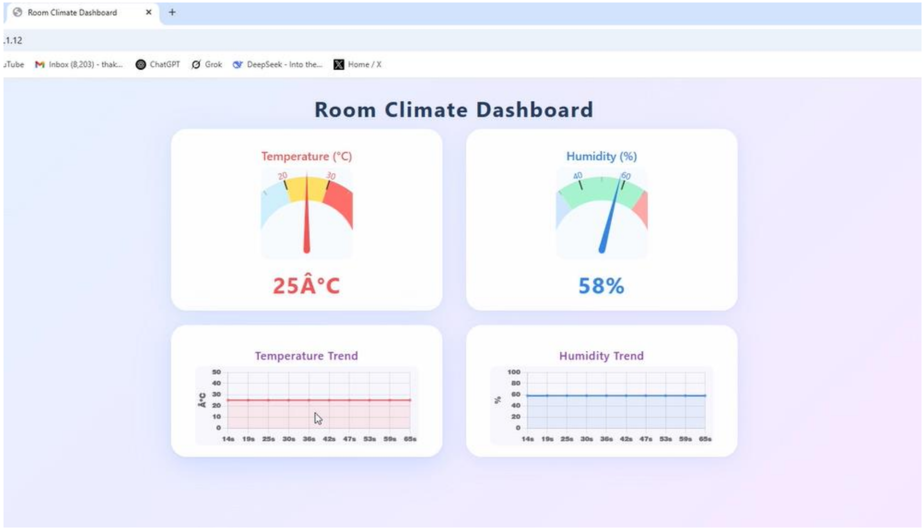
6.1 System in Action

- **Sensors send measurements every 5 seconds to aggregator.**
- **Aggregator computes the average of all received sensor data every 30 seconds and forwards to the edge server.**
- **Edge server displays real-time data and historical trends on a web dashboard.**

Example Hardware Setup



Dashboard Screenshot



7. Discussion

- **Resilience:** If a sensor or aggregator goes offline, the system continues to function for other nodes.
- **Flexibility:** Easy to expand the network by adding more sensors or aggregators.
- **Edge Analytics:** Aggregators can run basic analytics (e.g., outlier filtering, event detection) before sending to the server.
- **Security:** Each layer can implement its own authentication and encryption.

8. Conclusion

A three-layer wireless communication architecture using ESP32 enhances the robustness, scalability, and efficiency of IoT edge networks. By leveraging hierarchical communication, local aggregation, and edge processing, such systems are well-suited for real-time environmental monitoring, industrial IoT, and smart building applications.