International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Edge-Enabled Smart Chairs with Obstacle Avoidance and MQTT Coordination in IoT Environments

Anmol Kumar[1], Joshua Atulya Lakra[2]

[1,2]Karunya Institute of Technology and Sciences
[1]anmolkumar@karunya.edu.in, [2]joshuaatulya@karunya.edu.in

Abstract
In modern smart environments, automation of office furniture ensures greater efficiency, organization, and energy efficiency. The paper presents the design of an Android-based IoT smart chair system with real-time obstacle avoidance, multi-chair synchronization, and power-efficient control to enable completely autonomous self-parking chairs. Unlike traditional systems that use basic triggers, the design leverages ultrasonic and infrared sensors for environmental sensing and accurate navigation. The chairs are connected using the MQTT communication protocol to allow them to coordinate their movements through a local edge computing hub, minimizing latency and maximizing responsiveness.

Every chair has the ability to perceive its environment and drive around moving objects, while at the same time coordinating with neighboring units to ensure safe parking without crashes. To conserve energy, the system makes use of adaptive sleep states and low-power routing algorithms, allowing for optimized power consumption during idle times or path planning. Powered through a custom-developed Android app, users can initiate and monitor functions remotely, thus allowing for scalability and easy user experience. This sophisticated self-parking chair system is well adapted for automated conference rooms, laboratories, and collaborative spaces, thus contributing to the creation of energy-efficient smart infrastructure.

1. Introduction

In the modern age of smart environments, automation of workspace infrastructure, including smart furniture, has seen significant progress. Meeting rooms and shared workspaces are increasingly being designed to reduce manual intervention, maximize organization, and improve energy efficiency. In this world of innovation, self-parking intelligent chairs represent a new technology to enable the creation of fully automated and adaptable interior spaces.

Current solutions tend to be based on intrinsic triggers, for instance, sound recognition (clapping) or remote-control processes, which lack real-time situational information or coordination among devices. These systems are limited in scalability, precision, and power efficiency.

To provide for the requirements of intelligent environments, the integration of technologies to facilitate autonomous navigation, cooperative coordination, and energy-awareness is crucial. Recent IoT automation and edge computing research presents encouraging prospects: edge computing-based architectures provide low latency response suitable for real-time applications, hence reducing reliance on cloud resources and improving reliability and privacy [1].

Power-efficient IoT paradigms, such as adaptive sleep modes and duty cycling, substantially reduce power consumption in embedded systems [2]. Multi-agent coordination using MQTT protocols also enables lightweight and scalable device synchronization in localized IoT environments [3], while ultrasonic sensing coupled with infrared modules has proven effective in enabling real-time obstacle detection in mobile robotics [4]. However, there exists a research void in the development of a comprehensive system integrating real-time obstacle avoidance, co-synchronized multi-device parking, as well as energy-efficient operation for smart chairs simultaneously.

Most of the existing systems address mostly one or two such aspects, thus failing to have the all-round approach that is suitable for interactive and scalable environments.

This article presents an Android-controlled, IoT-based autonomous parking chair system that integrates three essential elements:

Ultrasonic and infrared sensor obstacle detection for secure travel,
Multi-chair synchronization through MQTT and edge-computing hubs to provide collision-free and optimal parking, Power-saving operation through sleep scheduling and optimal movement tactics.

The objective is to make conference rooms more automated by lowering energy consumption and optimizing user comfort through an intelligent and scalable furniture system.

2. Related Work

The creation of smart furniture in smart spaces tends to spark a number of advancements in automation, navigation, power consumption, and control systems. Many autonomous or self-navigating chairs, self-propelled robots, and smart device coordinating systems have been created with limited automation. Most systems are, however, not yet fully combined with real-time obstacle avoidance, multi-device coordination, and power-efficient operation into one platform.

The majority of initial applications were developed with simple triggering mechanisms, such as infrared remotes and acoustic sensors, that were not environment-aware. Additional research moved on to using ultrasonic and infrared sensors to provide more responsive outputs. For example, robotic wheelchairs and mobile platforms with bumping obstacle detection systems were researched for healthcare and service industry uses [1][2]. These types of systems were usually standalone with minimal coordination logic.

Meanwhile, IoT control systems emerged, enabling wireless communication between devices using protocols like MQTT and CoAP [3]. However, the capabilities of MQTT in coordinating numerous independent devices—especially in edge conditions—are not being maximized in furniture automation. Similarly, Android-based control panels have been proposed for mobile robots [4], but not applied extensively in coordinated chair systems.

The following table summarizes selected prior work and highlights the distinctions of the proposed system:

Research Work / System	**Features Implemented**	**Limitations**
Kim et al., 2020 [1]	Obstacle detection using ultrasonic sensors in mobile robots	No coordination between multiple units, no energy-saving logic
Thangavel et al., 2014 [3]	MQTT for IoT device communication	Lacks real-time motion coordination and application in robotics
Android-controlled robotic wheelchair [4]	Android interface, movement control	No obstacle avoidance, no multi-device integration
Edge-based robotic navigation systems [5]	Obstacle avoidance + edge processing	Not designed for synchronized, coordinated environments
Sound-controlled smart chairs (commercial concepts)	Clap-triggered chair movement	Lacks sensing, no intelligence, no coordination, energy inefficient
Proposed System (This Paper)	Android control, obstacle avoidance, MQTT multi-chair sync, power-aware edge IoT	Unified platform, optimized path planning, real-time safety

While the literature reveals progress in individual domains—such as navigation, control interfaces, and communication protocols—there is a **clear lack of unified systems** that integrate:

- Real-time **obstacle detection**,
- Efficient **multi-device synchronization** using MQTT,
- **Power optimization** for sustained, scalable operation,
- And a **user-friendly Android interface** for real-time control.

The proposed solution bridges these gaps by offering a modular, coordinated, and intelligent self-parking chair ecosystem designed for smart collaborative spaces such as conference rooms and labs.

3. Methodology

This section elaborates the systematic approach followed in designing, developing, and testing the Self-Parking Smart Chair System. The methodology involves hardware selection, software development, communication protocol integration, real-time sensing and motion control, and performance optimization for multi-chair coordination.

3.1 System Overview

The smart chair system comprises a microcontroller-based platform (e.g., ESP32 or NodeMCU), ultrasonic sensors for obstacle detection, geared motors with driver modules, and a rechargeable battery. The system receives commands via an Android application over the MQTT protocol and executes real-time self-parking, coordinated movement, and collision avoidance.

3.2 Hardware Architecture

Component	Description
Microcontroller	ESP32/NodeMCU for Wi-Fi, MQTT, and real-time processing
Ultrasonic Sensors	HC-SR04 used for front, side, and rear distance detection
Motor Driver	L298N to control bi-directional movement of geared motors
DC Motors	Two geared motors (12V/100RPM) for motion control
Power Supply	12V rechargeable battery with onboard power regulation
Wheels & Frame	Custom mobile chair frame with caster for balance and rotation

3.3 Software Architecture

3.3.1 Android Application

- Built using Java/Kotlin
- Interface provides:
  - Manual Directional control (forward, back, left, right, stop)
  - Smart Parking Trigger
  - Chair status monitoring (e.g., parked/active, distance to object).
- Sends JSON payload via MQTT to chair microcontrollers.

3.3.2 Microcontroller Firmware

- Programmed using Ardiuno IDE.

- Responsibilities:
  - Connect to WiFi and subscribe to MQTT topics.
  - Parse commands and execute motion control logic.
  - Read real-time data from ultrasonic sensors.
  - Send acknowledgements/status to MQTT broker.

3.3.3 MQTT Communication

- Lightweight publish/subscribe protocol.
- Broker: Mosquitto hosted locally via cloud (e.g., HiveMQ).
- Topics:
  - chairX/control: receives control signals.
  - chairX/status: publishes sensor data and action results.
  - Chair/coordination: used for avoiding overlap or collision in multi-chair systems.

3.4 Edge Computing & Obstacle Avoidance

The system incorporates edge processing at the microcontroller level for real-time decisions without cloud dependence:

- Ultrasonic sensors are read every 200ms.
- Obstacle is flagged if distance < 20 cm.
- Upon detection, movement is halted and route is re-evaluated.
- Each chair is assigned a unique MQTT ID and coordinates its movement via inter-chair signalling.

```
Algorithm 1: BEGIN

  Initialize all ultrasonic sensors (Front, Back, Left, Right)
  Initialize motor driver pins (IN1, IN2, IN3, IN4)
  Set target distances: targetFront, targetBack, targetLeft, targetRight
  Set tolerance value

  LOOP FOREVER
    Measure frontDist, backDist, leftDist, rightDist using ultrasonic sensors

    IF |frontDist - targetFront| > tolerance THEN
      IF frontDist > targetFront THEN
        Move Forward
      ELSE
        Move Backward
      ENDIF

    ELSE IF |leftDist - targetLeft| > tolerance OR |rightDist - targetRight| > tolerance THEN
      IF leftDist > targetLeft AND rightDist < targetRight THEN
        Turn Left
      ELSE IF rightDist > targetRight AND leftDist < targetLeft THEN
```

```
        Turn Right
      ENDIF

    ELSE
      STOP movement
      Display "Target Position Reached"
    ENDIF

    Wait for short delay (e.g., 200 ms)
  END LOOP

END
```

Power Efficiency Measures

To conserve battery:

- Chairs enter **idle mode** after 3 seconds of no command.
- Sensor read frequency drops in idle mode.
- Low battery alert is triggered to the app via MQTT.

3.5 Multi-Chair Coordination Logic

Using MQTT topics and real-time location approximation (e.g., from ultrasonic proximity to known walls or docking points):

- Each chair waits for its turn if another chair is currently moving.
- A shared MQTT topic chairs/coordination flags which chair is active.
- Deadlock is avoided using simple **token-based logic**:

```
 Algorithm 2: Multi-Chair Synchronization
IF token is free THEN
  Acquire token
  Publish status = "Active"
  Execute movement
  Release token
  Publish status = "Idle"
ELSE
  Wait until token is free
ENDIF
```

3.6 Safety & Fail-Safes

- Emergency stop button on the top app overrides all actions.
- If obstacle is within 10cm, buzzer alert is triggered and movement is blocked.
- Status LED indicators (green=moving, red=blocked,blue=idle).

3.7 Flow Diagram

[Start]
  ↓
[Wi-Fi & MQTT Connect]

↓
[Listen to MQTT Control Topic]
↓
[Receive Command]
↓
[Check Obstacle]
→ If Yes → [Avoid / Stop / Turn]
↓
[Move Chair]
↓
[Send Status Update]
↓
[Wait / Listen Again]
↓
[End]

Tools & Technologies Used

Tool/Tech	**Use Case**
ESP32/NodeMCU	Main processing unit
MQTT (Mosquitto)	Publish-subscribe communication
Android Studio	Android application development
Arduino IDE	Firmware programming
Google Firebase (Optional)	Logging & user data (future enhancement)

4. Results and Discussion

Although the proposed system—**Edge-Enabled Android-Based Smart Chairs Using MQTT for Efficient Multi-Chair Coordination with Obstacle Avoidance and Power Optimization**—has not yet been physically implemented or tested, its design addresses several critical limitations observed in current smart chair systems and traditional conference room automation.

Anticipated Benefits and Improvements

1. Multi-Chair Coordination via MQTT:
   Leveraging MQTT as the communication protocol offers lightweight, low-latency messaging between Android devices and microcontroller-equipped chairs. This enables real-time coordination

of multiple chairs simultaneously, eliminating the delays or interference issues seen in IR- or Bluetooth-based systems.

2. Obstacle Avoidance Integration:
   Most existing automated chairs rely on predefined paths or manual guidance. The proposed system utilizes ultrasonic sensors with real-time edge processing to detect obstacles and re-route accordingly, thus minimizing collisions and ensuring safe, autonomous movement.

3. Power Optimization Techniques:
   Edge computing allows local decision-making and reduces communication overhead with central servers, thereby saving energy. Additionally, intelligent sleep-mode triggering when chairs are idle and efficient motor control reduce the overall power consumption—a known bottleneck in many current robotic furniture systems.

4. Scalability and Fault Tolerance:
   Since MQTT supports publish-subscribe architecture, it allows scalability without significant redesign. Any number of chairs can join the system without centralized bottlenecks. Moreover, MQTT supports message queuing, ensuring reliability even under network disturbances.

Theoretical Comparison with Existing Solutions

Feature	**Existing Smart Chairs**	**Proposed System**
Communication Protocol	IR / Bluetooth	MQTT (Publish-Subscribe)
Multi-chair Coordination	Manual or Sequential	Synchronized via Android MQTT Controller
Obstacle Avoidance	Limited or Absent	Real-time Ultrasonic Sensor + Edge Processing
Power Management	Continuous operation	Sleep Mode + Efficient Path Planning
Integration with Mobile Devices	Minimal	Android-based control + feedback interface
Network Scalability	Low	High via MQTT broker

Discussion of Impact

If implemented, this solution would mark a significant shift in smart infrastructure design—especially in high-density environments like conference halls, educational institutions, and labs. It aligns with the growing push toward **IoT-Edge convergence**, ensuring low-latency, energy-efficient automation systems.

Furthermore, by providing a modular and extensible architecture, the design supports future extensions such as voice control, RFID-based user identification, and machine learning-driven path prediction. Such flexibility ensures that this system stays relevant in rapidly evolving smart environments.

5. Conclusion and Future Work

Conclusion

This research presents a conceptual design for an **Edge-Enabled Android-Based Smart Chair System** using **MQTT** for efficient multi-chair coordination, integrated **obstacle avoidance**, and **power optimization**. The proposed solution addresses key shortcomings in current smart furniture systems, such as limited mobility, poor scalability, inefficient communication protocols, and high energy consumption.

By integrating **Android control**, **ultrasonic sensors**, **edge computing**, and a **lightweight MQTT protocol**, this architecture offers a highly adaptable, modular, and efficient approach to smart infrastructure automation. The focus on decentralized edge processing significantly reduces latency and power usage, while MQTT's publish-subscribe mechanism ensures scalability and reliable chair coordination.

Though this work is not supported by real-world deployment at this stage, it provides a strong foundation for future development, implementation, and testing in real environments like classrooms, smart offices, and conference halls.

Future Work

To advance this proposed system from concept to real-world implementation, several future directions are envisioned:

1. **Prototyping and Testing**
   - Develop a functional prototype using Arduino or ESP32 microcontrollers.
   - Test multi-chair coordination, obstacle avoidance efficiency, and power consumption metrics in various room setups.
2. **Integration with AI and ML Models**
   - Use machine learning to predict chair usage patterns and optimize movement paths dynamically.
   - Introduce voice or gesture-based commands via AI-enabled Android interfaces.
3. **User Identification and Personalization**
   - Implement RFID or facial recognition systems to auto-adjust chair settings based on user profiles.
4. **Energy Harvesting Solutions**
   - Explore incorporating solar panels or kinetic energy systems to enhance the sustainability of the smart chair.
5. **Security Enhancements**
   - Add encryption to MQTT communication and secure access controls on Android devices to prevent unauthorized use or attacks.
6. **Cross-Platform Application**
   - Expand support to iOS or web dashboards for broader user control and monitoring.

This work lays the groundwork for a next-generation smart chair ecosystem, bridging gaps in automation, safety, and energy-aware design. With further development and validation, it holds the potential to revolutionize automated furniture in smart spaces.

References

1. F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things," 2012.
2. A. e. a. Zanella, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, vol. 1, no. 1, p. 22–32, 2014.
3. N. a. A. S. Bakar, "Obstacle avoidance for mobile robot with ultrasonic sensors," in Proc. IEEE Int. Conf. Mechatronics, 2011.
4. E. e. a. Koutroulis, "Development of a microcontroller-based, photovoltaic maximum power point tracking control system," IEEE Trans. Power Electronics, vol. 16, no. 1, p. 46–54, 2001.
5. M. R. e. a. Palattella, "Standardized Protocol Stack for the Internet of (Important) Things," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, p. 1389–1406, 2013.
6. Y. e. a. Elkhatib, "The role of cloud computing in smart environments," IEEE Internet Computing , vol. 17, no. 2, p. 12–19, 2013.
7. B. e. a. Naresh, "Design of a Smart Chair using Arduino," IJARCSSE, vol. 7, no. 4, p. 89–93, 2017.
8. L. e. a. Atzori, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, p. 2787–2805, 2010.
9. M. e. a. Abu Alsheikh, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, p. 1996–2018, 2014.
10. T. a. B. M. Black, "Design of an MQTT-Based Architecture for Smart Office Automation,," in 2021, Intl. Conf. on IoT and Smart Infrastructure.