COMP413: Internet of Things

‘Smart Parking System’ Project Report

Instructor

Asst. Prof. Abdulkadir KÖSE

Group Members

Tolgahan Keleş - 2311051801

Muhammet Çağrı Akkuş - 2211051034

Umut Kaya - 2111051064

Submission Date

12.01.2025

ABDULLAH GUL UNIVERSITY

KAYSERI, 2025

Table of Contents

1. Introduction

Problem Statement

Efficient parking management has become a critical issue in modern urban areas. With increasing urbanization, the number of vehicles on the road has surged, leading to overcrowded parking lots and streets. Drivers spend significant amounts of time searching for available parking spaces, which not only causes traffic congestion but also results in increased fuel consumption and air pollution. The lack of real-time parking information exacerbates this issue, making it difficult to optimize parking space usage.

These inefficiencies have far-reaching implications, including economic losses due to wasted time and resources, environmental damage caused by greenhouse gas emissions, and heightened stress levels for drivers. To address these challenges, this project introduces a Smart Parking System leveraging IoT technology to streamline parking management and improve urban mobility.

Objectives

The primary objective of the Smart Parking System is to provide a scalable, efficient, and real-time solution for parking space management. By integrating IoT devices with cloud-based platforms, the system aims to:

- Minimize the time drivers spend searching for parking spots, thereby reducing traffic congestion.

- Lower fuel consumption and greenhouse gas emissions, contributing to a cleaner environment.

- Enhance user convenience through a mobile application and web interface that display real-time parking availability.

- Develop a modular architecture that can be easily scaled to accommodate large urban areas.

Scope

This project focuses on the design and implementation of a smart parking solution utilizing ultrasonic sensors, ESP32 microcontrollers, and a Firebase backend. The system is intended for deployment in urban areas, where efficient parking management can significantly impact traffic flow and environmental sustainability. The solution is designed to be adaptable for future integrations, such as payment systems and advanced analytics.

2. System Model

Overall Design

The Smart Parking System employs a three-layer architecture, ensuring modularity and efficiency. Each layer is designed to perform specific functions, enabling seamless communication and real-time data processing.

1. Sensing Layer:

The sensing layer comprises ultrasonic sensors connected to ESP32 microcontrollers. These sensors detect the presence or absence of vehicles in parking spaces by measuring the distance between the sensor and potential obstructions. The ESP32 microcontrollers process the sensor data locally and transmit state changes (e.g., from empty to full) to the gateway device. This approach minimizes redundant communication and conserves power.

2. Communication Layer:

The communication layer is anchored by an ESP32-S2 Mini device, which serves as the gateway. This device aggregates data from multiple sensor nodes and forwards it to a Firebase database. The gateway ensures reliable and efficient communication between the sensing layer and the application layer.

3. Application Layer:

The application layer includes a Flutter-based mobile application and a web interface. These platforms retrieve data from Firebase and present real-time parking availability to users via interactive maps. The application layer prioritizes user experience, offering intuitive navigation and seamless access to parking information.

3. Hardware Design

Components Used

The hardware components of the Smart Parking System were selected for their reliability, cost-effectiveness, and compatibility with IoT applications. Key components include:

- **Ultrasonic Sensors (HCSR04)**: These sensors measure the distance to nearby objects, enabling the detection of vehicle presence in parking spaces. They are capable of accurately detecting distances within a range suitable for parking applications.
- **ESP32 and ESP32-S**: These microcontrollers handle data processing and communication. They are equipped with Wi-Fi capabilities, making them ideal for IoT applications.
- **ESP32-S2 Mini (Gateway)**: Acting as the central communication hub, this device collects data from the sensor nodes and sends it to the Firebase database. It ensures low-latency communication and robust data handling.

Configuration

The system's hardware configuration involves connecting ultrasonic sensors to ESP32 devices, which are programmed to monitor parking spaces. Each sensor node operates independently, sending updates to the gateway only when a change in parking status occurs. The gateway is configured to aggregate this data and transmit it to Firebase, ensuring real-time updates for the application layer.

The system recognizes three states for each parking spot:

- Empty: Indicates the parking spot is available.
- Full: Indicates the parking spot is occupied.

- Unknown: Represents inconclusive sensor data due to environmental factors or hardware limitations.

4. Software Design

Mobile Application

The mobile application serves as the user interface for the Smart Parking System. Developed using Flutter, the app provides real-time updates on parking availability through an interactive map. The `flutter_map` package is used to display parking lots, making it easy for users to locate available spots. Additionally, the app integrates Firebase to ensure seamless data synchronization and reliability.

Cloud Backend

Firebase is utilized as the backend platform for data storage and real-time synchronization. It acts as the central repository for parking space data, allowing both the mobile application and web interface to access updates simultaneously. Firebase's robust infrastructure ensures low latency and high availability, making it ideal for this IoT application.

Algorithms

- **Distance-Based Detection:**

The system uses ultrasonic sensors to measure distances and determine whether a parking spot is occupied. Threshold values are set to classify each spot's status as empty, full, or unknown.

- **State Change Detection:**

To minimize redundant communication, sensor nodes transmit data only when there is a change in state (e.g., from empty to full). This approach reduces network traffic and conserves power, enhancing system efficiency.

TinyML Integration

Future iterations of the project aim to incorporate TinyML for on-device data processing. By enabling local analysis of sensor data, TinyML can further reduce latency and enhance system performance.

5. Results and Discussion

Key Outcomes

The Smart Parking System successfully demonstrates the feasibility of IoT-based parking management. Real-time monitoring of parking spaces was achieved, and the mobile application and website effectively display parking availability. The system's modular architecture ensures scalability and adaptability for various urban environments.

Challenges

During the development process, several challenges were encountered:

- **Environmental Factors:** Rain, uneven surfaces, and other environmental conditions occasionally affected sensor accuracy, leading to inconclusive data.

- **Communication Optimization**: Balancing power consumption and communication latency required careful tuning of the system's algorithms and hardware configurations.

Future Scope

The project holds significant potential for future improvements and expansions:

- Enhance sensor accuracy to address environmental challenges.

- Deploy multiple gateways to support large-scale applications in densely populated urban areas.

- Integrate advanced features, such as payment systems for reserving parking spaces and predictive analytics for demand forecasting.

6. Sustainable Development Goals (SDGs) Alignment

SDG 11: Sustainable Cities and Communities

By optimizing parking management, the Smart Parking System contributes to reduced traffic congestion and promotes sustainable urban mobility. It aligns with the goal of creating inclusive, safe, and resilient cities.

SDG 13: Climate Action

The system minimizes greenhouse gas emissions by reducing the time and fuel consumed during parking searches. This aligns with global efforts to combat climate change and mitigate its impacts.

8. How to Set Up

Hardware Setup

1. Connect HCSR04 sensors to ESP32 devices.

2. Configure the ESP32-S2 Mini as the gateway device.

3. Ensure all devices are connected to the gateway network.

Software Setup

1. Clone the repository.

2. Install required libraries for ESP32 microcontrollers using the Arduino IDE.

3. Deploy the Firebase configuration to the ESP32 gateway.

4. Set up the Flutter mobile application

9. Live Demo

- **Video Demo:** https://0x0.st/s/3riFpAlwlm9EzLAOGXGpZg/8-B9.webm

- **Live Website**: https://iot-project-db90a.web.app/