**Design and Deployment of an IoT-Enabled Car Parking System with Raspberry Pi Integration for Seamless Real-Time Status Updates on Mobile Applications via MQTT Protocol**

Sucharita Gorai, Nikhilesh Kumar Mohanta, Prem Kumar Khuntia, Dibyajyoti Sahoo,

Ruchismita Bastia, Suryasminu Nayak, Aryan Mohanty, Muskan Moharana,,

Biswaranjan Swain

Centre for Internet of Things, Computer Science Engineering, Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar 751030, India

**Abstract.**

The project is dedicated to designing and deploying an IoT-enabled car parking system that integrates Raspberry Pi technology with the MQTT protocol to offer real-time status updates via mobile applications. This system leverages Raspberry Pi and MQTT to enhance parking management by providing drivers with up-to-date information on parking space availability. The goal is to improve convenience and efficiency in finding parking, ultimately streamlining the parking experience for users and optimizing the use of parking facilities. It enhances user convenience and satisfaction while offering valuable insights to parking lot operators for optimized management and planning. Additionally, the system is designed to be scalable and adaptable, making it suitable for a wide range of parking environments, from small-scale to large-scale applications.Top of Form

Bottom of Form

**Keywords**: IoT-enabled car parking, Raspberry Pi, MQTT protocol, Mobile applications, Parking management.

1. Introduction

With the rapid growth in urban populations and the increasing number of vehicles, efficient parking management has become a critical challenge. Traditional parking systems often result in congestion, wasted time, and increased pollution due to prolonged vehicle searches for available spots. In response, Internet of Things (IoT) technology provides a transformative approach to developing a smart car parking system that can optimize parking space utilization and improve the user experience.

This paper introduces an IoT-enabled smart car parking system that utilizes the Message Queuing Telemetry Transport (MQTT) protocol, a lightweight messaging protocol ideal for IoT applications. The proposed system employs sensors, such as infrared (IR) or ultrasonic sensors, to detect the availability of parking spots in real-time. These sensors communicate data regarding the occupancy status to a central server via MQTT, allowing efficient, low-latency transmission over wireless networks. The server processes this information and updates a mobile or web application, enabling users to check for available parking spaces and reserve them remotely.

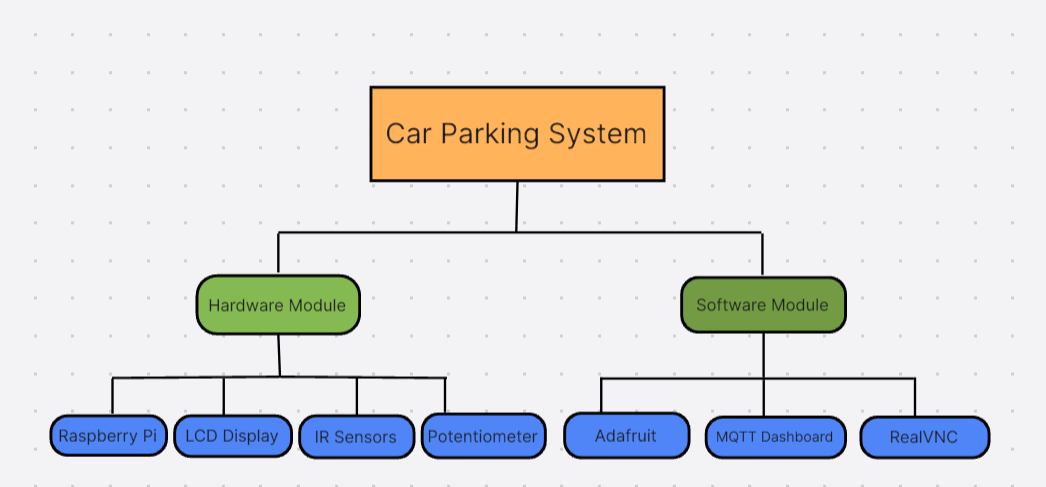
The IoT-based smart parking system integrates adaptive algorithms to handle dynamic parking environments and provides continuous updates to users. The MQTT protocol is well-suited for this application due to its minimal bandwidth requirements, efficient message delivery, and support for various network conditions. The system not only reduces the time spent by drivers searching for parking but also minimizes traffic congestion and carbon emissions in urban areas.

The use of MQTT enhances the scalability and flexibility of the parking system, enabling it to operate seamlessly in diverse and unpredictable environments. This approach allows the system to learn from real-time data, identify patterns, and optimize parking management dynamically. The paper also demonstrates a prototype implementation using MQTT for data exchange and discusses the results of simulations, focusing on key performance metrics such as response time, data accuracy, and network efficiency. Future work will explore integrating advanced machine learning algorithms to further improve the system's adaptability to changing conditions, such as varying traffic patterns or evolving user preferences.

1. System Architecture

The architecture is divided into two main modules: **Hardware Module** and **Software Module**.

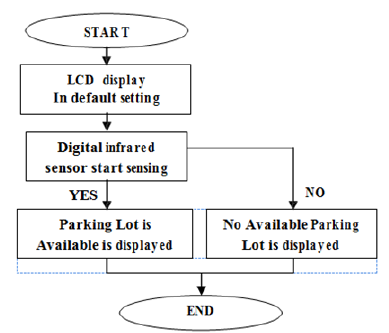
1. **Hardware Module:**
   * **Raspberry Pi:** Acts as the central processing unit for the system, handling data collection, processing, and communication with the cloud server.
   * **LCD Display:** Provides real-time information about parking space availability to the users at the parking site.
   * **IR Sensors:** Installed in each parking space to detect whether a parking slot is occupied or vacant. The data from these sensors is sent to the Raspberry Pi.
   * **Potentiometer:** Used for adjusting the sensitivity of the IR sensors or calibrating the sensor values.
   * **Adafruit:** A component used for integrating sensor data with the Raspberry Pi and transmitting the data to the cloud.
2. **Software Module:**
   * **MQTT Dashboard:** Used for communication between the hardware and software components. It provides a platform for transmitting data from the Raspberry Pi to a cloud server or application using the MQTT protocol.
   * **Real VNC:** A remote access tool that allows monitoring and controlling the Raspberry Pi from a remote location for maintenance or management purposes.



**Figure : Layout of project module**

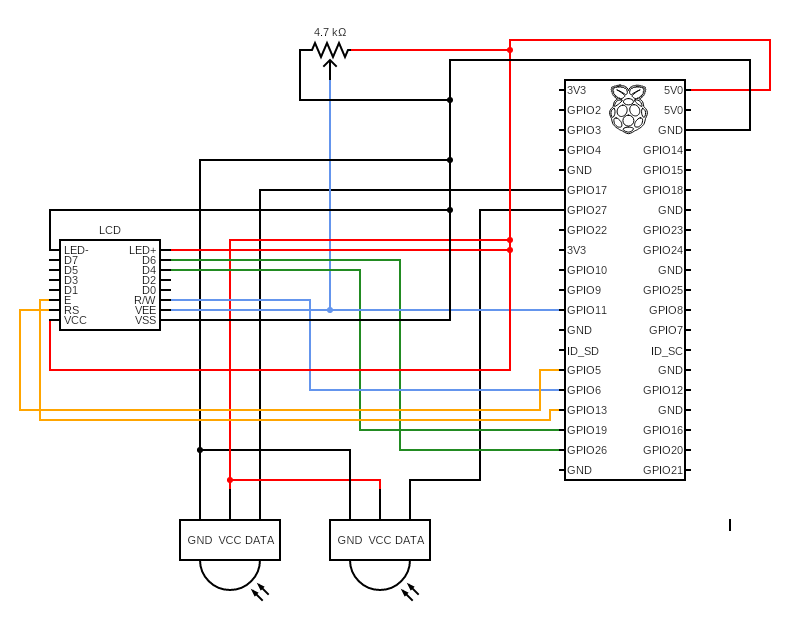
**System Workflow:**

1. The system starts by initializing the **LCD Display** with default settings. The **Digital Infrared (IR) Sensor** begins sensing the parking slots. If a parking lot is available, a message indicating its availability is displayed on the LCD. If no parking slot is available, a corresponding message is displayed. This data is then transmitted through the MQTT protocol to the server, and users can access it via the mobile or web application.
2. This architecture provides a real-time, efficient, and automated solution for managing car parking spaces using IoT and MQTT.



**Logic and Flowchart**

1. Block diagram of the proposed system



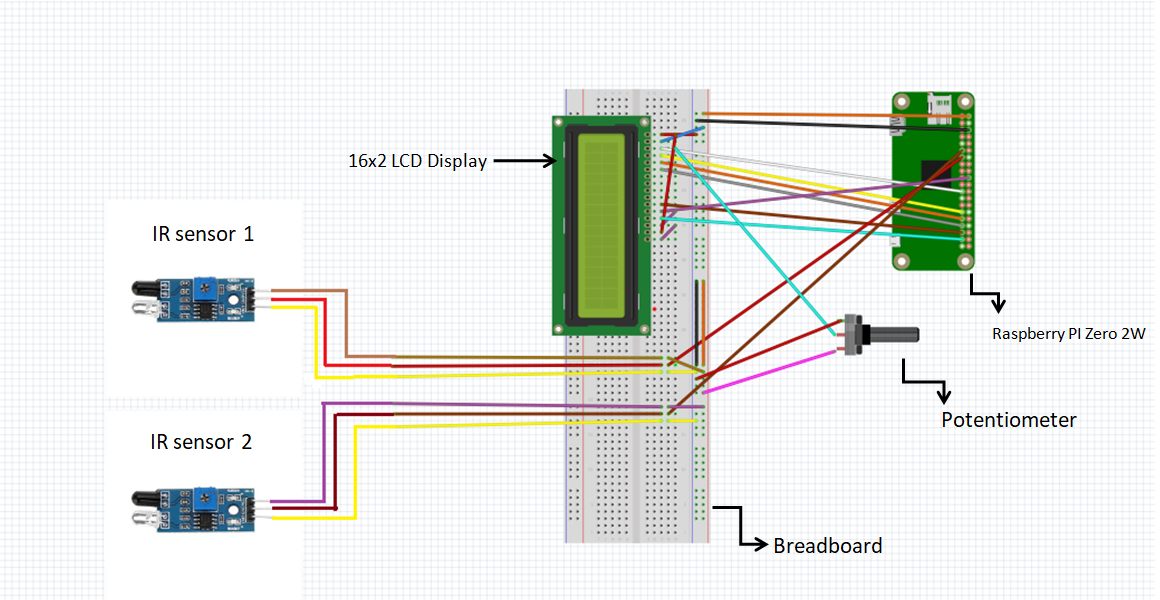
The IoT-based car parking system operates by integrating IR sensors, microcontrollers, and MQTT protocol to provide real-time monitoring and management of parking spaces. Here’s a step-by-step explanation of how the system works:

* Detection of Vehicles: - IR Sensors: Each parking spot is equipped with an IR sensor. These sensors emit infrared light, which gets reflected back when a vehicle is present. The change in the reflected signal indicates the presence or absence of a vehicle in the spot.
* Data Collection:- Microcontrollers: The IR sensors are connected to microcontrollers, such as a Raspberry Pi Zero 2 W. The microcontroller reads the signals from the IR sensors to determine the occupancy status of each parking space.
* Data Transmission:- MQTT Protocol: The microcontroller uses the MQTT (Message Queuing Telemetry Transport) protocol to transmit the occupancy data to a central server. MQTT is chosen for its lightweight and efficient communication, ideal for IoT applications. MQTT Broker: An MQTT broker facilitates the communication by receiving the data from the microcontroller and routing it to the appropriate clients.
* Data Processing and Storage: - Central Server: The central server aggregates the data received from all the microcontrollers. It processes this data to maintain an up-to-date status of the parking lot.
* User Interface:- Dashboard: A web-based or mobile application dashboard displays real-time information about the availability of parking spots. This dashboard provides a visual representation of the parking lot, indicating which spaces are occupied and which are free.
* User Access: Users can access the dashboard via their smartphones or computers to find available parking spaces quickly.
* Real-Time Updates:- Notifications: The system can send real-time notifications to users about the availability of parking spots or changes in the status of specific areas.
* Predictive Analytics: By analyzing historical data, the system can predict peak usage times and suggest optimal parking strategies.

Benefits:

* Efficiency: Reduces the time drivers spend searching for parking spots.
* Optimization: Enhances the utilization of parking spaces.
* Convenience: Provides real-time updates and easy access to parking information.
* Scalability: Can be scaled to accommodate large parking facilities.
* Environmental Impact: Reduces vehicle emissions by minimizing idle time searching for parking.

### **Circuit Diagram**



1. Pseudocode
2. **Initialize GPIO and LCD Settings:**
   * Set the GPIO mode to BOARD and disable warnings.
   * Define pins for the LCD display and sensors.
   * Set up the GPIO pins for LCD and sensors (slot 1 and slot 2).
3. **Initialize MQTT Communication:**
   * Set up MQTT client and assign event callbacks for connecting and publishing messages.
   * Parse the MQTT broker URL and establish a connection.
4. **Define LCD Functions:**
   * lcd\_init(): Initialize the LCD with required commands.
   * lcd\_byte(bits, mode): Convert byte data to bits and send to LCD.
   * lcd\_toggle\_enable(): Toggle the enable pin for LCD.
   * lcd\_string(message, line): Print a message on the LCD screen.
5. **Display Initial Message on LCD:**
   * Call lcd\_init() to initialize the LCD.
   * Display "Welcome" and "Car Parking System" messages.
6. **Start Monitoring Sensor Inputs:**
   * Enter an infinite loop to continuously check sensor status:
     + Read input status of slot1\_Sensor and slot2\_Sensor.
     + If slot 1 is occupied, display "Slot1 Parked" on the LCD and publish "1" to MQTT topic "slot1".
     + If slot 1 is free, display "Slot1 Free" and publish "0" to MQTT.
     + Repeat similar checks for slot2\_Sensor.
7. **Repeat the Loop:**
   * Continue to monitor and update the LCD and MQTT topics based on sensor inputs.
8. Algorithm
9. **Setup Phase:**
   * Set GPIO mode and warnings.
   * Define and set up GPIO pins for LCD and sensors.
   * Initialize the LCD display using lcd\_init() to display a welcome message.
   * Set up MQTT client with connection and publish handlers.
10. **MQTT Setup:**
    * Parse the MQTT broker URL.
    * Connect to the MQTT broker using the parsed URL.
11. **Loop Phase:**
    * Start an infinite loop to:
      1. Call mqttc.loop() to handle MQTT communication.
      2. Read the state of slot1\_Sensor:
         + If occupied (False), display "Slot1 Parked" on the LCD and publish "1" to MQTT topic "slot1".
         + If free (True), display "Slot1 Free" and publish "0" to MQTT.
      3. Repeat similar steps for slot2\_Sensor.
      4. Introduce a delay between readings.
12. **Repeat:**
    * Continue the loop, repeating the steps to update the LCD display and MQTT broker with the current parking slot statuses.
13. Results and Discussion

* Stage 1: When there is no car in the parking area IR sensor signals that the parking slot is free.



* Stage 2: When car enters the parking area IR sensor that is present before the car, the sensor turns on signalling that the parking slot is occupied.



* Stage 3: When car is parked or not, then the MQTT(Message Queuing Telemetry Transport) dashboard on the mobile phone shows the slot is already is occupied or not.



6 Conclusions

Using digital parking solutions, drivers can log into a web or mobile portal before leaving their homes to receive recommendations for available parking spots near their destination. The adoption of smart parking systems can reduce traffic congestion for Indian drivers. These systems allow users to reserve and pay for parking spaces in advance through an integrated payment system, eliminating the usual hassles of finding a parking spot. Upon arrival at the selected parking location, the system automatically identifies the car and generates an electronic ticket. The associated smartphone app then guides the driver to their designated parking space and assists in exiting the parking lot.

The Internet of Things (IoT) offers many exciting applications, one ofwhich is

smart parking. IoT-based parking systems can more efficiently track available

spaces in a parking lot, making it easier for drivers to find open spots. With the

increasing number of connected devices, IoT-enabled smart parking solutions

are becoming more popular. These systems provide drivers with valuable

information, such as available parking slots, current pricing, and occupancy

rates, enabling a more convenient parking experience.

References

1. <https://www.youtube.com/watch?v=w-9ddDrOlew>
2. <https://www.youtube.com/watch?v=azCb5Zj76UQ>
3. Raspberry Pi Cookbook , Author:- Simon Monk
4. Title: "Smart Parking System using Raspberry Pi and IR Sensors"

Authors: Shashank Vishwakarma, Rahul Kumar, and Tanya Srivastava

1. Sasi Priya S, Sumathi K, Arun Sekar R, Stewart Kirubakaran K, “Efficient Vehicle parking system using Cloud based IoT”, European Journal of Molecular & Clinical Medicine, ISSN 2515-8260 Volume 7, Issue 4, 2020.
2. Thanh Nam Pham1, Ming-Fong Tsai1, Duc Binh Nguyen1, Chyi-Ren Dow1, And Der-Jiunn Deng2. “A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies”, ICTAT Journal on Image and Video Processing, May 2019. [3] Mrs. Mohini Avatade2, Mrs. S. V. Gavhane3, Kedar Bhoyar, “Smart Parking System Using IOT and Cloud Computing”, 2018 IJCRT, Volume 6, Issue 2 April 2018, ISSN: 2320-2882.
3. P. Surekha, Pavan Gurudath, R. Prithvi and V.G. Ritesh Ananth, “Automatic License Plate Recognition Using Image Processing”, ICTACT Journal on Image and Video Processing, May 2018.
4. Tae-Gu Kim, Byoung-Ju Yun, Tae-Hun Kim, Jae-Young Lee, Kil-Houm Park, Yoosoo Jeong and Hyun Deok Kim, “Recognition of Vehicle License Plates Based on Image Processing”, Applied Science Journal-July 2021. Web Url: www.mdpi.com/2076-3417/11/14/6292
5. B. Swain, P P, Nayak, D P, Kar, S. Bhuyan, L. P. Mishra, Review of Scientific Instruments. 87 (2016).
6. https://www.intuz.com/blog/iot-in-smart-parking-management-benefits-challenges