Major Project-II

Car Parking System using Arduino uno

Problem Statement: Manual parking systems are inefficient, causing congestion and time wastage. An IoT-based solution is needed to automate real-time parking space detection and management.

Project Group Number: **27**

Group Member Details:

- Abhishek singh lodhi **(0187CS211011)**
- Aman raj **(0187CS211020)**
- Bharat pal **(0187CS211050)**
- Dipesh **amode** **(0187CS211067)**

Guide Details: PROF SHWETA SINGH (Assistant professor)

Idea/Approach Details

Describe your idea Solution/Prototype here:

This project focuses on developing an IoT-based smart car parking system using an Arduino Uno. The system uses sensors to detect the availability of parking spaces and shares this information in realtime through a connected network. When a car enters the parking area, the system detects an open spot and guides the driver to it. This helps in reducing traffic congestion, saving time, and optimizing parking space usage. The Arduino Uno acts as the central controller, communicating with sensors and transmitting data to a user interface. Additionally, the system can be accessed via a mobile app, allowing users to view available spaces and navigate to their chosen parking spot efficiently.

Project Requirements

Hardware and Software requirements

➢ HARDWARE REQUIREMENTS

- **Arduino Uno:** Acts as the central microcontroller for processing data and controlling the system.
- **Ultrasonic sensor**: Used to detect the presence of a car to determine if a parking space is occupied.
- **Wi-Fi module(ESP8266)**: Enables wireless communication between the Arduino and the MQTT Dashboard.
- **LED display**: Provides visual feedback on the availability of parking spaces.
- **IR sensors**: To detect objects nearby and send signals to an Arduino project.
- **Connecting wires and breadboard**: For connecting the sensors, modules and Arduino.

➢ SOFTWARE REQUIREMENTS

- **Arduino IDE :** For programming the Arduino uno with the necessary code to handle sensors and MQTT
- communication.
- **MQTT Dashboard App** : For visualizing and monitoring the parking space data. Ensure it's set up to connect to your MQTT broker.
- **Bluetooth or Wi-Fi Libraries**: For handling communication between the Arduino and your app interface.

Implementation Plan

Phase 1:

- Define project goals and hardware requirements.
- Acquire and test components (Arduino Uno, sensors, modules).
- Set up the Arduino IDE and MQTT broker.

Phase 2:

- Develop and upload Arduino code for sensor data and MQTT communication.
- Configure the MQTT Dashboard app to display real-time data.
- Integrate and test the system with all its components.

Phase 3:

- Conduct system testing to ensure accurate data reporting.
- Debug and refine hardware and software.
- Optimize the user interface and document the system for development.


Thank you!