INTERNET OF THINGS - Group 1

SMART PARKING

PHASE 3 – DEVELOPMENT PART 1

TEAM MEMBERS:

KARTHICK.P
KARTHICK RAJAN.V.T
SHYAM KUMAR.K
IJAZ AHAMED.K

Project Title: Smart Parking

Phase 3: Development part 1

Sensor configuration:

Smart parking occupancy detection sensors are a crucial component of smart parking systems, providing real-time information about the availability of parking spaces. There are several types of sensors and technologies used for this purpose. Here are some common IoT sensors used for detecting smart parking occupancy:

Ultrasonic Sensors:

- Ultrasonic sensors use sound waves to detect the presence of vehicles in parking spaces. They are typically mounted on a wall or a post near the parking space.

Infrared Sensors:

- Infrared sensors emit an infrared beam across the parking space entrance and detect interruptions in the beam when a vehicle enters or leaves the space.

Magnetic Sensors:

- Magnetic sensors are embedded in the ground of each parking space. They detect the presence of a vehicle by measuring changes in the Earth's magnetic field.

Camera-Based Sensors:

- IP cameras equipped with image recognition software can capture images or video of parking spaces. Advanced algorithms analyze these images to determine parking space occupancy.

LoRa or NB-IoT Sensors:

- Low-power, wide-area network (LPWAN) sensors can transmit occupancy data wirelessly to a central server or cloud platform. They are suitable for outdoor parking areas.

RFID or NFC Sensors:

- RFID or NFC tags can be placed in vehicles, and corresponding sensors are placed near parking entrances and exits. When a tagged vehicle enters or exits the parking facility, the sensors detect it and update the occupancy status.

Inductive Loop Sensors:

- Inductive loop sensors are coils of wire embedded in the road surface at parking space entrances. When a vehicle parks over the loop, it disrupts the electromagnetic field, indicating occupancy.

Acoustic Sensors:

- Acoustic sensors use microphones to listen for the sound of vehicles parking or leaving parking spaces. They can detect changes in noise levels.

Laser Sensors:

- Laser-based sensors use laser beams to detect the presence of vehicles and measure the distance between the sensor and the vehicle. They are highly accurate and can work in various lighting conditions.

WiFi and Bluetooth Sensors:

- WiFi and Bluetooth sensors can detect nearby devices (e.g., smartphones or in-car Bluetooth systems) to determine the presence of vehicles in parking spaces.

UWB (Ultra-Wideband) Sensors:

- Ultra-Wideband technology can precisely locate and track vehicles in real-time, making it suitable for high-accuracy parking space occupancy detection.

Ground Pressure Sensors:

- These sensors are placed under the road surface and detect the weight of a vehicle when it enters a parking space.

Capacitive Sensors:

- Capacitive sensors can be installed under parking spaces and detect changes in capacitance when a vehicle occupies the space.

Microwave Sensors:

- Microwave sensors use radar technology to detect the presence and movement of vehicles in parking spaces. They can cover large areas and work well in adverse weather conditions.

PYTHON SCRIPT:

```
import random

import time

import requests

def simulate_sensors():

    flow_data = random.uniform(1.0, 10.0)  # Simulate flow rate in gallons per minute

    water_quality_data = {

        "pH": round(random.uniform(6.5, 8.5), 2),

        "turbidity": round(random.uniform(0.1, 5.0), 2),

        "dissolved_oxygen": round(random.uniform(4.0, 10.0), 2),

        "contaminant_level": round(random.uniform(0.0, 1.0), 2)

    }

    level_data = random.uniform(0.0, 100.0)  # Simulate water level in percentage

    pressure_data = random.uniform(30.0, 80.0)  # Simulate pressure in psi

    water_meter_data = random.uniform(0.1, 5.0)  # Simulate water consumption in gallons

    water_quality_surveillance_data = {
```

```
        "bacteria_level": round(random.uniform(0.0, 1000.0), 2),
        "virus_level": round(random.uniform(0.0, 100.0), 2)
    }
    return {
        "flow": flow_data,
        "water_quality": water_quality_data,
        "level": level_data,
        "pressure": pressure_data,
        "water_meter": water_meter_data,
        "water_quality_surveillance": water_quality_surveillance_data
    }
def send_data_to_iot_platform(data):
    url = "https://your-iot-api-url.com/data"
    headers = {"Content-Type": "application/json"}
    response = requests.post(url, json=data, headers=headers)
    if response.status_code == 200:
        print("Data sent to IoT platform successfully.")
    else:
        print("Failed to send data to IoT platform. Status code:",
response.status_code)
while True:
    sensor_data = simulate_sensors()
    send_data_to_iot_platform(sensor_data)
    time.sleep(5)
```