**Environmental monitoring in parking system in IOT**

**Introduction:**

Environmental monitoring in a parking system typically involves using various sensors and technologies to collect data, which can then be processed and analyzed to make informed decisions or trigger actions. Implementing a complete system with code can be a complex task, but I can provide a simple example in Python to demonstrate the concept of environmental monitoring within a parking system using a temperature sensor as an illustration. In a real-world scenario, you would need the appropriate hardware and APIs for other sensors and systems.

Environmental monitoring in a parking system refers to the practice of using various sensors, technologies, and data analysis methods to continuously assess and manage environmental conditions within and around a parking facility. The data collected through environmental monitoring systems can be used for real-time decision-making, improved security, energy efficiency, and enhancing the overall user experience within the parking facility. Drivers can access this information through a mobile app or website, or through signs that are posted in the parking lot itself.

**Project description:**

The "Environmental Monitoring in Parking System" project aims to enhance the sustainability and user experience of parking facilities by implementing a comprehensive environmental monitoring system. This system will utilize a network of sensors and advanced technologies to collect, analyze, and manage various environmental parameters within and around parking areas.

Hardware components:

* Sensors
* Data Logger
* Central Control System
* User Interface
* Alarms and Notifications
* Power Supply
* Environmental Enclosures
* Cameras and Video Monitoring
* Display Panels
* Access Control Systems
* Weather Station (Optional)

Software components:

* User Interface (UI)
* Database
* Notification System
* Reporting and Analytics
* Security and Authentication
* GIS Integration
* Administration and Configuration
* APIs

Workflow:

1. **Project Initiation:**

a. **Define Objectives:** Clearly state the goals of your environmental monitoring project. Determine what specific aspects of the parking system you want to monitor, such as occupancy, air quality, security, and lighting.

b. **Budget and Resources:** Establish a budget and allocate resources for the project. Determine the hardware and software components required and assess the available funds.

1. **Site Assessment:**

a. **Survey the Parking Facility:** Conduct a comprehensive assessment of the parking facility. Identify the layout, the number of parking spaces, entry and exit points, and other relevant structural details.

b. **Environmental Analysis:** Analyze the specific environmental conditions that need monitoring, such as air quality and temperature, and consider the locations where sensors and cameras will be placed.

1. **Hardware Selection and Procurement:**

a. **Select Hardware Components:** Choose the appropriate sensors, cameras, access control systems, communication infrastructure, LED displays, and other hardware components based on your project objectives and site assessment.

b. **Vendor Selection:** Identify and select vendors or suppliers for the hardware components. Ensure they meet your requirements in terms of quality and cost-effectiveness.

1. **System Design:**

a. **Hardware Placement:** Create a detailed plan for where each sensor, camera, and other hardware components will be installed within the parking facility. Ensure that they are strategically positioned to achieve your monitoring goals.

b. **Wiring and Connectivity:** Design the wiring and connectivity infrastructure required to link the hardware components to a central control system. Determine whether wireless or wired connections are needed.

1. **Software Development:**

a. **Central Control Software:** Develop or configure software for central control, data collection, and analysis. Ensure the software can integrate data from the selected hardware components.

b. **User Interface:** Design a user-friendly interface for monitoring and managing the system. Include dashboards and reporting tools for real-time data visualization.

1. **Testing and Calibration:**

a. **Functionality Testing:** Test the entire system to ensure that all hardware components, sensors, cameras, and software are working as expected.

b. **Calibration:** Calibrate sensors to provide accurate data. Adjust settings as needed.

1. **Training and Documentation:**

a. **User Training:** Provide training for operators and administrators who will manage and monitor the system.

b. **Documentation:** Create comprehensive documentation that includes system manuals, maintenance procedures, and troubleshooting guides.

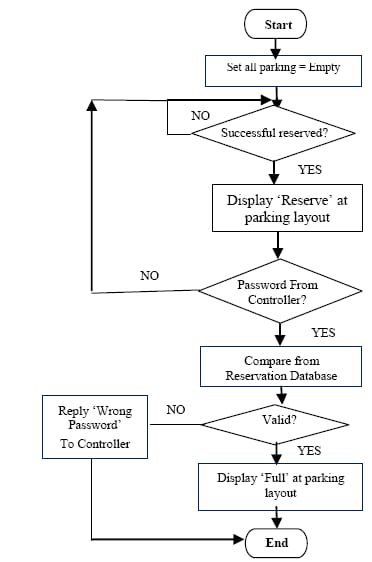
1. **Deployment and Monitoring:**

a. **System Launch:** Deploy the system for full-time operation.

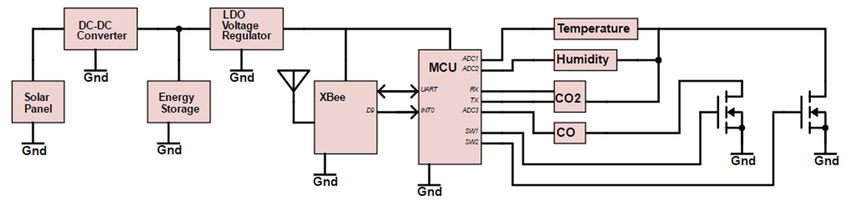
b. **Ongoing Monitoring:** Continuously monitor the system to ensure it operates effectively and meets the project objectives.

Top of Form

Flowchart:



Circuit diagram:



Program:

int value\_sensor = 0;

void setup()

{

pinMode(A1, INPUT);

Serial.begin(9600);

pinMode(6, OUTPUT);

}

void loop()

{

// Gas senor with buzzer

value\_sensor = analogRead(A1);

Serial.println(value\_sensor);

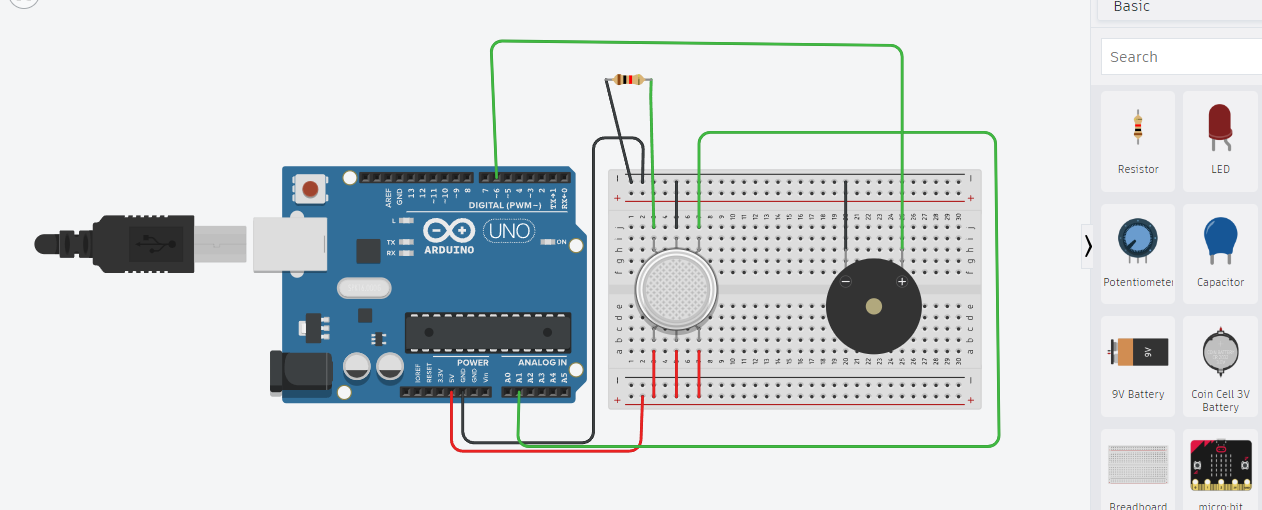
if (value\_sensor > 200) {

tone(6, 523, 1000); // play tone 60 (C5 = 523 Hz)

}

delay(10); // Delay a little bit to improve simulation performance

OutPut:



Conclusion:

This code snippet simulates a temperature sensor and checks the temperature every 5 minutes. If the temperature goes above a certain threshold (25°C in this case), it prints a message indicating that the temperature is too high. In a real-world implementation, you would replace the temperature simulation code with actual sensor readings and replace the print statements with actions or alerts that are relevant to your parking system's needs.

Team Members:

* Vinisha M
* Punitha S
* Swathi SJ
* Ruthranayaki D