**SMART PUBLIC RESTROOM USING IOT**

**Abstract:**

* To Deploy loT sensors (e.g., occupancy sensors, cleanliness sensors) in public restrooms to collect data.
* To Develop a Python script on the lot sensors to send real-time occupancy and cleanliness .

**Sensor for Deployment:**

* Deploying some sensors like cleanliness and occupancy sensors in a smart public restroom using IoT (Internet of Things) can greatly improve user experience and operational efficiency.

**Sensor Selection:**

* Choose appropriate sensors for cleanliness and occupancy detection. For cleanliness, you might use infrared sensors to detect the presence of waste or litter. For occupancy, consider ultrasonic or infrared motion sensors

**Sensors used :**

* **Turbidity Sensor:** A turbidity sensor is a sensor which is mainly used to measure scattered light suspended by solids in water. As the number of total suspended solids in the water source is increased, the turbidity level of water is increased accordingly. Turbidity sensors can be used in the determination of water quality of small as well as large water bodies.
* **MQ3 gas Sensor:** MQ3 gas sensor is an alcohol gas sensor which can detect the presence of gases which contain alcohol traces in them. It is made out of tin in the form of stannic oxide. It can detect alcohol, ethanol and smoke.
* **Node MCU wifi module:** NodeMCU is a development kit that aids in making an IoT project. This module runs on ESP8266 Wi-Fi system on a chip. It is a microcontroller unit which includes a built in WiFi module.
* **Passive Infrared (PIR) Sensors:** PIR sensors detect changes in heat signatures. When a person enters the restroom, their body heat triggers the sensor, signaling occupancy. PIR sensors are often used for controlling lighting and ventilation.
* **IOT Hardware:** Install IoT hardware like microcontrollers (e.g., Arduino or Raspberry Pi), Wi-Fi/Bluetooth modules, and power supplies. These will be the backbone of your sensor network.

**Sensor Placement:**

* Strategically place cleanliness sensors near trash cans, sinks, or toilet areas. For occupancy, install sensors near the restroom entrance or inside each stall.

**Data Connectivity:**

* Ensure your IoT devices can connect to the internet. Wi-Fi is commonly used, but you can also consider cellular or LoRaWAN for remote locations.

**Data Collection:**

* Set up a central system to collect data from the sensors. You can use cloud platforms like AWS IoT, Google Cloud IoT, or Azure IoT Hub for this purpose.

**Data Analysis:**

* Implement algorithms to process the sensor data. For cleanliness, analyze when trash cans need emptying or when restocking is required. For occupancy, track the number of available stalls and waiting times.

**User Interface:**

* Develop a user-friendly interface for restroom users. This can be a mobile app or a display outside the restroom showing cleanliness status and stall availability.

**Feedback Loop:**

* Use the data collected over time to make improvements in cleaning schedules and restroom management.

**Python script to send Data from Passive Infrared (PIR) Sensors to the restroom information platform**

**Source code:**

Install Required Libraries : pip install requests

import RPi.GPIO as GPIO

import requests

import time

# Define the PIR sensor pin and API endpoint

PIR\_SENSOR\_PIN = 17 # Replace with actual GPIO pin

API\_ENDPOINT = 'https://your-restroom-platform-api.com/submit\_occupancy\_data' # Replace with the actual API endpoint

# Set up GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(PIR\_SENSOR\_PIN, GPIO.IN)

# Main loop to continuously collect and send data

try:

while True:

# Read data from the PIR sensor (0 for no motion, 1 for motion)

occupancy\_status = GPIO.input(PIR\_SENSOR\_PIN)

# Prepare data to send to the API

data = {"occupancy": occupancy\_status}

# Send data to the restroom platform API

response = requests.post(API\_ENDPOINT, json=data)

if response.status\_code == 200:

print("Data sent successfully to the platform.")

else:

print("Failed to send data to the platform. Status code:", response.status\_code)

# Wait for some time before the next data submission

time.sleep(10) # Adjust as needed

except KeyboardInterrupt:

print("Script terminated by the user.")

finally:

GPIO.cleanup()

This script sets up the GPIO pin for the PIR sensor, reads its data, and sends it to the specified API endpoint as a JSON payload. Replace the placeholder values with the actual GPIO pin and API endpoint.

**Python script to send Data from Cleanliness Sensors(Turbidity sensor) to the restroom information platform:**

**Source code:**

Install Required Libraries : pip install requests

import requests

import time

# Define the API endpoint and sensor data acquisition method

API\_ENDPOINT = 'https://your-restroom-platform-api.com/submit\_cleanliness\_data' # Replace with the actual API endpoint

CLEANLINESS\_SENSOR\_PIN = 0 # Replace with the actual sensor pin

# Main loop to continuously collect and send data

try:

while True:

# Read data from the turbidity sensor

cleanliness\_data = read\_turbidity\_sensor(CLEANLINESS\_SENSOR\_PIN)

# Prepare data to send to the API

data = {"cleanliness": cleanliness\_data}

# Send data to the restroom platform API

response = requests.post(API\_ENDPOINT, json=data)

if response.status\_code == 200:

print("Data sent successfully to the platform.")

else:

print("Failed to send data to the platform. Status code:", response.status\_code)

# Wait for some time before the next data submission

time.sleep(300) # Adjust as needed

except KeyboardInterrupt:

print("Script terminated by the user.")

The script sends the collected data as a JSON payload to the specified API endpoint.

**Conclusion:**

In this we have discussed that how IOT sensors are measure Cleanliness and Occupancy data and collect the data and implemented python program to send this data to Information Platform