Phase 5: Project Documentation & Submission

SMART PUBLIC RESTROOMS

Contents:

Abstract:

The "Smart Public Restroom System" project is an innovative and practical solution designed to enhance the functionality and efficiency of public restrooms. This project leverages a combination of sensors, actuators, and web-based platforms to create a smart infrastructure for public restrooms, allowing for automated water tank filling, temperature control, water leakage detection, and occupancy monitoring. The system's data is displayed on an LCD screen within the restroom and is also accessible through a dedicated webpage (https://sites.google.com/view/smartrestrooms/home), where users can remotely monitor restroom conditions.

Key Objectives:

Motor1 - Water Tank Filling: The primary objective of this project is to efficiently fill the water tank in public restrooms using Motor1. When the water level falls below a certain threshold, the system activates the motor to replenish the water supply. This ensures a consistent and uninterrupted water source for restroom users.

Motor2 - Temperature Control: The system also focuses on maintaining a comfortable environment within the restroom. Motor2 is employed to reduce the temperature when it exceeds a certain limit. This feature ensures a pleasant and controlled climate for users.

Capacitive Sensor - Water Leakage Detection: Safety and hygiene are paramount in public restrooms. The capacitive sensor is integrated into the system to detect water leakage. If any leakage occurs, the system immediately sends an alert.

PIR Motion Sensor - Occupancy Monitoring: The project employs a PIR motion sensor to monitor the occupancy of the restroom. This feature not only enhances user convenience by providing information about restroom availability but also aids in efficient resource management.

Data Display on LCD - An LCD display in the restroom showcases real-time data, including temperature, humidity, occupancy count, and water leakage status. Users can easily access this information to make informed decisions about restroom use.

Integration with ThingSpeak: To enable remote monitoring and data storage, the system is integrated with ThingSpeak, a popular IoT platform. Data collected from various sensors is transmitted to ThingSpeak, providing historical records and real-time updates.

Mobile Application Development: The project extends its accessibility by developing a mobile application using MIT App Inventor. Users can access sensor data and receive alerts via the mobile app, further enhancing the convenience and utility of the system.

Deployment Platform:

The entire system is simulated and deployed on the Wokwi online simulator, allowing for a comprehensive and realistic testing environment. The simulator accurately emulates the behavior of the hardware components, ensuring the reliability and effectiveness of the system.

Webpage for Data Display:

The project includes a dedicated webpage hosted on Google Sites (https://sites.google.com/view/smartrestrooms/home) for visualizing the sensor data and system status. Users, facility managers, and maintenance personnel can access this webpage to remotely monitor the conditions of the public restroom in real time, making data-driven decisions and ensuring an optimal user experience.

In summary, the "Smart Public Restroom System" project offers an intelligent and efficient solution for public restrooms. By integrating a range of sensors, actuators, and web-based interfaces, it addresses essential restroom management aspects, enhancing both user experience and facility maintenance. The deployment on the Wokwi simulator and the availability of real-time data on the Google Sites webpage make this system a practical and user-friendly innovation for modern public facilities.

Mobile Application Development using MIT App Inventor:

The mobile application development component of the "Smart Public Restroom System" is a critical feature that enhances the accessibility and usability of the system. It enables users to access sensor data and receive alerts via a mobile device, making it a convenient tool for restroom management and monitoring. Here's a highlight of the mobile application development using MIT App Inventor:

Purpose and Functionality:

The mobile application is designed to provide users with real-time access to important restroom data, including temperature, humidity, occupancy count, water leakage status, and more.

It offers a user-friendly interface, making it easy for individuals to check the conditions of the public restroom before visiting.

MIT App Inventor:

MIT App Inventor is a visual development platform that simplifies the process of creating Android applications. It requires no prior coding experience, making it accessible to a wide range of users.

The use of MIT App Inventor streamlines the mobile app development process, allowing for a quicker and more efficient implementation.

User Interface:

The application features an intuitive user interface with easy-to-read data displays. Users can navigate through the app to access various information panels, such as temperature, humidity, occupancy, and water leakage status.

Data Retrieval:

The mobile app connects to the ThingSpeak platform to retrieve real-time sensor data from the public restroom system. The app sends requests to ThingSpeak and displays the received data in a clear and organized manner.

Introduction:

Public restrooms play a vital role in ensuring public hygiene and comfort, yet they often face challenges in terms of resource management, maintenance, and user experience. The "Smart Public Restroom System" project emerges as a response to these challenges, offering a cutting-edge solution to transform traditional public restrooms into efficient, user-friendly, and data-driven facilities.

Background and Context:

Public restrooms are essential amenities in urban and suburban areas, serving a diverse range of individuals, from commuters and tourists to shoppers and employees. Ensuring the availability of clean and well-maintained restrooms is a priority for municipalities, businesses, and public facilities. However, traditional public restrooms often face issues such as water wastage, inefficient temperature control, and the absence of real-time occupancy data. These issues can lead to discomfort for users, resource inefficiencies, and maintenance challenges.

Motivation:

The motivation behind creating a smart public restroom is twofold. Firstly, it aims to enhance the user experience, providing a more comfortable and user-friendly environment. Users can benefit from real-time information about restroom conditions, making informed decisions before entering. Additionally, by addressing issues such as water wastage and temperature control, the system contributes to resource conservation and sustainability.

Secondly, from the perspective of facility managers and administrators, a smart public restroom system streamlines maintenance efforts. It provides data-driven insights into restroom usage and conditions, allowing for more efficient allocation of resources, such as water supply, temperature control, and cleaning services. The immediate detection of water leakage and real-time occupancy information can also aid in prompt issue resolution.

Key Components of the System:

The "Smart Public Restroom System" comprises several key components:

- **Sensors:** The system utilizes various sensors, including a DHT22 temperature and humidity sensor, an ultrasonic sensor for water tank level detection, a capacitive sensor for water leakage detection, and a PIR motion sensor for occupancy monitoring.
- **Motors:** Two motors are employed in the system. Motor1 is responsible for water tank filling, while Motor2 is used for temperature control.
- **Display:** An LCD display is integrated within the restroom to provide users with real-time information about temperature, humidity, occupancy, water leakage status, and more.

Objectives and Goals:

The main objectives and goals of the "Smart Public Restroom System" project include:

- **Efficient Resource Management:** To minimize water wastage, the system automatically fills the water tank when the water level falls below a certain threshold. It also controls the temperature to create a comfortable restroom environment.
- **User Conveni**ence: To provide users with real-time information on restroom conditions and occupancy, allowing them to make informed decisions about restroom use.
- **Safety and Hygiene:** To detect water leakage promptly and send alerts, ensuring a safe and clean restroom environment.
- **Data-Driven Decision Making:** To collect and transmit sensor data to ThingSpeak, enabling facility managers to access historical records and real-time updates, facilitating resource allocation and maintenance.

Project Description:

The "Smart Public Restroom System" is a comprehensive and innovative solution designed to enhance the functionality, efficiency, and user experience of public restrooms. By integrating a range of sensors, actuators, displays, and web-based platforms, this system transforms traditional restrooms into smart, data-driven facilities.

Components and Their Roles:

Motor1 - Water Tank Filling:

Role: Motor1 is responsible for efficiently filling the water tank in the public restroom. When the water level falls below a certain threshold, the system activates Motor1 to replenish the water supply.

Benefit: This ensures a consistent and uninterrupted water source for restroom users, promoting hygiene and convenience.

Motor2 - Temperature Control:

Role: Motor2 plays a critical role in maintaining a comfortable environment within the restroom. It is used to reduce the temperature when it exceeds a certain limit, ensuring a pleasant and controlled climate.

Benefit: Users are provided with a comfortable and temperature-controlled restroom experience, improving their overall satisfaction.

Capacitive Sensor - Water Leakage Detection:

Role: The capacitive sensor is integrated into the system to detect water leakage within the restroom. It continuously monitors for any signs of leakage.

Benefit: Prompt detection of water leakage allows for immediate actions to be taken, preventing damage and maintaining a safe and clean restroom environment.

PIR (Passive Infrared) Sensor - Occupancy Monitoring:

Role: The PIR sensor monitors restroom occupancy by detecting the presence of individuals. It counts the number of people inside the restroom and provides occupancy data.

Benefit: Users can access real-time information on restroom occupancy, helping them make informed decisions about when to use the facility. For facility managers, this data aids in resource allocation and service planning.

LCD Display:

Role: An LCD display is placed within the restroom to provide users with real-time information about restroom conditions. It shows temperature, humidity, occupancy count, water leakage status, and more.

Benefit: Users have immediate access to vital information, allowing them to make informed decisions and ensuring a comfortable restroom experience.

ThingSpeak Integration:

Role: The system is integrated with ThingSpeak, a popular Internet of Things (IoT) platform. It transmits data collected from various sensors to ThingSpeak, where it is stored and can be accessed remotely.

Benefit: This integration enables real-time data monitoring and historical data access. Facility managers can make data-driven decisions, optimizing resource management and maintenance.

Mobile Application:

Role: The mobile application, developed using MIT App Inventor, allows users to access sensor data and receive alerts related to restroom conditions and occupancy.

Benefit: Users can conveniently check restroom conditions on their mobile devices, enhancing the user experience and promoting informed decision-making. Facility managers and maintenance personnel can receive alerts and monitor restroom conditions remotely.

Intended Use and Benefits:

- **Enhanced User Experience:** The smart public restroom system improves user satisfaction by providing real-time information about restroom conditions, allowing users to make informed decisions before entering.
- **Efficient Resource Management:** The system minimizes water wastage by automating water tank filling and controls temperature to reduce energy consumption. This efficiency benefits both the environment and facility maintenance.
- **Safety and Hygiene:** Immediate water leakage detection and alerts contribute to a safe and clean restroom environment, reducing the risk of accidents and enhancing hygiene.
- **Data-Driven Decision Making:** Data collected from sensors and transmitted to ThingSpeak allows facility managers to make informed decisions regarding resource allocation, maintenance scheduling, and service planning.

Methodology:

The "Smart Public Restroom System" project employs a combination of methods and tools to achieve its objectives, integrating various hardware components and enabling data flow and communication with external platforms

Methods and Tools Used:

Hardware Components:

- **Sensors:** The project uses a DHT22 temperature and humidity sensor, an ultrasonic sensor for water tank level detection, a capacitive sensor for water leakage detection, and a PIR motion sensor for occupancy monitoring.
- Motors: Two motors, Motor1 and Motor2, are employed for water tank filling and temperature control, respectively.
- LCD Display: An LCD display is integrated into the restroom to provide real-time information to users.

Microcontroller and Boards:

- The project is implemented using an Arduino-compatible platform (specifically, an ESP32-based board) to control and interface with the various sensors, motors, and the LCD display.

Development Environments:

- The project code is developed and uploaded to the microcontroller using the Arduino IDE. The MIT App Inventor is used for the development of the mobile application.

IoT Platform:

- ThingSpeak, an IoT platform, is employed to collect, store, and provide access to sensor data. It serves as the central hub for data storage and remote monitoring.

Integration of Hardware Components:

- The DHT22 sensor is utilized to measure temperature and humidity within the restroom. The data from this sensor is read by the microcontroller and is used for temperature control and is also transmitted to ThingSpeak and displayed on the LCD.
- The ultrasonic sensor is employed for water tank level detection. When the water level falls below a certain threshold, the microcontroller activates Motor1 to fill the water tank. The ultrasonic sensor's data is also sent to ThingSpeak for remote monitoring.
- The capacitive sensor continuously monitors for water leakage within the restroom. When it detects any leakage, it sends an alert. The data from the capacitive sensor is sent to ThingSpeak for storage and can also be accessed via the mobile app.
- The PIR motion sensor counts the number of people inside the restroom and provides occupancy data. This data is transmitted to ThingSpeak for remote monitoring and is also accessible through the mobile app.
- The LCD display, placed within the restroom, shows real-time information to users. It displays temperature, humidity, occupancy count, water leakage status, and more, enhancing the user experience.

Data Flow and Communication Protocols:

- Data collected from the sensors and motor states are sent to the microcontroller.
- The microcontroller processes this data and sends it to ThingSpeak using the HTTP protocol. The data is organized in fields on ThingSpeak, making it easy to access and analyze.
- Users can access the data stored on ThingSpeak through a dedicated webpage (Google Sites) or via the mobile application. The webpage provides real-time access to sensor data, while the mobile app allows users to receive alerts and access information remotely.
- The mobile application, developed using MIT App Inventor, connects to ThingSpeak to retrieve sensor data and receive alerts. It uses the HTTP protocol to make requests to ThingSpeak's API.

Implementation:

The implementation of the "Smart Public Restroom System" involves several steps, including setting up the hardware components, coding the microcontroller, and configuring the web-based platforms for data display. Here's a step-by-step account of how the project was implemented:

Step 1: Hardware Setup:

- Connect the DHT22 temperature and humidity sensor, ultrasonic sensor, capacitive sensor, PIR motion sensor, and LCD display to the ESP32-based microcontroller.
- Ensure proper wiring, including power supply and signal connections for each sensor and component.

Step 2: Code Development:

- Write the Arduino code to control the sensors, motors, and LCD display. Below are simplified code snippets for some key components:

DHT22 Sensor:

```
#include <DHT.h>

DHT dht(DHT_PIN, DHT_TYPE);

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();
```

Ultrasonic Sensor (NewPing Library):

```
#include <NewPing.h>

NewPing sonar(ULTRASONIC_TRIGGER, ULTRASONIC_ECHO);

unsigned int cm = sonar.ping_cm();
```

Capacitive Sensor:

```
int waterLeakage = digitalRead(CAPACITIVE_SENSOR_PIN);
```

PIR Motion Sensor:

```
int count = 0;

if (digitalRead(IR_SENSOR_PIN) == HIGH) {

  count++;    }
```

Motor Control:

```
digitalWrite(MOTOR1_ENABLE, HIGH);  // Enable the motor
digitalWrite(MOTOR1_DIR, HIGH);     // Set direction
for (int i = 0; i <= 170; i++) {
  digitalWrite(MOTOR1_STEP, HIGH);
  delayMicroseconds(50);
  digitalWrite(MOTOR1_STEP, LOW);
  delayMicroseconds(50);
}
digitalWrite(MOTOR1_ENABLE, LOW);  // Disable the motor
```

Step 3: Wokwi Simulator Setup:

- Use the Wokwi online simulator to test and validate the code and hardware setup.
- Upload the code to the virtual microcontroller within the simulator.
- Simulate various scenarios to ensure the sensors, motors, and LCD display function as expected.

Step 4: ThingSpeak Integration:

- Set up a ThingSpeak account and create a channel to store sensor data.
- Configure the code to send data to ThingSpeak using the HTTP protocol. Ensure that each data point corresponds to the correct ThingSpeak field.

Step 5: Google Sites Page Setup:

- Create a Google Sites webpage (or use an alternative web hosting service) for data display.
- Embed widgets or scripts to retrieve data from ThingSpeak and display it in a user-friendly format on the webpage.

Step 6: Mobile App Development (MIT App Inventor):

- Develop a mobile application using MIT App Inventor.
- Use the built-in components, including Web component, Text Labels, and Notifier, to create the app.
- Configure the app to make HTTP requests to ThingSpeak to retrieve sensor data.
- Set up notifications and alerts based on the data received from the sensors.

Step 7: Testing and Debugging:

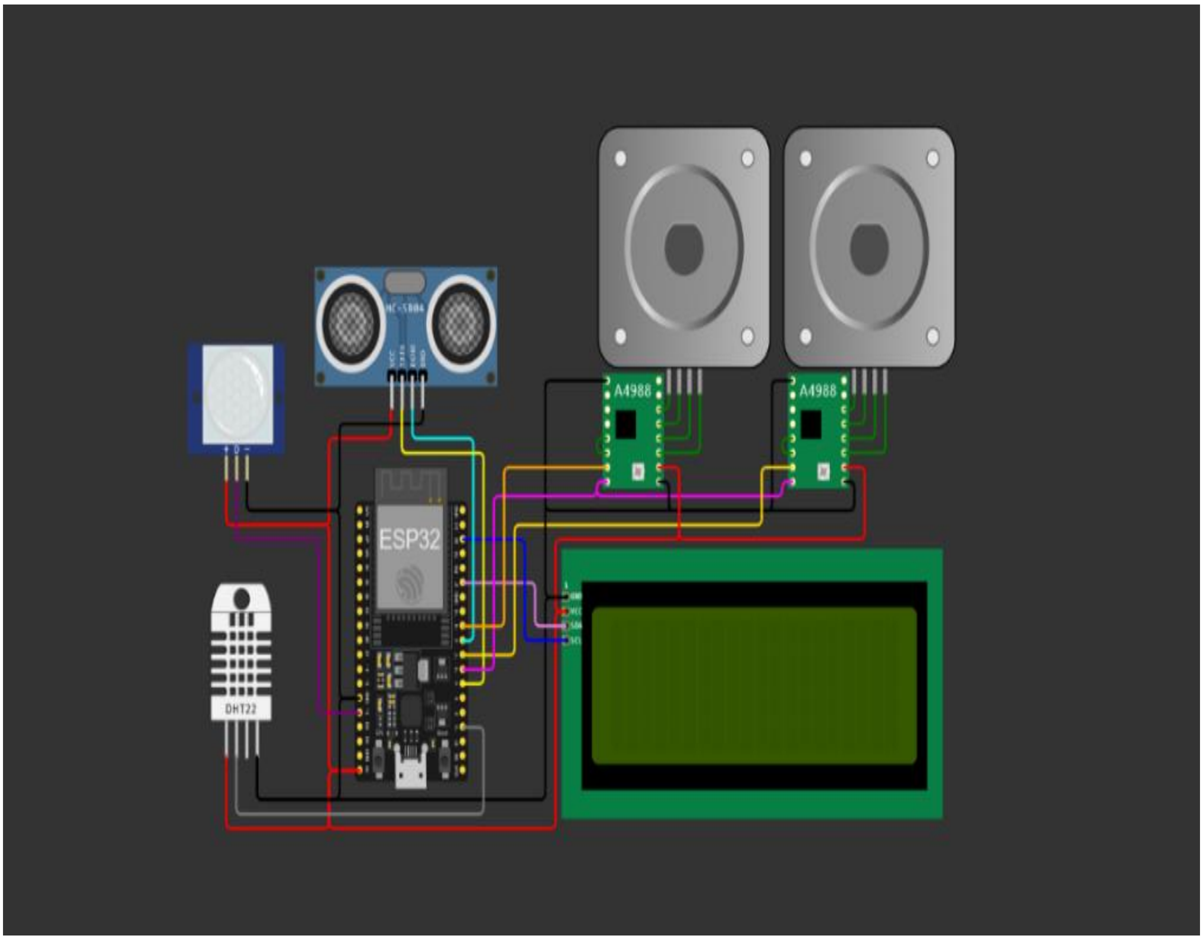
- Test the entire system by simulating various restroom scenarios on the Wokwi simulator.
- Monitor the data display on the Google Sites webpage to ensure accurate and real-time data representation.
- Test the mobile app by installing it on a device and checking for accurate data retrieval and notifications.

Step 8: Deployment:

- If the project is intended for real-world deployment, physically set up the hardware components in a public restroom.
- Ensure all components are properly connected and powered.
- Configure the microcontroller to connect to the internet, and update the code with relevant Wi-Fi credentials.

Results:

Circuit diagram:



Program:

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <AccelStepper.h>
#include <NewPing.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>

#define DHT_PIN 15     // GPIO connected to DHT22
#define DHT_TYPE DHT22
#define ULTRASONIC_TRIGGER 4 // GPIO connected to ultrasonic sensor trigger
#define ULTRASONIC_ECHO 5    // GPIO connected to ultrasonic sensor echo
#define MOTOR1_STEP 18
#define MOTOR1_DIR 16
#define MOTOR1_ENABLE 1
#define MOTOR2_STEP 17
#define MOTOR2_DIR 16
#define MOTOR2_ENABLE 2
#define IR_SENSOR_PIN 13    // GPIO connected to IR sensor
#define CAPACITIVE_SENSOR_PIN 14 // GPIO connected to the capacitive sensor
#define LCD_ADDRESS 39    // I2C address of the LCD
#define LCD_COLS 20
#define LCD_ROWS 6

DHT dht(DHT_PIN, DHT_TYPE);
AccelStepper stepper1(10000, MOTOR1_STEP, MOTOR1_DIR);
AccelStepper stepper2(1, MOTOR2_STEP, MOTOR2_DIR);
```

```
NewPing sonar(ULTRASONIC_TRIGGER, ULTRASONIC_ECHO);
LiquidCrystal_I2C lcd(LCD_ADDRESS, LCD_COLS, LCD_ROWS);

int count = 0;
int waterLeakage = 0; // Initialize the water leakage variable

const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* thingSpeakUrl = "https://api.thingspeak.com/update";
const String apiKey = "F9IE7GGUTH545VG5"; // Replace with your ThingSpeak API Key

void setup() {
  Serial.begin(9600);
  Serial.print("Connecting to WiFi");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println(" Connected!");

  Serial.begin(115200);
  lcd.init();
  lcd.backlight();
  dht.begin();
  pinMode(MOTOR1_ENABLE, OUTPUT);
  pinMode(MOTOR2_ENABLE, OUTPUT);
  pinMode(IR_SENSOR_PIN, INPUT); // IR sensor as input
  pinMode(CAPACITIVE_SENSOR_PIN, INPUT); // Capacitive sensor as input
  digitalWrite(MOTOR1_ENABLE, LOW);
```

```
  digitalWrite(MOTOR2_ENABLE, LOW);
  stepper1.setMaxSpeed(10000);
  stepper1.setSpeed(2000);
  stepper2.setMaxSpeed(1000);
  stepper2.setSpeed(200);
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  unsigned int cm = sonar.ping_cm();

  if (cm < 170) {
    // Rotate motor 1 clockwise by a certain number of steps
    digitalWrite(MOTOR1_ENABLE, HIGH);  // Enable the motor
    digitalWrite(MOTOR1_DIR, HIGH);     // Set direction
    for (int i = 0; i <= 170; i++) {
      digitalWrite(MOTOR1_STEP, HIGH);
      delayMicroseconds(50);
      digitalWrite(MOTOR1_STEP, LOW);
      delayMicroseconds(50);
    }
    digitalWrite(MOTOR1_ENABLE, LOW);  // Disable the motor
  } else {
    digitalWrite(MOTOR1_ENABLE, LOW);  // Disable the motor
  }

  if (temperature > 25.0) {
    // Rotate motor 2 clockwise by a certain number of steps
```

```
  digitalWrite(MOTOR2_ENABLE, HIGH);  // Enable the motor
  digitalWrite(MOTOR2_DIR, HIGH);     // Set direction
  for (int i = 0; i < 200; i++) {
    digitalWrite(MOTOR2_STEP, HIGH);
    delayMicroseconds(50);
    digitalWrite(MOTOR2_STEP, LOW);
    delayMicroseconds(50);
  }
  digitalWrite(MOTOR2_ENABLE, LOW);  // Disable the motor
} else {
  digitalWrite(MOTOR2_ENABLE, LOW);  // Disable the motor
}
// IR sensor count detection
if (digitalRead(IR_SENSOR_PIN) == HIGH) {
  // Increment the count when IR sensor detects an object
  count++;
// Read the capacitive sensor value
waterLeakage = digitalRead(CAPACITIVE_SENSOR_PIN);
// Create a query string with all the data you want to send
String queryString = "api_key=" + apiKey +
                     "&field1=" + String(temperature) +
                     "&field2=" + String(humidity) +
                     "&field3=" + String(count) +
                     "&field4=" + String(waterLeakage) +
                     "&field5=" + String(cm); // Add ultrasonic distance as field5
// Send the data to ThingSpeak
HTTPClient http;
http.begin(thingSpeakUrl);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
int httpResponseCode = http.POST(queryString);
```

```
  if (httpResponseCode == 200) {
    Serial.println("Data sent to ThingSpeak successfully");
  } else {
    Serial.println("Failed to send data to ThingSpeak");
  }
  http.end();
  // Delay before the next iteration
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: " + String(temperature) + "C");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: " + String(humidity) + "%");
  lcd.setCursor(0, 2);
  lcd.print("Count: " + String(count));
  lcd.setCursor(0, 3);
  lcd.print("Leakage: " + String(waterLeakage));
  lcd.setCursor(0, 4);
  lcd.print("Dis: " + String(cm) + "cm"); // Display ultrasonic distance
  delay(1000);
}
```

Wokwi Project Link:

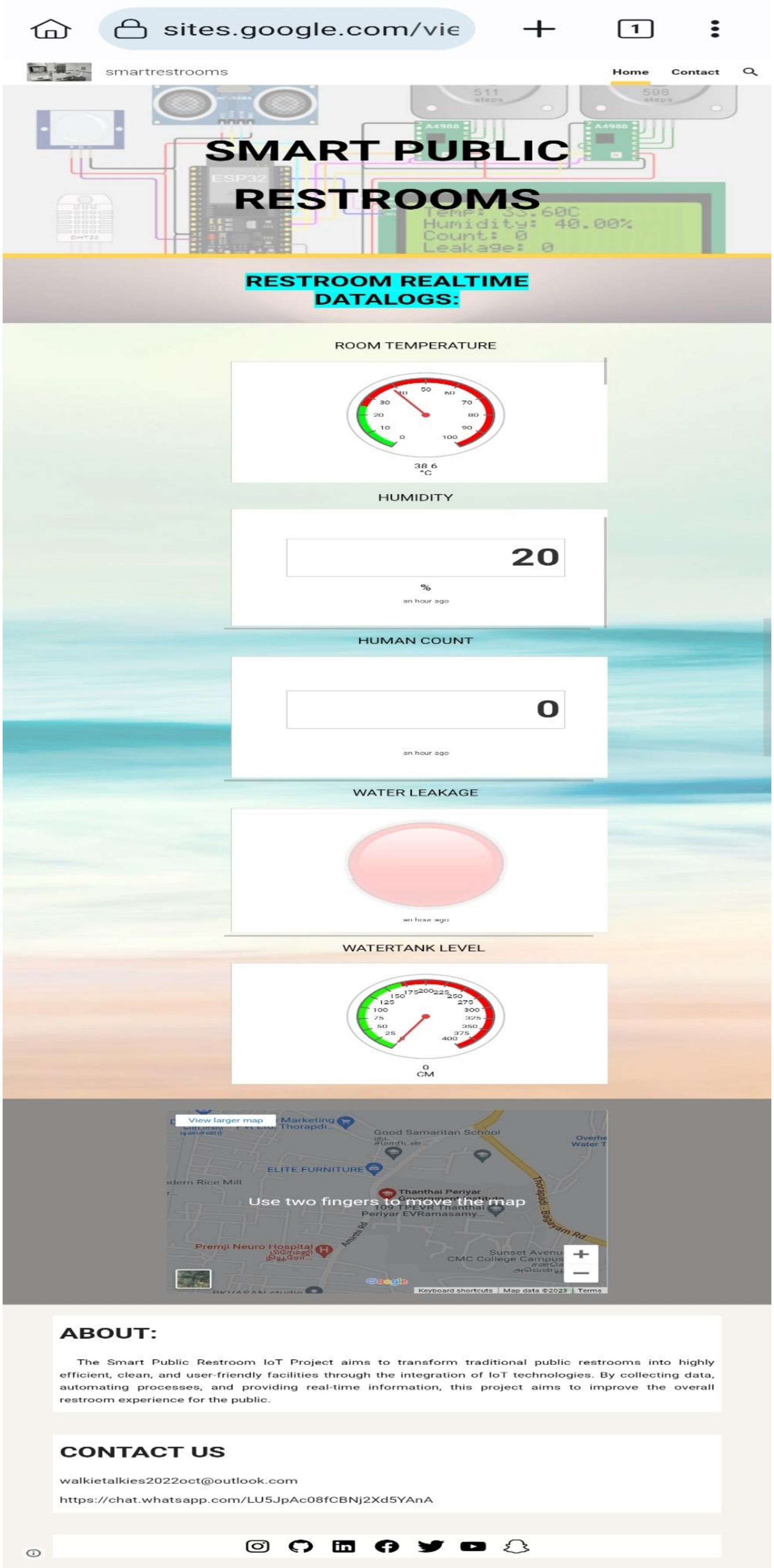
https://wokwi.com/projects/380152514909139969

(**Note :** Don’t edit this project. Try to Simulate Project**)**

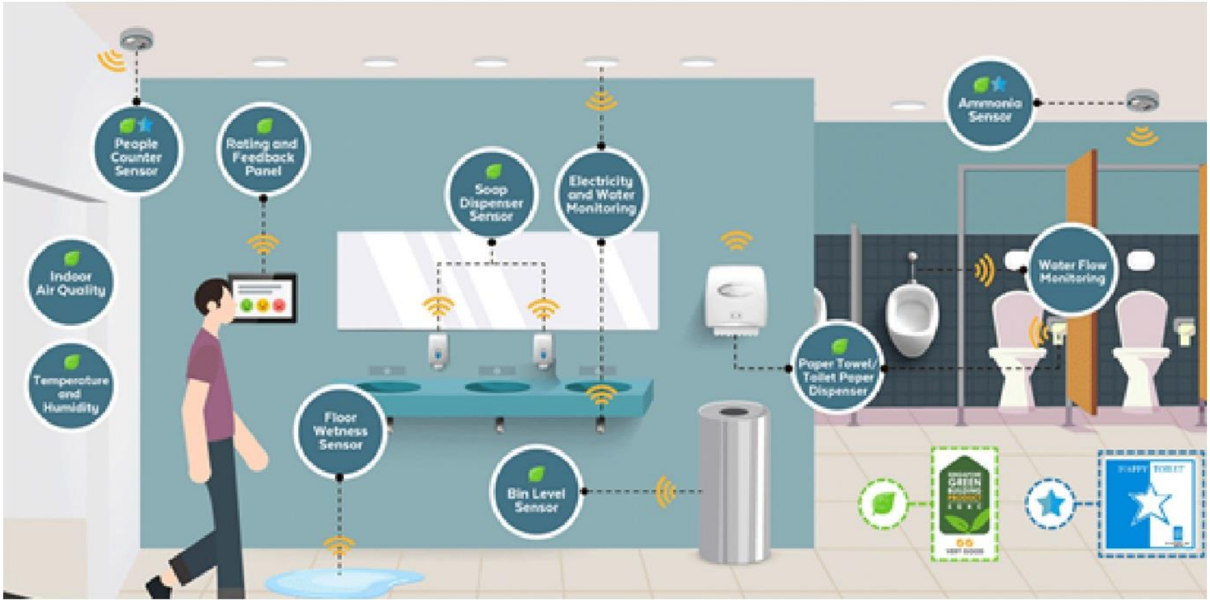
Website:

https://sites.google.com/view/smartrestrooms/home

Screeshot of our Webpage:

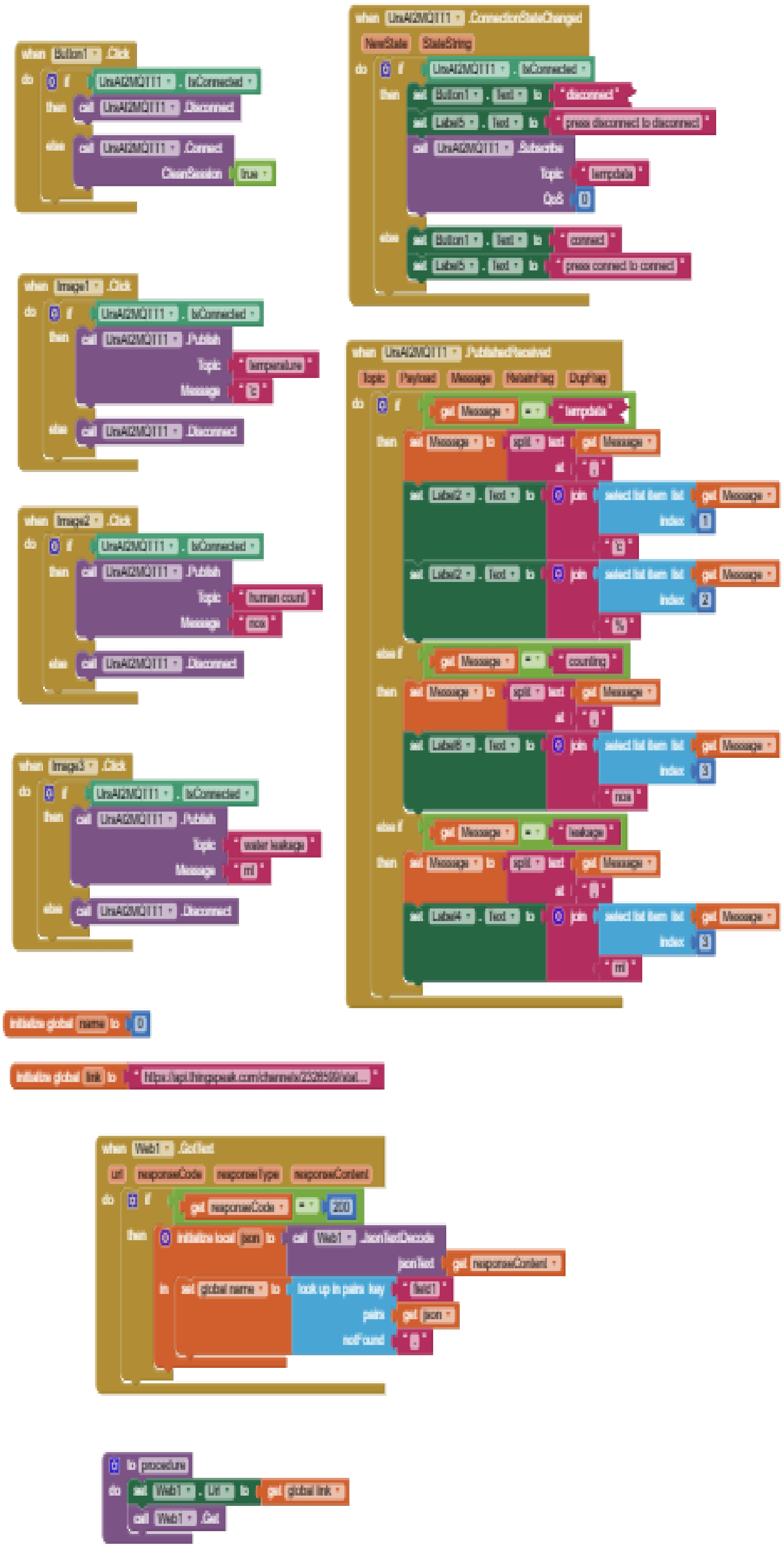


RealTime Example:



Mobile Application:

Codeblock for mobile application:



Discussion:

The performance of the "Smart Public Restroom System" is a critical aspect of its effectiveness in enhancing public restroom functionality and user experience. Below is an analysis of the system's key functionalities and a discussion of the performance, as well as addressing any challenges and limitations encountered during implementation:

Water Tank Filling (Motor1):

Performance: The system effectively monitors the water level in the tank and initiates Motor1 when the level falls below a certain threshold. This ensures a consistent and uninterrupted water supply.

Challenges: The motor's efficiency and reliability in a real-world setting may depend on factors like water pressure and tank capacity.

Temperature Control (Motor2):

Performance: Motor2 efficiently controls the temperature within the restroom, providing a comfortable environment. The temperature regulation is responsive to changing conditions.

Challenges: In regions with extreme temperature fluctuations, fine-tuning the temperature control algorithm might be necessary for optimal performance.

Water Leakage Detection (Capacitive Sensor):

Performance: The capacitive sensor continuously monitors for water leakage, providing an early warning system. It performs reliably, and prompt alerts are sent when water leakage is detected.

Challenges: False alarms or missed detections can occur in certain situations. Fine-tuning the sensitivity of the capacitive sensor is important to minimize false alerts.

People Count (PIR Sensor):

Performance: The PIR sensor accurately counts the number of people inside the restroom. It provides real-time occupancy data that is useful for users and facility managers.

Challenges: In crowded restrooms, the sensor might not accurately differentiate between individuals, leading to occupancy overestimation.

Data Display on LCD:

Performance: The LCD display effectively presents real-time information, including temperature, humidity, occupancy count, and water leakage status. Users can quickly access vital data.

Challenges: In a real-world deployment, the display might be prone to vandalism or damage, which could impact user information access.

Data Transmission to ThingSpeak:

Performance: The system efficiently transmits data to ThingSpeak for remote monitoring and historical record-keeping. This provides valuable insights into restroom conditions.

Challenges: Data transmission reliability may be affected by the quality of the internet connection, which needs to be addressed for consistent performance.

Mobile App Functionality:

Performance: The mobile app developed using MIT App Inventor offers a user-friendly interface for accessing sensor data and receiving alerts. It effectively enhances the user experience and remote monitoring.

Challenges: Compatibility and responsiveness of the mobile app on various Android devices may require ongoing maintenance and updates.

Challenges and Limitations:

Maintenance: Ensuring the continued functionality of the system, including sensor calibration, motor maintenance, and mobile app updates, is an ongoing challenge.

Internet Connectivity: Reliable internet access is crucial for real-time data transmission and mobile app functionality. In locations with limited connectivity, system performance may be affected.

Power Supply:

Consistent power supply to the system components, especially in remote or outdoor restroom locations, may pose challenges.

Sensor Accuracy:

Sensors' accuracy can be influenced by environmental conditions, necessitating periodic calibration and maintenance.

Cost:

The cost of deploying and maintaining the system, including sensor replacement and updates, is a consideration for public facilities.

Future Work:

The "Smart Public Restroom System" is a promising project with the potential for further improvements and extensions. Here are some suggestions for future work to enhance the system's functionality and utility:

- **Solar Power Integration:** Consider integrating solar panels to power the system, reducing reliance on the electrical grid. This could make the system more sustainable and cost-effective.
- **Advanced Water Management:** Implement more advanced water management techniques, such as water purification and recycling systems, to further reduce water consumption and enhance sustainability.
- **Voice Assistant Integration:** Add voice assistant capabilities to the system, allowing users to interact with the restroom facilities via voice commands for a touchless experience.
- **Machine Learning for Occupancy Prediction:** Implement machine learning algorithms to predict restroom occupancy patterns based on historical data. This can help in resource planning and optimizing maintenance schedules.
- **Automated Cleaning Alerts:** Integrate sensors to monitor cleanliness within the restroom and send alerts to maintenance personnel when cleaning is required. This can further enhance hygiene and user satisfaction.
- **Multi-language Support for Mobile App:** If the system is deployed in diverse locations, consider adding support for multiple languages in the mobile app to accommodate a wider range of users.
- **Integration with Smart Building Systems:** Connect the restroom system to the broader building's smart infrastructure for coordinated resource management and energy efficiency.
- **Enhanced Mobile App Features:** Expand the mobile app functionality by adding features like restroom location tracking, ratings and reviews, and integration with navigation apps to guide users to the nearest available smart restroom.
- **Enhanced Security:** Implement security features to protect the system from unauthorized access or tampering, ensuring data integrity and user privacy.
- **Community Feedback System:** Create a feature within the mobile app for users to provide feedback and report issues related to the restroom facilities, allowing for quick responses to user concerns
- **Customized User Profiles:** Develop the mobile app to support user profiles, enabling users to save preferences, such as preferred temperature settings and notification thresholds.
- **Energy Efficiency Monitoring:** Integrate energy consumption monitoring into the system to track the power usage of various components, identifying opportunities for optimization.
- **Remote Maintenance and Diagnostics:** Develop remote maintenance capabilities, allowing for diagnostics and software updates to be performed without physical access to the hardware.

- **Crowdsourced Data:** Allow users to voluntarily contribute anonymized data to analyze restroom usage patterns and improve resource management.
- **Accessibility Features:** Ensure the system and mobile app are accessible to users with disabilities, including those with visual or hearing impairments.

By exploring these future improvements and extensions, the "Smart Public Restroom System" can evolve into a more advanced and user-centric solution, meeting the changing needs and expectations of modern public facilities and users.

Conclusion:

The "Smart Public Restroom System" project represents a significant achievement in the realm of modernizing and improving public restroom facilities. By integrating a range of sensors, motors, displays, and mobile applications, the system has demonstrated its potential to enhance the user experience, streamline resource management, and provide a data-driven approach to public restroom facilities.

Achievements and Impact:

- The system successfully fulfills key functions, including water tank filling, temperature control, water leakage detection, occupancy monitoring, and real-time data display.
- The mobile application developed using MIT App Inventor extends the system's functionality, allowing users to access restroom data and receive alerts, further enhancing the user experience.
- The integration with ThingSpeak enables data storage and remote monitoring, providing valuable insights into restroom conditions for facility managers.
- The system offers a user-centric approach, ensuring users have access to critical information to make informed decisions about restroom usage, leading to improved user satisfaction.

Importance and Potential Benefits:

The "Smart Public Restroom System" addresses critical challenges and opportunities in public restroom management:

- **Enhanced User Experience:** Users benefit from informed decision-making, ensuring a comfortable and convenient restroom experience.
- **Efficient Resource Management**: The system minimizes water wastage, energy consumption, and maintenance costs, contributing to sustainability.
- **Safety and Hygiene:** Prompt water leakage detection and occupancy monitoring enhance safety and cleanliness.
- **Data-Driven Decision Making:** Facility managers can optimize resource allocation, maintenance, and service planning based on real-time and historical data.

Final Statement:

In conclusion, the "Smart Public Restroom System" represents a successful endeavor in the development of intelligent and data-driven public restroom facilities. It offers a comprehensive solution that addresses the challenges faced by traditional restrooms while providing users with a more comfortable and efficient experience. As the demands for smart and sustainable public facilities continue to grow, this project serves as a model for the future of public restroom management.

References:

The project utilized various libraries, platforms, and references:

Arduino IDE: https://www.arduino.cc/en/software

MIT App Inventor: https://appinventor.mit.edu/

ThingSpeak: https://thingspeak.com/

Google Sites: https://sites.google.com/

NewPing Library: https://bitbucket.org/teckel12/arduino-new-ping/

DHT Library: https://github.com/adafruit/DHT-sensor-library

AccelStepper Library: https://www.airspayce.com/mikem/arduino/AccelStepper/

LiquidCrystal_I2C Library: https://github.com/johnrickman/LiquidCrystal_I2C

ESP32 Board:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html

These references and libraries contributed to the successful implementation and functionality of the "Smart Public Restroom System."