**IOT PHASE- 3**

**PROJECT TITLE:** SmartPublicRestroom

**PHASE 3 :** Development part-1

Presented by

* SNEHA K
* JENIFER TABITHA S
* GAYATHRI P
* JASMIN MARY J

**INRODUCTION:**

In an era characterized by technological advancements, the concept of a "Smart Public Toilet" represents a significant leap forward in enhancing public infrastructure. This project harnesses the power of Internet of Things (IOT) sensors to transform conventional public restrooms into intelligent and efficient facilities. By integrating a range of sensors, this smart toilet aims to optimize resource utilization, improve hygiene, and enhance the overall user experience.

****

**A smart public restroom can incorporate various tools and techniques to enhance functionality, hygiene, and user experience:**

**Sensor** **Technology**: Installing motion sensors for faucets, soap dispensers, and Installing hand dryers to reduce touchpoints and conserve water and energy.

**Smart** **Toilets**: High-tech toilets with features like automated flushing, heated seats, and built-in bidets.

**IOT** **Devices**: Using Internet of Things devices to monitor restroom conditions, such as occupancy, temperature, and cleaning needs.

**Real**-**time** **Feedback**: Collecting user feedback through QR codes or kiosks to improve maintenance and cleanliness.

**Accessibility** **Features**: Ensuring accessibility for people with disabilities through features like automated doors, braille signage, and accessible fixtures.

**Energy** **Efficiency**: Employing energy-efficient lighting and HVAC systems to reduce power consumption.

**Smart** **Dispensers**: Smart dispensers that notify staff when supplies like toilet paper or soap are running low.

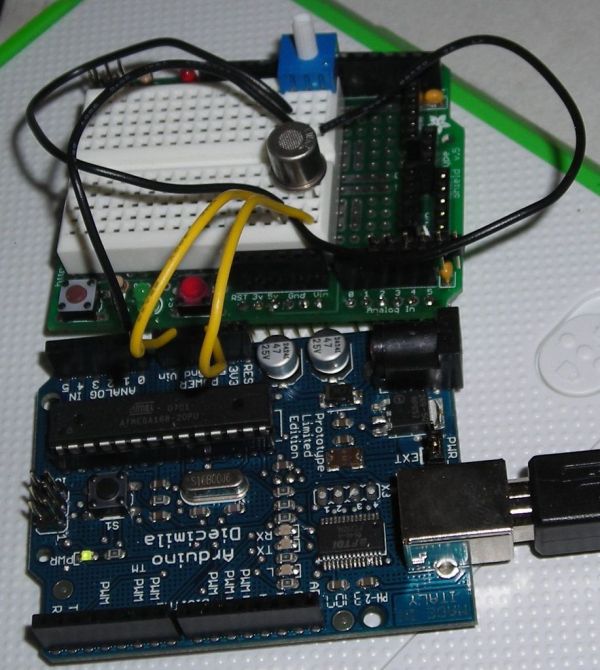
**Data** **Analytics**: Using data analytics to optimize restroom maintenance schedules and improve resource allocation.

**Environmental** **Sustainability**: Incorporating sustainable design elements, like low-flow toilets and water-saving fixtures.

**App** **Integration**: Developing smartphone apps for users to find and rate nearby public restrooms, as well as receive alerts for cleanliness or maintenance issues.

**DEPLOY IOT SENSORS(E.G. OCCUPANCY SENSOR,CLEANLINESS SENSOR)IN PUBLIC RESTROOM TO COLLEC DATA:**

**ODOR SENSORS:**

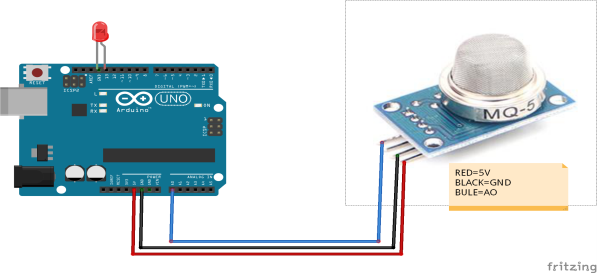
Gas Sensors (e.g., methane or ammonia detectors) : These sensors continuously monitor air quality in the toilet. If unusual odors are detected, the system can activate a ventilation system to improve air quality 

**FIG2**. ODOR SENSOR WITH ARDUINO

**WORKING PRINCIPLE:**

* A gas sensor in a smart public toilet, connected to an Arduino, operates by measuring the concentration of specific gases, like methane or hydrogen sulfide, in the air.
* It uses a gas-sensitive material that changes its electrical properties when exposed to these gases.
* The Arduino reads these changes and converts them into digital data.
* If gas levels exceed safe limits, the Arduino triggers an alarm or ventilation system, ensuring public safety.
* This real-time monitoring enhances hygiene and prevents accidents in public restrooms.

**PIN CONNECTION WITH ARDUINO:**



RED-11V

BLACK-GND,

BLUE-AD

PYTHON SCRIPT FOR ODOUR SENSOR:

import serial

import time

ser = serial.Serial('COM3', 9600) # Replace 'COM3' with the actual serial port

try:

while True:

serial\_data = ser.readline().decode('utf-8').strip()

if serial\_data:

gas\_value = int(serial\_data)

printf("Gas Sensor Value: {gas\_value}")

time.sleep(2) # Read data every 2 seconds

except KeyboardInterrupt:

ser.close()

**ALGORITHM:**

The code you provided is an Arduino sketch for reading analog data from an MQ-2 gas sensor connected to an analog pin and displaying the sensor values on the serial monitor. Here's an algorithmic description of what the code does:

Define a constant "gasSensorPin" to specify the analog pin connected to the MQ-2 gas sensor.

In the setup function:

a. Initialize serial communication with a baud rate of 9600 to communicate with the computer.

In the loop function:

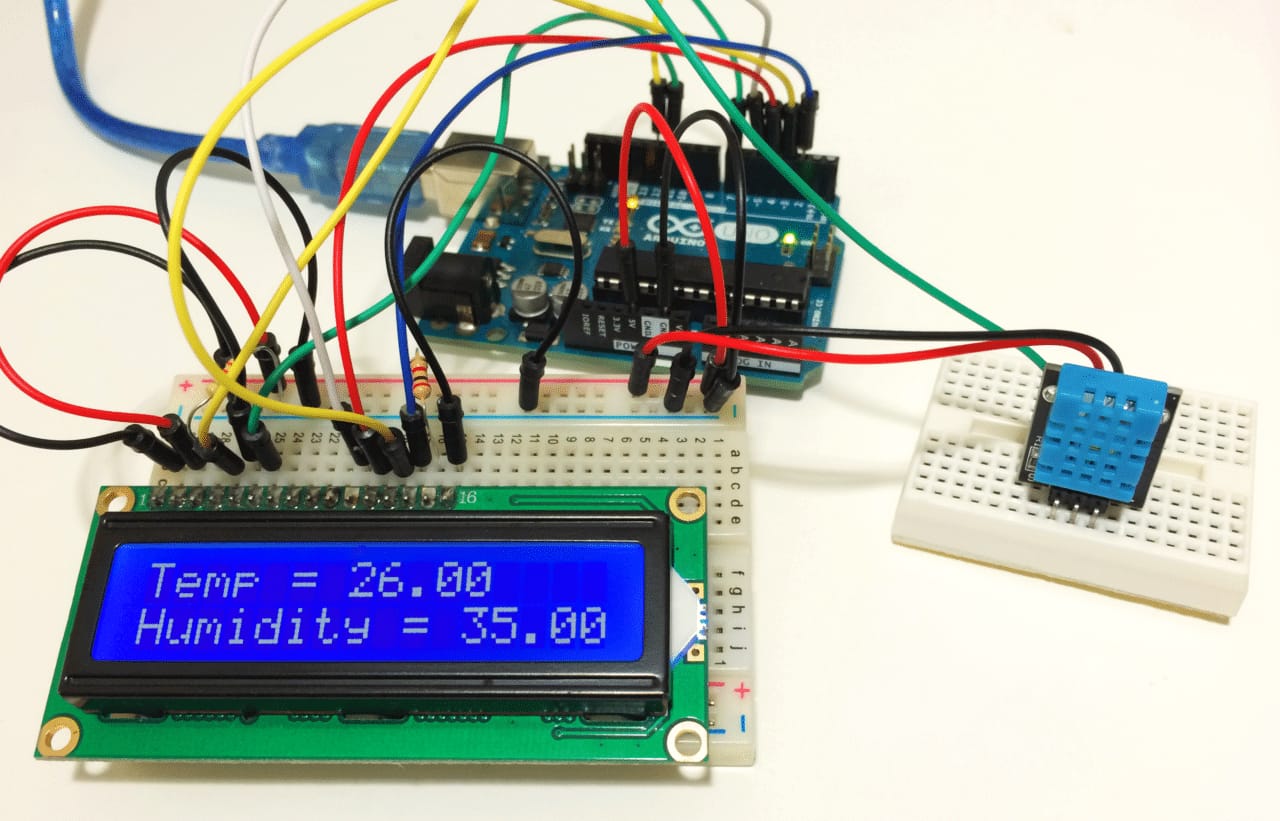
a. Read the analog value from the gas sensor by calling analogRead(gasSensorPin) and store it in the "sensorValue" variable.

b. Print the "sensorValue" to the serial monitor using Serial.println() to send the data to the computer for monitoring.

c. Add a delay of 1000 milliseconds (1 second) using delay…

**TEMPERATURE AND HUMIDITY SENSOR:**

* To monitor restroom conditions, which could be useful for maintenance or user comfort.

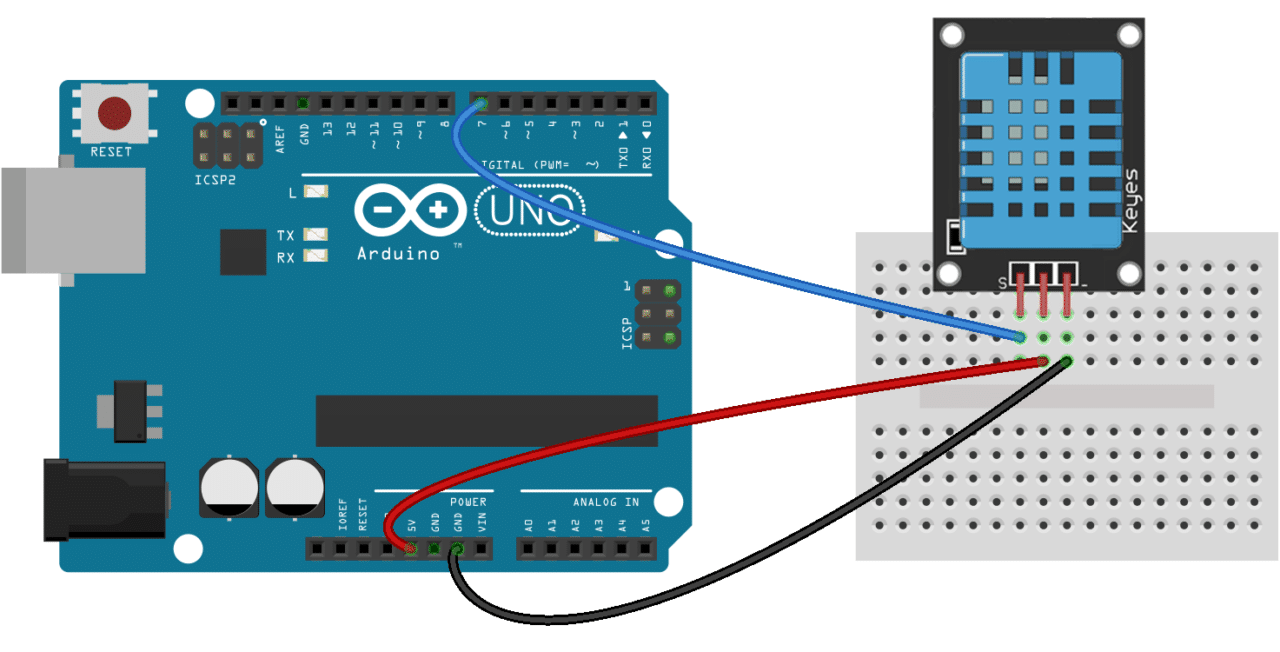


**FIG3.** Temperature and humidity sensor

**WORKING PRINCIPLE:**

* A temperature and humidity sensor, like the DHT22, is connected to an Arduino in a smart public toilet.
* The sensor measures ambient temperature and humidity levels.
* The Arduino reads data from the sensor through its digital pins and converts it into meaningful values. These values can be displayed on an LCD screen or sent to a central control system for monitoring and control.
* The system can trigger actions like activating fans or heaters based on these readings, ensuring user comfort and efficient energy use.
* This makes the smart public toilet more user-friendly and environmentally friendly.

**PIN CONNECTION WITH ARDUINO:**



* Connect GND Pin of T&H sensor to GND on Arduino
* Connect Vcc Pin of T&H sensor to 11V on Arduino
* Connect Data Pin of T&H sensor to D3 on Arduino

**PYTHON SCRIPT FOR TEMPERATURE AND HUMIDITY SENSOR:**

import serial

import time

ser = serial.Serial('COM3', 9600) # Replace 'COM3' with the correct serial port

try

while True:

serial\_data = ser.readline().decode('utf-8').strip()

if serial\_data:

print(serial\_data)

time.sleep(2) # Read data every 2 seconds

except KeyboardInterrupt:

ser.close()

**ALGORITHM:**

The code you provided is an Arduino sketch for reading data from a DHT (Digital Humidity and Temperature) sensor and displaying the humidity and temperature values on the serial monitor. Here’s an algorithmic description of what the code does:

Include the DHT library for interfacing with the sensor.

Define constants for the DHT sensor pin (DHTPIN) and its type (DHTTYPE).

Create a DHT object named “dht” and associate it with the specified pin and sensor type.

In the setup function:

a. Initialize the serial communication with a baud rate of 9600.

b. Initialize the DHT sensor.

In the loop function:

a. Add a delay of 2 seconds to wait between sensor readings.

b. Read the humidity and temperature values from the DHT sensor.

c. Check if the sensor reading has failed by using the isnan() function.

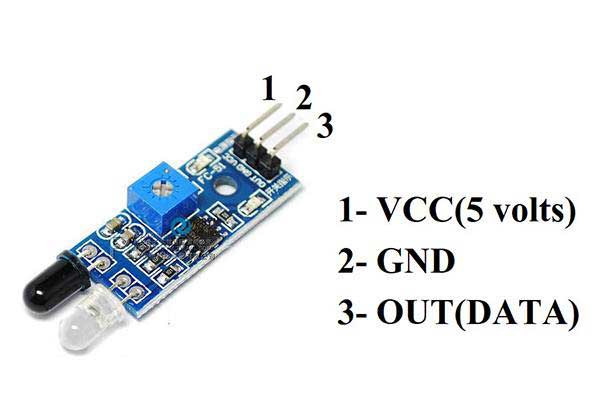
d. If the reading has failed, print an error message and exit the loop.

e. If the reading is successful, print the humidity and temperature values to the serial monitor with appropriate labels.

This algorithm describes the high-level steps and logic of the code. The code reads data from the DHT sensor, checks for errors, and displays the results on the serial monitor.

**OCCUPANCY SENSOR:**

**Infrared** **Sensors**: These sensors can detect the presence of a user within the toilet stall. When someone enters, the sensor triggers the lighting and ventilation systems to turn on, ensuring a well-lit and comfortable environment.

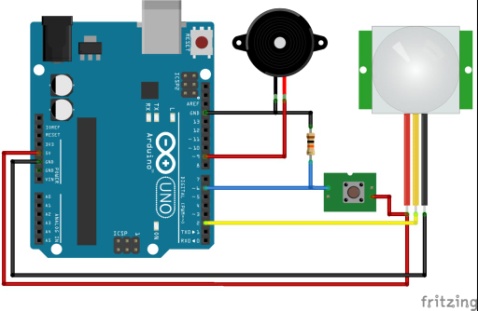


**FIG11.** INFRAREDSENSOR

**WORKING PRINCIPLE:**

* An infrared occupancy sensor in a smart public toilet operates by emitting infrared light.
* When a person enters, their body heat reflects this light back to the sensor.
* Arduino processes the signal, triggering actions like activating lights, flushing, or opening doors.
* The sensor continually monitors for changes in the infrared signal, allowing it to detect occupancy and respond accordingly.
* This efficient, non-contact method ensures an automated and hygienic user experience in the smart toilet facility.

**PIN CONNECTION WITH ARDUINO:**

****

* Connect the VCC pin of the infrared sensor to the 11V pin of the Arduino.
* Connect the GND pin of the infrared sensor to the GND pin of the Arduino.
* Connect the OUT pin of the infrared sensor to any digital pin of the Arduino.

**PYTHON SCRIPT FOR IR SENSOR:**

import serial

import time

ser = serial.Serial('COM3', 9600) # Replace 'COM3' with the correct serial port

try:

while true:

serial\_data = ser.readline().decode('utf-8').strip()

if serial\_data:

ir\_value = int(serial\_data)

if ir\_value == 0:

print("Infrared sensor detected an object.")

else:

print("Infrared sensor did not detect an object.")

time.sleep(2) # Read data every 2 seconds

except KeyboardInterrupt:

ser.close()

**ALGORITHM:**

The code provided is an Arduino sketch for reading digital data from an IR sensor connected to a digital pin and displaying the sensor values on the serial monitor. Here's an algorithmic description of what the code does:

Define a constant "ir Sensor Pin" to specify digital pin connected to the IR sensor.

In the setup function:

a. Initialize serial communication with a baud rate of 9600 to communicate with the computer.

b. Set the "irSensorPin" as an input using pinMode() to prepare it for reading digital data.

In the loop function:

a. Read the digital value from the IR sensor using digitalRead(irSensorPin) and store it in the "sensorValue" variable.

b. Print the "sensorValue" to the serial monitor using Serial.println() to send the data to the computer for monitoring.

c. Add a delay of 1000 milliseconds (1 second) using delay(1000) to pause before taking the next reading. This delay is used to avoid flooding the serial monitor with data too quickly.

* The algorithm outlines the key steps and logic of the code, which continuously reads the digital value from the IR sensor and sends it to the serial monitor for observation.

**Develop a Python script on the loT sensors to send real-time occupancy and cleanliness data to the restroom information platform.**

Import serial

Import time

# Set the COM port where your Arduino is connected

Arduino\_port = ‘COM3’ # Change this to the appropriate port

# Set the baud rate to match your Arduino’s configuration

Baud\_rate = 9600

# Open a connection to the Arduino

Ser = serial.Serial(arduino\_port, baud\_rate)

Try:

While True:

# Read data from the Arduino

Arduino\_data = ser.readline().decode().strip()

# Split the data into sensor values

Sensor\_values = arduino\_data.split(‘,’)

If len(sensor\_values) == 4:

Mq2\_value, ir\_sensor\_value, temperature, humidity = sensor\_values

Print(f’MQ-2 Value: {mq2\_value}, IR Sensor Value: {ir\_sensor\_value}’)

Print(f’Temperature: {temperature}°C, Humidity: {humidity}%’)

Else:

Print(“Invalid data received from Arduino”)

# Delay for a moment before reading again

Time.sleep(1)

Except KeyboardInterrupt:

Print(“Exiting the program”)

Finally:

Ser.close()

**SAMPLE OUTPUT** :

MQ-2 Value: 123, IR Sensor Value: 456

Temperature: 25°C, Humidity: 40%

MQ-2 Value: 124, IR Sensor Value: 457

Temperature: 26°C, Humidity: 41%

MQ-2 Value: 125, IR Sensor Value: 458

Temperature: 27°C, Humidity: 42%

...

**CONCLUSION:**

Finally our conclusion is that, IOT-enabled smart public restroom project has demonstrated significant advantages in terms of efficiency, cleanliness, and user satisfaction. The real-time monitoring of restroom conditions, including occupancy and supply levels, ensures optimal resource allocation. User-friendly mobile apps enhance the overall experience by providing essential information. Cost savings through predictive maintenance and resource optimization make this system economically viable. Furthermore, the data collected can be used for data-driven decision-making and future improvements. Overall, our smart restroom solution offers a tangible and beneficial upgrade to public facilities.

NOTE: File naming Convention: IOT\_Phase3