

SMART_TOILET_USING_IOT

(PHASE 1-NAAN_MUDHALAVAN)

Why we approach IOT in Smart_Toilet?

Using temperature IoT sensors in a smart toilet, as opposed to a normal toilet, can serve various purposes and enhance user experience and comfort. Here are some reasons why temperature sensors might be incorporated into a smart toilet

Temperature Comfort Control: Temperature sensors can be used to control features that enhance comfort, such as heated toilet seats or bidets. These sensors ensure that the toilet's features are at a comfortable temperature, which can be customized by the user.

Smart Alerts: Temperature sensors can help detect anomalies or issues in the toilet's plumbing system. For example, a sudden drop in water temperature could indicate a malfunction in the water heating system, and the IoT system can alert maintenance or the user.

Personalization: Temperature sensors allow users to personalize their toilet experience. Users can set their preferred temperature for various features, ensuring a tailored and comfortable experience.

PHASE_1 Project:

- To create a more advanced IoT-based smart toilet system:
- Replace the LED control with code to control an actual light or lighting system in your smart toilet.
- Integrate Wi-Fi or other communication protocols to make your system IoT-enabled, so you can control the lights remotely or collect data.
- Add additional sensors or features as needed, such as occupancy detection, environmental monitoring, and user interfaces for control and monitoring.

- Remember to choose appropriate components and take safety precautions when working with electrical circuits.

Step 1: Define Your Requirements

- ✚ Determine what temperature range you want to monitor in the smart toilet. Decide on the type of temperature sensor you'll use (e.g., DHT22, DS18B20). Define whether you want real-time monitoring, data logging, or remote access.

Step 2: Gather Hardware and Software

- ✚ Collect the necessary hardware components, including the selected temperature sensor, an IoT-capable microcontroller (e.g., Arduino, Raspberry Pi), and Wi-Fi or Ethernet connectivity (e.g., ESP8266 or ESP32 for Wi-Fi). Set up your development environment by installing the required software (Arduino IDE for Arduino boards, Raspberry Pi OS for Raspberry Pi).

Step 3: Sensor Setup

- ✚ Connect the temperature sensor to your microcontroller following its datasheet or pinout. Ensure that the sensor is properly powered and grounded. If using a digital sensor (like DS18B20), make sure to install any required libraries for communication.

Step 4: Code Implementation

- ✚ Write the code to read temperature data from the sensor. Here's a simplified example using an Arduino and a DHT22 sensor.

```
#include <DHT.h>
```

```
#define DHTPIN 2
```

```
#define DHTTYPE DHT22
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
}  
void loop() {  
  delay(2000); // Delay for 2 seconds between readings.  
  
  float temperature = dht.readTemperature();  
  if (!isnan(temperature)) {  
    Serial.print("Temperature: ");  
    Serial.print(temperature);  
    Serial.println(" °C");  
  } else {  
    Serial.println("Failed to read temperature from DHT sensor!");  
  }  
}
```

Step 5: IoT Connectivity

- Implement code to connect your microcontroller to the internet using Wi-Fi or Ethernet. Depending on your project, you may need to set up Wi-Fi credentials, use MQTT for communication, or send data to a cloud platform.

Step 6: Data Storage and Visualization

- Choose where to store the temperature data. Options include a local database, cloud platforms (e.g., AWS, Google Cloud, ThingSpeak), or a dedicated IoT platform (e.g., Adafruit IO, Cayenne). Implement code to send temperature data to your chosen storage or visualization platform.

Step 7: Power Supply and Enclosure

- ✚ Ensure your smart toilet system has a reliable power supply, especially if it's installed in a remote location. Consider an appropriate enclosure to protect the electronics from moisture and humidity in a bathroom environment.

Step 8: Testing and Debugging

- ✚ Test the system thoroughly to make sure it accurately measures and reports temperature. Debug and refine your code and hardware as needed.

Step 9: Deployment

- ✚ Install the temperature detection system in your smart toilet. Monitor and maintain the system over time to ensure its reliability. Remember that this is a simplified overview, and the actual implementation can vary based on your specific requirements and hardware choices. Additionally, ensure that you follow safety precautions when working with electrical components in a bathroom environment.

CONCLUSION:

- IoT sensors, including temperature sensors, are integrated into smart toilets to bring a range of benefits and features that enhance user comfort, energy efficiency, health monitoring, and overall bathroom experience.
- These sensors enable the smart toilet to provide personalized settings, optimize energy usage, and ensure safe and comfortable operation. Smart toilets with IoT capabilities represent a modern and luxurious approach to bathroom technology, catering to the needs and preferences of users while promoting convenience and efficiency in the bathroom environment.