Smart water system

Build IoT Water Monitor

Building an IoT water consumption monitoring system involves several steps. Here's a high-level overview of the process:

1. **Select IoT Sensors:** Choose appropriate IoT sensors like flow meters to measure water consumption. Ensure they are compatible with your IoT platform.

2. **Hardware Setup:** Physically install the sensors at the locations you want to monitor water consumption. Connect them to a microcontroller or single-board computer like Raspberry Pi.

3. **IoT Platform Selection:** Choose an IoT platform or framework for data collection and sharing. Popular options include AWS IoT, Google Cloud IoT, or Microsoft Azure IoT.

4**. Set up IoT Device**: Configure the IoT device (Raspberry Pi, Arduino, etc.) with the necessary libraries and dependencies. Ensure it has internet connectivity, either through Wi-Fi or cellular.

5. **Python Script Development:** Develop a Python script that reads data from the sensors. The script should be programmed to collect data at regular intervals (e.g., every minute) and send it to the chosen IoT platform. Use appropriate libraries to interface with the sensors (e.g., GPIO libraries for Raspberry Pi).

6. **Data Transmission:** In your Python script, use MQTT, HTTP, or other relevant protocols to send the data to your chosen IoT platform. You'll need authentication credentials to connect to the platform.

7. **Data Storage and Processing:** Set up data storage and processing on the IoT platform. Store the incoming data and design rules or triggers for data analysis, if needed.

8. Real-Time Monitoring: Implement real-time monitoring and visualization. You can use dashboards or applications to display the water consumption data in a user-friendly manner.

9. **Alerts and Notifications:** Configure alerts or notifications in your IoT platform to trigger actions when certain conditions are met, such as water leakage or abnormal consumption.

10. **Security:** Ensure that your system is secure. Use encryption and access controls to protect the data and the IoT devices.
11. **Testing**: Thoroughly test your system to ensure the sensors, script, data transmission, and platform integration work seamlessly.
12. **Deployment**: Deploy the IoT water consumption monitoring system in public places where you want to monitor water usage.

Program:

```
import time
import RPi.GPIO as GPIO
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

# Set up GPIO pins for the flow meter
FLOW_METER_PIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(FLOW_METER_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# AWS IoT Configuration
IoT_ENDPOINT = “your-iot-endpoint”
IoT_PORT = 8883
IoT_ROOT_CA_PATH = “path/to/root-CA.crt”
IoT_PRIVATE_KEY_PATH = “path/to/private-key.key”
IoT_CERT_PATH = “path/to/certificate.crt”
IoT_CLIENT_ID = “your-client-id”

# Initialize the AWS IoT client
awsIoTClient = AWSIoTMQTTClient(IoT_CLIENT_ID)
awsIoTClient.configureEndpoint(IoT_ENDPOINT, IoT_PORT)
awsIoTClient.configureCredentials(IoT_ROOT_CA_PATH, IoT_PRIVATE_KEY_PATH, IoT_CERT_PATH)
```

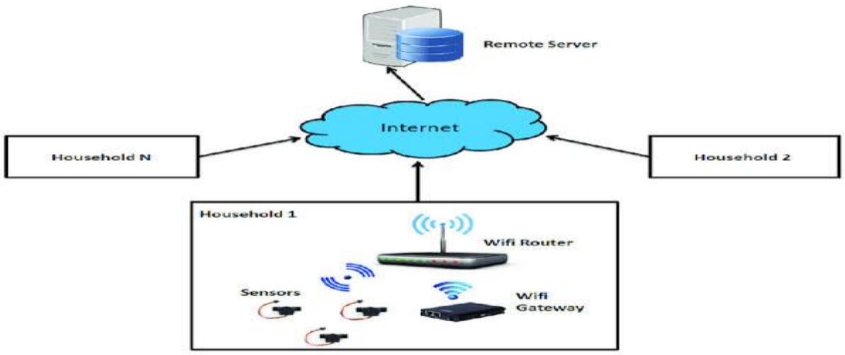
```
# Connect to AWS IoT
awsIoTClient.connect()


def on_flow_pulse(channel):
    # This function is called whenever a pulse is detected from the flow meter
    # You can customize this function to calculate and send the water consumption data
    # In this example, we simply send a static value for demonstration purposes
    Water_consumed = 0.5  # Change this to your actual calculation
    Message = ‘{“water_consumed”: ‘ + str(water_consumed) + ‘}’
    awsIoTClient.publish(“water_consumption”, message, 1)


# Add an event listener for the flow meter
GPIO.add_event_detect(FLOW_METER_PIN, GPIO.RISING, callback=on_flow_pulse)


Try:
    While True:
        Time.sleep(1)
Except KeyboardInterrupt:
    GPIO.cleanup()
```

Diagram:


Remote Server
Internet
Household N
Household 2
Household 1
Wifi Router
Sensors
Wifi
Gateway