**DEVELOPMENT PART-1 PHASE-3**

|  |  |
| --- | --- |
| DATE | 18/10/2023 |
| PROJECT NAME | SMART WATER MANAGEMENT |

**Start building the IoT water consumption monitoring system.**

* Building an IoT water consumption monitoring system involves several components, including hardware, software, and networking technologies. Here's a step-by-step guide to help you get started:

**Define System Requirements:**

Functionality: Determine what features you want in your system, such as real-time monitoring, historical data analysis, leak detection, and user notifications.

Hardware: Choose appropriate sensors to measure water consumption. Flow sensors or smart water meters are common choices.

Software: Decide on the platform and programming languages for developing the software components.

Networking: Consider communication protocols (like MQTT, HTTP, or CoAP) and whether you'll use Wi-Fi, cellular, or other IoT-specific networks.

**Select Hardware:**

Choose compatible sensors for measuring water consumption accurately.

Opt for microcontrollers (like Arduino or Raspberry Pi) to interface with sensors and send data to the cloud.

**Connect Sensors to Microcontroller:**

Connect the water flow sensors to the microcontroller following the datasheet or the sensor’s technical documentation.

Write code to read data from the sensors. Test and calibrate the sensors for accuracy.

**Choose IoT Platform:**

Select an IoT platform (such as AWS IoT, Microsoft Azure IoT, or Google Cloud IoT) to manage devices, store data, and enable communication between devices and the cloud.

Set up the necessary accounts and configurations on your chosen IoT platform.

**Develop Firmware:**

Write firmware for the microcontroller to collect data from sensors.

Implement error handling and data validation in the firmware.

Include code for connecting the microcontroller to the IoT platform using the chosen communication protocol.

**Configure IoT sensors (e.g., flow meters) to measure water consumption in public places.**

**Flow meters:**

Flow meters are devices used to measure the rate of flow or quantity of a gas or liquid moving through a pipe. They are essential instruments in various industries such as manufacturing, petrochemical, water treatment, and food and beverage, where accurate measurement of fluid flow is crucial for process control and monitoring. There are several types of flow meters, each with its own working principle and applications. Here are some common types of flow meters:

**Mechanical Flow Meters:**

**Rotary Vane Meters:**

These meters use rotating blades to measure the flow of liquids.

**Turbine Meters:**

Turbine flow meters have a rotor with blades placed axially along the flow. The flow of liquid or gas causes the rotor to spin.

**Oval Gear Meters:**

Oval gear meters use two oval gears that rotate to measure the flow.

**Differential Pressure Flow Meters:**

**Orifice Meters:**

Orifice meters use a plate with a hole (orifice) to create a pressure drop, which is correlated to the flow rate.

**Venturi Meters:**

Venturi meters have a constricted throat that accelerates the fluid flow, creating a pressure drop that is used to calculate the flow rate.

**Pitot Tubes:**

Pitot tubes use a combination of static and dynamic pressure to determine the flow velocity.

**Ultrasonic Flow Meters:**

**Transit-Time Flow Meters:**

These meters use ultrasonic signals to measure the time it takes for an ultrasonic pulse to travel with and against the flow, determining the flow rate.

**Doppler Flow Meters:**

Doppler flow meters use the frequency shift of reflected ultrasonic waves from moving particles or bubbles in the fluid to calculate flow velocity.

**Electromagnetic Flow Meters:**

These meters use Faraday's law of electromagnetic induction to measure the flow of conductive liquids. A magnetic field is applied to the liquid, and the induced voltage is proportional to the flow rate.

**Coriolis Mass Flow Meters:**

Coriolis flow meters operate on the principle of Coriolis effect, where the fluid flow causes a tube to twist, and this twisting is proportional to the mass flow rate.

**Vortex Flow Meters:**

Vortex flow meters utilize the principle that a vortex shedding occurs behind an obstruction placed in a flow stream. The frequency of vortex shedding is proportional to the flow rate.

**Thermal Flow Meters:**

Thermal flow meters measure the mass flow rate of gases based on the convective heat transfer from a heated sensor to the flowing fluid.

**Develop a Python script on the IoT sensors to send real-time water consumption data to the data-sharing platform.**

**Firstly, you need to install the MQTT library using pip if you haven't already:**

**Bash:**

pip install paho-mqtt

Here's a basic Python script to send real-time water consumption data to an MQTT broker:

**Python:**

import paho.mqtt.client as mqtt

**coding:**

import time

import random

# MQTT broker information

mqtt\_broker\_host = "mqtt.yourbroker.com"

mqtt\_broker\_port = 1883

mqtt\_topic = "water\_consumption"

# Function to simulate water consumption data

def get\_water\_consumption\_data():

# Simulate water consumption data (in liters)

return random.randint(1, 100)

# MQTT on\_connect callback

def on\_connect(client, userdata, flags, rc):

print("Connected with result code " + str(rc))

# MQTT on\_publish callback

def on\_publish(client, userdata, mid)

print("Message Published")

# Create an MQTT client

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_publish = on\_publish

# Connect to the MQTT broker

client.connect(mqtt\_broker\_host, mqtt\_broker\_port, 60)

# Loop to send real-time water consumption data

**while True:**

# Get water consumption data

water\_consumption = get\_water\_consumption\_data()

# Publish the data to the MQTT broker

client.publish(mqtt\_topic, str(water\_consumption))

# Wait for 1 second before sending the next data

time.sleep(1)

1. 'mqtt\_broker\_host' is the hostname of your MQTT broker.
2. 'mqtt\_broker\_port' is the port on which the MQTT broker is running (usually 1883).
3. 'mqtt\_topic' is the topic to which the water consumption data will be published.

This script generates random water consumption data (between 1 and 100 liters) every second and publishes it to the specified MQTT topic. Adjust the 'get\_water\_consumption\_data' function if you have a real sensor or data source to fetch the actual water consumption data.