**IBM NAAN MUDHALVAN**

**SKILL UP**

**PROJECT TITLE: SMART WATER SYSTEM**

**COLLEGE: PERI INSTITUTE OF TECHNOLOGY**

**DEPT: ELECTRONICS AND COMMUNICATION ENGINEERING**

**DOMAIN: INTERNET OF THINGS (IOT)**

**Submitted By**

**AVINASH.S (411521106010)**

**PHASE-1**

* 1. **Abstract:**

* Water is one of the essential parts of life. Water pollution is one of the big problems to the world. In order to ensure the safe supply of the drinking and useful water for different purposes like agricultural, the water should be monitored. This document presents a design of a low cost system for real time monitoring of the water quality and quantity of water in IOT (internet of things). The system having of several sensors is used to measuring physical of the water. The parameters flow sensor of the water can be measured. The measured values from the sensors can be processed by the controller. The Arduino model can be used as a controller. Finally, the sensor data can be shown on internet using WI-FI system. A cloud server was configured as data saving and analysis. This data can be used in future research and development.

* 1. **Introduction:**

* Currently drinking water is very prized for all the humans. In recent times water levels are very low and water in the lakes are going down. So its too important to find the solution for water monitoring & control system. IoT is a solution. In recent days, development in computing and electronics technologies have triggered Internet of Things technology . This paper present a low cost water monitoring system, which is a solution for the water wastage and water quality. Microcontrollers and sensors are used for that system. Ultrasonic Sensor is used to measuring water level. The other parameters like pH, TDS, and Turbidity of the water can be calculated using different corresponding sensors. This system use the flow sensor which can measure the water flow and if the necessary quantity of water flow through the pipe then water flow can be stopped automatically. The calculated values from the sensors can be processed by the Microcontrollers and uploaded to the internet through the Wi-Fi module (ESP 8266).

* 1. **Project Definition:**

* The project involves implementing IoT sensors to monitor water consumption in public places such as parks and gardens. It employs advanced technology to monitor and provide immediate insights into water usage. This empowers individuals and industries to make informed decisions, leading to more efficient and sustainable water practices.

* 1. **Objective:**

* The objective of this project is to promote water conservation by providing real-time water consumption data to the public. This empowers individuals, communities, and industries to make informed decisions and take proactive measures to reduce water wastage and improve overall efficiency in water usage. By leveraging technology and data transparency, the project aims to foster a culture of responsible water management for a more sustainable future.

* 1. **IoT Sensor Design:**

* In this project many components such as sensors, modules , power sources and so on are used
* **Sensors:**
* Flow Sensor
* Ultrasonic sensor
* **Connectivity:**
* Wi-fi module
* **Power Source:**
* Battery / Solar
* **Data Processing:**
* Data processing will send raw data to a central processing unit..

* 1. **Integration Approach :**

* Integrating real-time water consumption data involves deploying IoT sensors strategically within the water supply network. These sensors collect continuous data on water usage, which is then transmitted to a central processing unit using a reliable communication protocol. This unit serves as the core for aggregating, parsing, and validating the incoming data. Real-time analytics are applied for instant processing, enabling prompt identification of consumption patterns. The validated data is stored in a structured database, and a user-friendly interface, such as a web-based dashboard or mobile app, is created for visualization. Security measures, including encryption and access controls, are implemented to safeguard data integrity. Thorough testing and documentation ensure the system's reliability, accuracy, and scalability, making it an efficient tool for water management and conservation efforts.

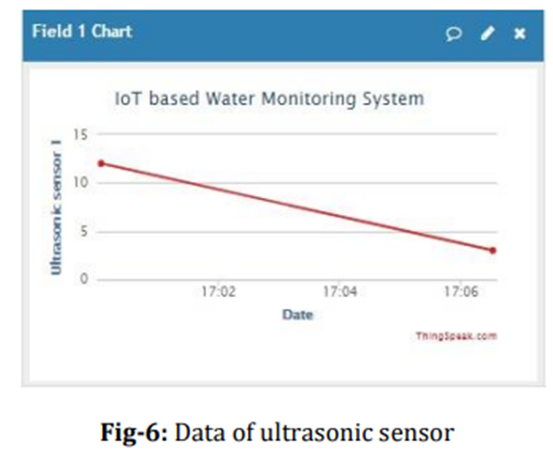
**PHASE-2**

**2.1 Innovation:**

* In this, we present the theory on real time monitoring of water quality and quantity using IoT. The system consists of Arduino, microcontroller, different type of sensors like water flow sensor, pH and turbidity sensor and ultrasonic sensor. The Arduino is the main processor of the system which control and process the data generated by the sensors. A Wi-Fi module is connected to the Arduino device which help to transfer the data to the cloud over internet. The ultrasonic sensor helps to measure the water level when the water flow reach certain level then the water flow can be stopped automatically by turning the motor off or close the water flow in pipe by the help of Arduino. The water flow sensor measure the quantity of water flow through the pipe in a given time, this data will be sent to cloud for storage and analysis purposes. The other sensor like temperature, pH and turbidity sensor measure the water quality and help to determine whether the water is useful for drinking or any agricultural purposes.

**2.2 How To Use Cloud?**

* This system is using Wi-Fi module (Esp8266) to send the sensor data to the cloud. All the sensors are connected with Wi-Fi module. Wi-Fi module needs the internet. So here Mobile data or Wi-Fi is the access point for the internet. And after all this data sends to the cloud. The following figure show the data stored in cloud



**2.3 Hardware and software requirements :**

* **IoT Sensors:** Choose water flow sensors or meters that can accurately measure water usage. These sensors should be durable and weather-resistant for outdoor use.
* **Communication Modules:** You'll need communication modules like Wi-Fi, LoRa, or cellular modems to transmit data from the sensors to a central server or cloud platform.
* **Microcontrollers:** Use microcontrollers (e.g., Arduino, Raspberry Pi) to interface with sensors and manage data transmission.
* **Power Supply:** Depending on the location, you may need solar panels or long-lasting batteries to power the sensors and IoT devices.
* **Enclosures:** Protect the sensors, microcontrollers, and other hardware from environmental elements with suitable enclosures.
* **Data Storage Devices:** To store data locally if needed, consider using SD cards or external storage devices.
* **Gateway Devices:** These act as intermediaries between the sensors and the central server, aggregating data and sending it to the cloud. They may also perform data preprocessing.
* **Software Requirements:**
* **Embedded Software:** Develop or configure firmware for the microcontrollers to collect and transmit sensor data. This software should handle sensor interfaces, data formatting, and communication protocols.
* **Cloud Platform:** Choose a cloud platform such as AWS, Azure, or Google Cloud to securely store and manage the data. These platforms provide scalable and reliable data storage and processing capabilities.
* **Database Management System:** Utilize a database system (e.g., MySQL, PostgreSQL, MongoDB) to store and manage historical data.
* **Data Analysis Tools:** Implement software for real-time or batch data analysis to provide insights into water consumption trends. This might include machine learning algorithms for predictive analytics.
* **User Interface (UI):** Develop a user-friendly web or mobile application for end-users to access water consumption data. Include features for real-time monitoring, historical data retrieval, and customizable alerts.
* **Security Measures:** Implement strong security protocols to protect data transmission, storage, and access. Use encryption, authentication, and authorization mechanisms.
* **Notification System:** Set up alerting and notification systems to inform users or administrators of unusual water consumption patterns or issues.
* **Remote Management:** Create tools for remote management and configuration of the IoT sensors and devices.
* **APIs:** Develop or utilize APIs for integration with other systems and for data sharing with relevant authorities or stakeholders.
* **Scalability and Redundancy:** Design the system to be scalable to accommodate additional sensors and provide redundancy for critical components.
* **Data Visualization:** Use data visualization tools (e.g., Tableau, Grafana) to create intuitive dashboards and reports for users to easily interpret water consumption data.
* **Maintenance and Diagnostics:** Implement remote diagnostic tools to monitor the health of IoT sensors and devices and perform maintenance tasks as necessary.

**2.4 Program:**

Python program that simulates an IoT sensor reading for water consumption monitoring. This program generates random water consumption data and sends it to a cloud-based database, mimicking the behavior of a real sensor. Note that this is a simplified example, and in a real-world scenario, you would need to replace the random data generation with actual sensor data and set up a cloud database for storage.

Code:

import random

import time

from datetime import datetime

import pymongo # You'll need to install the pymongo library

# Replace these values with your MongoDB credentials

MONGODB\_URI = "your\_mongodb\_uri"

DB\_NAME = "your\_database\_name"

COLLECTION\_NAME = "water\_consumption"

# Initialize the MongoDB client

client = pymongo.MongoClient(MONGODB\_URI)

db = client[DB\_NAME]

collection = db[COLLECTION\_NAME]

def generate\_water\_consumption():

# Simulate water consumption data (replace this with real sensor data)

return random.uniform(0.1, 2.0) # Liters per minute

def send\_data\_to\_database(data):

timestamp = datetime.now()

data\_entry = {

"timestamp": timestamp,

"water\_consumption": data

}

collection.insert\_one(data\_entry)

print(f"Data sent to the database: {data\_entry}")

if \_\_name\_\_ == "\_\_main\_\_":

try:

while True:

water\_usage = generate\_water\_consumption()

send\_data\_to\_database(water\_usage)

time.sleep(60) # Simulate data sent every minute

except KeyboardInterrupt:

print("Data generation stopped.")

**2.5 Conclusion:**

* In this paper, a prototype water monitoring system using IoT is presented. So this application will be the best challenger in real time monitoring & control system and use to solve all the water related problems.