**IBM NAAN MUDHALVAN**

**SKILL UP**

**PROJECT TITLE:**

**SMART WATER MANAGEMENT SYSTEM**

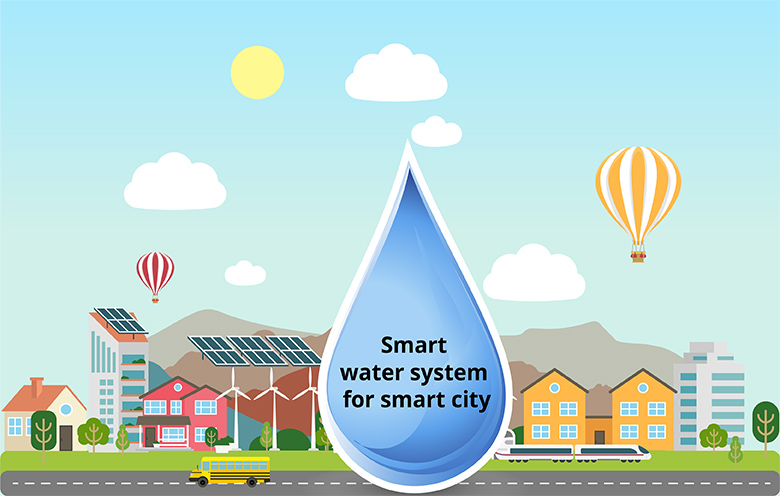
**COLLEGE: PERI INSTITUTE OF TECHNOLOGY**

**DEPT: ELECTRONICS AND COMMUNICATION ENGINEERING**

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

**Submitted By :**

**VIJAYA SRI Y (au411521106060)**



**PHASE-2 :**

* 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 it is 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.

**2.2 Need for smart water management:**

* The need for smart water management arises from several critical challenges that our global water systems face:

1. Water Scarcity and Stress.

2. Environmental Impact.

3. Climate Change.

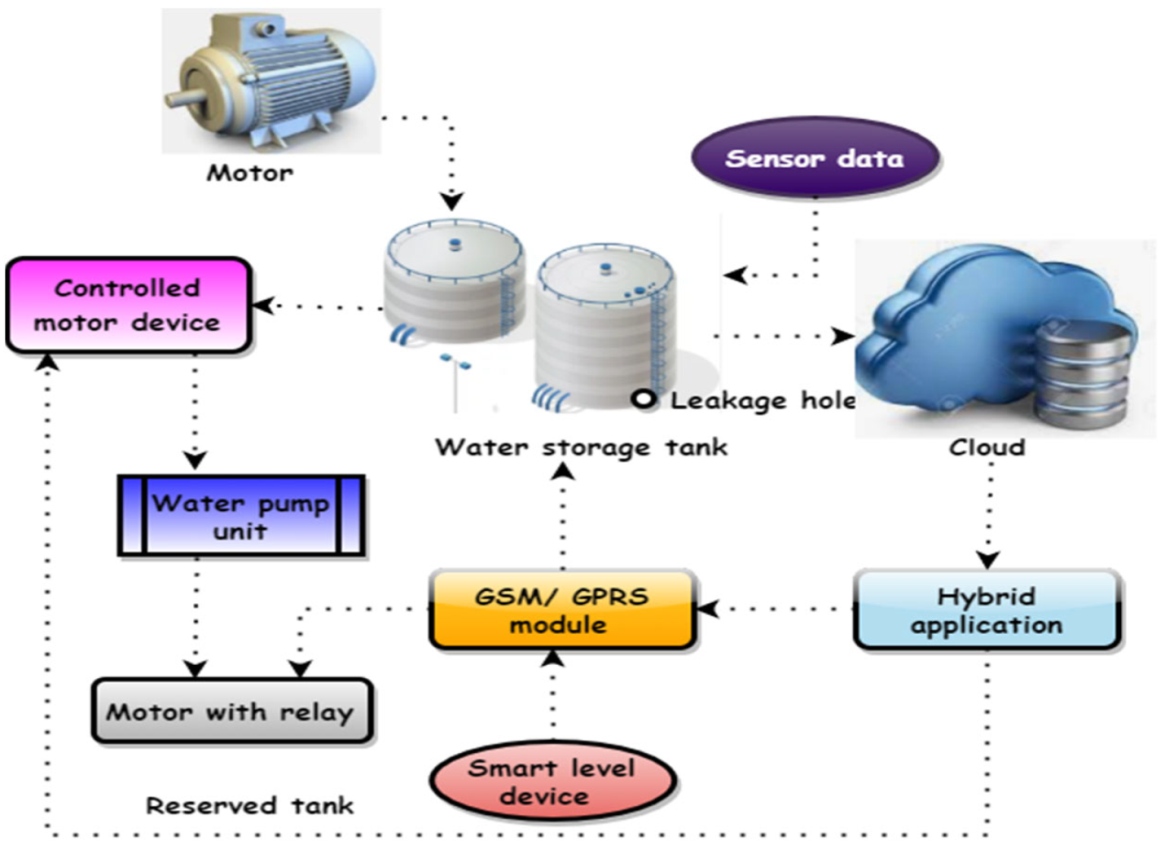
4. Urbanization.

5. Energy Consumption.

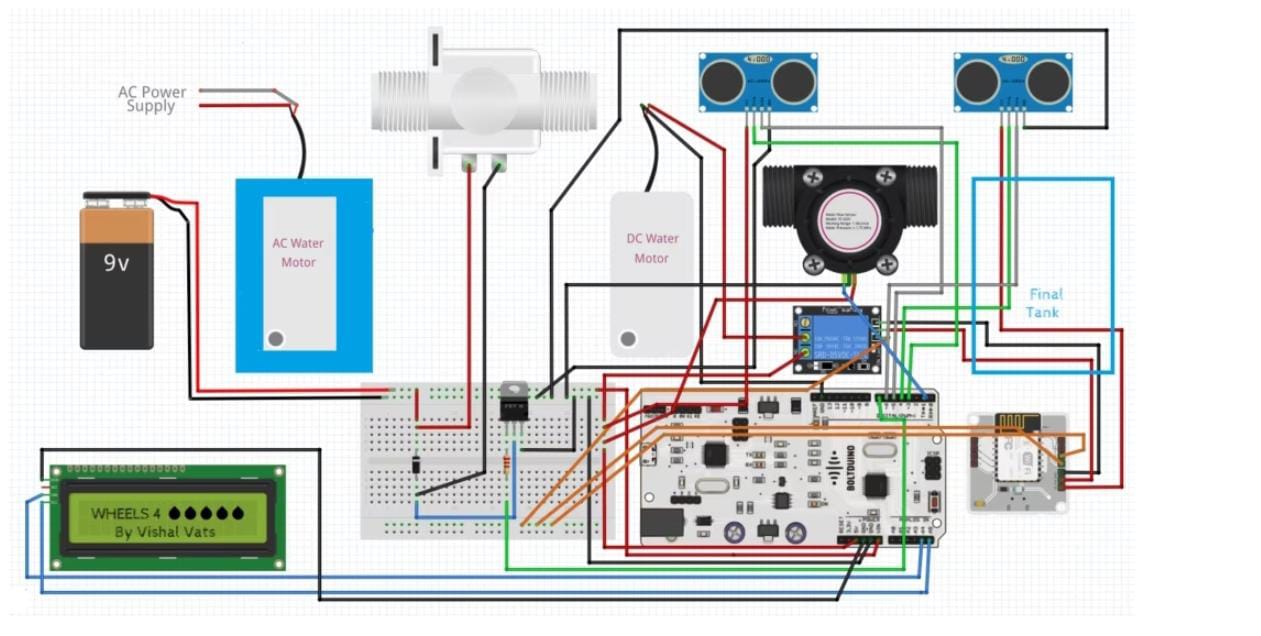
6. Resource Efficiency.

Overall, smart water management is essential for ensuring equitable access to clean water, protecting the environment, and building resilient communities in the face of increasing water-related challenges.

* 1. **Block Diagram:**

****

* 1. **Circuit Diagram :**



* 1. **Hardware and software requirements :**

**Hardware Requirements:**

1. **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.
2. **Communication Modules:** Communication modules like Wi-Fi, LoRa, or cellular modems to transmit data from the sensors to a central server or cloud platform.
3. **Microcontrollers:** Microcontrollers (e.g., Arduino, Raspberry Pi) are used to interface with sensors and manage data transmission.
4. **Power Supply:** Depending on the location, you may need solar panels or long-lasting batteries to power the sensors and IoT devices.
5. **Enclosures:** Protect the sensors, microcontrollers, and other hardware from environmental elements with suitable enclosures.
6. **Data Storage Devices:** To store data locally if needed, consider using SD cards or external storage devices.

**Software Requirements:**

1. **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.
2. **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.
3. **Database Management System**: Utilize a database system (e.g., MySQL, PostgreSQL, MongoDB) to store and manage historical data.
4. **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.
5. **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.
6. **Security Measures**: Implement strong security protocols to protect data transmission, storage, and access. Use encryption, authentication, and authorization mechanisms.
7. **Notification System**: Set up alerting and notification systems to inform users or administrators of unusual water consumption patterns or issues.
8. **Remote Management**: Create tools for remote management and configuration of the IoT sensors and devices.
9. **APIs**: Develop or utilize APIs for integration with other systems and for data sharing with relevant authorities or stakeholders.
10. **Scalability and Redundancy**: Design the system to be scalable to accommodate additional sensors and provide redundancy for critical components.
11. **Data Visualization**: Use data visualization tools (e.g., Tableau, Grafana) to create intuitive dashboards and reports for users to easily interpret water consumption data.
12. **Maintenance and Diagnostics**: Implement remote diagnostic tools to monitor the health of IoT sensors and devices and perform maintenance tasks as necessary.
    1. **Coding:**

Here, C++ language is used.

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

char auth[] = "YOUR\_BLYNK\_AUTH\_TOKEN";

char ssid[] = "YOUR\_WIFI\_SSID";

char pass[] = "YOUR\_WIFI\_PASSWORD";

#define WATER\_SENSOR\_PIN A0

#define PUMP\_PIN 2

int waterLevel;

BlynkTimer timer;

void setup() {

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

pinMode(WATER\_SENSOR\_PIN, INPUT);

pinMode(PUMP\_PIN, OUTPUT);

timer.setInterval(1000L, checkWaterLevel); // Check water level every second

}

void checkWaterLevel() {

waterLevel = analogRead(WATER\_SENSOR\_PIN);

Blynk.virtualWrite(V0, waterLevel); // Send water level data to Blynk

if (waterLevel < 500) { // Adjust threshold based on your sensor and setup

digitalWrite(PUMP\_PIN, HIGH); // Turn on the pump

} else {

digitalWrite(PUMP\_PIN, LOW); // Turn off the pump

}

}

void loop() {

Blynk.run();

timer.run();

}

* 1. **Conclusion:**
* Smart water management, enabled by IoT technology, revolutionizes how we monitor and conserve water. It provides real-time data, improves environmental sustainability, enhances public health, and strengthens communities. Ongoing collaboration, innovation, and investment are essential for a sustainable water future.