Smart Water Management
Project Documentation

Date	**23-10-2023**
Team ID	**536**
Project Name	**Smart Water Management**

Introduction

Smart Water Management (SWM) uses Information and Communication Technology (ICT) and real-time data and responses as an integral part of the solution for water management challenges. SWM is becoming an area of increasing interest as governments from around the world integrate smart principles into their urban, regional and national strategies. The potential application of smart systems in water management is wide and includes solutions for water quality, water quantity, efficient irrigation, leaks, pressure and flow, floods, droughts and much more.

Problem Statement

The current water management system in our region suffers from inefficiency, water losses, and a lack of real-time monitoring. The objective is to design, develop, and implement a “Smart Water Management” System that addresses these challenges by optimizing water distribution, reducing water wastage, enhancing system resilience, and promoting responsible water use."

Data:

Data from these sensors is collected and processed in realtime. The collected data provides insights into the condition of the water infrastructure and the availability and quality of water.

Design Thinking Approach

Empathize:

- Understand the needs and concerns of the community, government agencies, and businesses related to water management.
- Conduct interviews, surveys, and observations to gather insights from various stakeholders.

- Identify pain points and challenges in the current water management system.

Actions:

1. Conduct interviews with water management authorities to gather insights into their challenges in optimizing water usage.
2. Analyse historical water usage and environmental data to identify trends and patterns related to water usage efficiency.
3. Seek input from environmentalists to understand the factors that influence water conservation efforts.

Define:

- Clearly define the problem or opportunity based on the insights gained during the empathy stage.
- Create a problem statement that focuses on a specific aspect of smart water management, such as water conservation or quality improvement.
- Set specific goals and objectives for the project.

Objectives:

- Develop a model that optimizes water usage in smart water Management.
- Provide recommendations to water management authorities on efficient water usage strategies.

Ideate:

- Brainstorm innovative solutions to address the defined problem.
- Encourage diverse perspectives and creative thinking among your team members.
- Use techniques like mind mapping, brainstorming sessions, or workshops to generate ideas.

Actions:

1. Explore different regression algorithms suitable for water usage optimization.
2. Consider incorporating real-time data and foot traffic analysis as additional features.
3. Brainstorm ways to visualize and present the model's recommendations to water management authorities.

Prototype

- Develop low-cost, scaled-down prototypes of your smart water management solutions.
- Test these prototypes in controlled environments to gather feedback and refine them.
- Consider using emerging technologies like IoT sensors, data analytics, and AI for monitoring and optimizing water usage.

Action

1. Develop Python scripts or notebooks for data preprocessing, model training, and validation.
2. Create sample water usage predictions for a subset of Water Management.
3. Provide preliminary recommendations to water management authorities based on the prototype's insights.

Test

- Implement the refined prototypes in real-world scenarios.
- Collect data and feedback on the effectiveness of the solutions.
- Continuously refine and improve the solutions based on real-world performance and user feedback.
- Ensure scalability and sustainability of the chosen smart water management strategies.

Actions:

1. Apply the model to various water usage areas and assess its prediction accuracy.
2. Collect feedback from water management authorities regarding the relevance and usefulness of the recommendations.
3. Fine-tune the model based on feedback and additional data.

Implement

Implementing a smart water management system involves integrating technology, data analysis, and efficient practices to optimize water use and distribution.

Actions:

1. Apply the water usage optimization model to smart water Management in specific locations to guide water usage decisions.

2. Share predictions and recommendations with water management authorities.
3. Continuously monitor and improve the model's performance as more usage data becomes available.

Iterate

- Refine the model based on user feedback and testing results.
- Continuously improve the accuracy and reliability of predictions.

Actions:

1. Periodically retrain the model with updated Managing system and environmental data to improve accuracy.
2. Explore advanced techniques such as reinforcement learning for optimizing water usage.
3. Stay updated on industry trends and emerging data science techniques to enhance water usage optimization.

Design and Innovation Strategies:

Micro Controller Selection

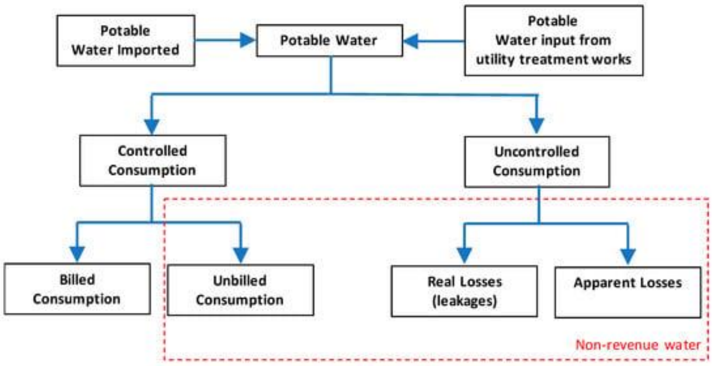
- For a Smart Water Management system using IoT, consider using a microcontroller like Arduino or Raspberry Pi.
- Arduino offers simplicity and low power, while Raspberry Pi provides more processing power for data analysis.
- Integrate sensors like water level detectors and rainfall sensors with IoT modules (e.g., ESP8266 or SIM800) for connectivity.
- Use a cloud platform such as AWS or Azure to process and analyze data, triggering alerts for early warnings.
- Remember to prioritize power efficiency and robust communication for reliable operation in remote areas.

Components :

Sensors Selection

For a Smart Water Management system using IoT, opt for water level sensors with real-time data transmission capabilities. Ultrasonic or pressure sensors are commonly employed to gauge water levels. Connect these sensors to an IoT platform for continuous monitoring. Ensure the system sends alerts based on predefined thresholds to enable timely warnings.

The volume of water in the RS, whether imported or extracted, is divided into billed water (BW) and NRW and even between controlled and uncontrolled consumption. The billed water is the consumed water that is directly charged to customers. The NRW is the volume that includes the water losses and the consumed volume by the authorized agents (e.g., social services, fire-fighting services). A simplifying schematic of this water balance is shown in Figure.



Cloud platform :

If water is flowing continuously than the expected time then it will be detected by water flow sensor and data will go to IOT server. User can see the real time data in IOT server and in this condition one notification will be sent to mobile App from IOT server. If the water PH level or dirt level is not good then the water will not be pumped into the tank. User can control water flow through mobile app by interacting with the server.

Overall design ie, System architecture is represented in Figure 1, which provides information of about how the components are interconnected and how components interact with each other. Sensors are placed in storage tank and overhead tank. In overhead tank three sensors are placed indicating levels of water along with these sensors PH sensor is also placed for checking purity.

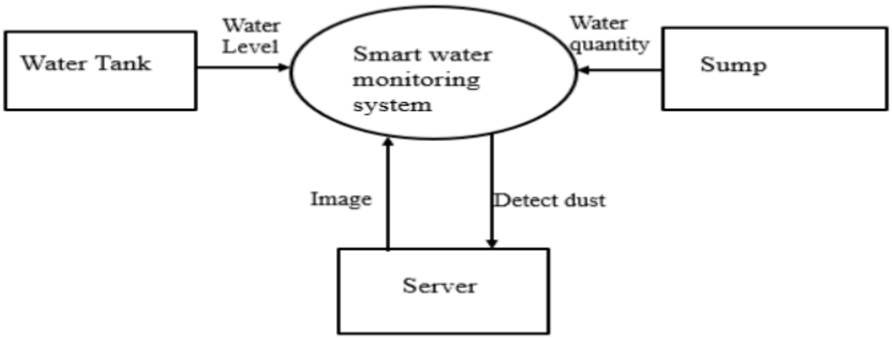
Any impurities present in storage tank are detected by using image processing concept. Pi camera module will capture the images and sends to server. Server will process the image and determines the status as either pure water or dirty water and then sends to the user android/mobile application. It also notifies the same to the user. The water from the water tank and in sump is monitored through android application via server connection. Presence of impurities in storage tank is detected by using image processing concept with Pi camera module.

Connectivity :

For Smart Water Management systems using IoT, AWS (Amazon Web Services) is often a popular choice.

Its scalable infrastructure and various services make it suitable for handling the data generated by IoT devices, analysing it in real-time, and triggering alerts or warnings.

AWS IoT services, along with data storage and analytics tools, can be seamlessly integrated into a comprehensive Smart Water Management System.



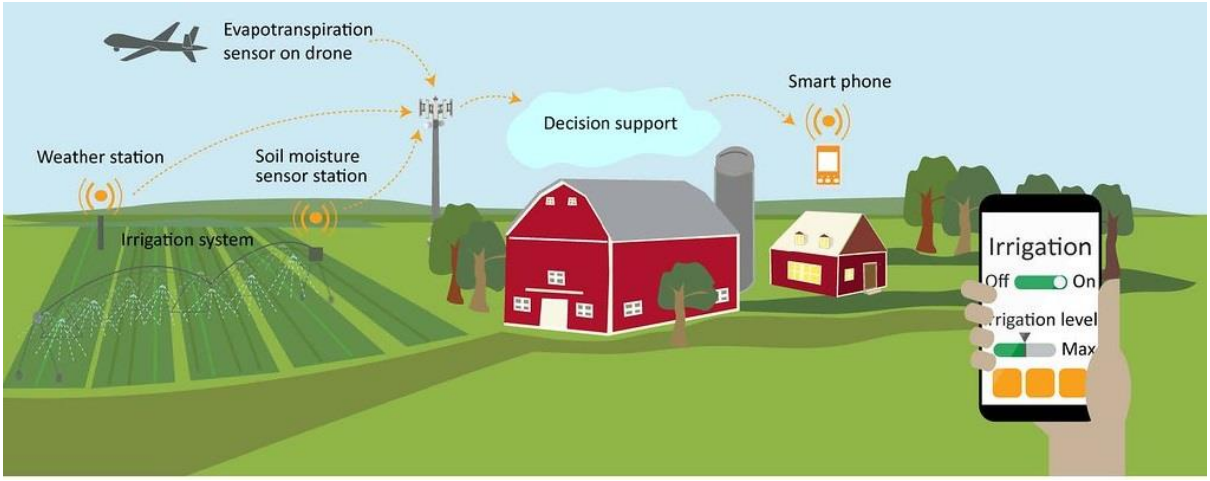
IoT-based Smart Water Management systems often utilize the MQTT (Message Queuing Telemetry Transport) protocol for efficient and real-time communication between sensors and the central monitoring system. MQTT's lightweight and publish-subscribe architecture make it ideal for transmitting data such as through the continuous analysis of sensor data, smart water systems can identify trends, assess the condition of infrastructure components, and predict maintenance needs.

This proactive approach to infrastructure maintenance extends the lifespan of critical assets, reduces the frequency of costly repairs, and ensures the consistent performance of the water distribution system. By optimizing infrastructure management, water utilities can allocate resources more efficiently, saving both time and financial resources. The synergy between real-time alerts and infrastructure optimization is a cornerstone of effective smart water management, ensuring the reliability, sustainability, and cost-effectiveness of water distribution networks while safeguarding the quality of water services for communities and consumers.

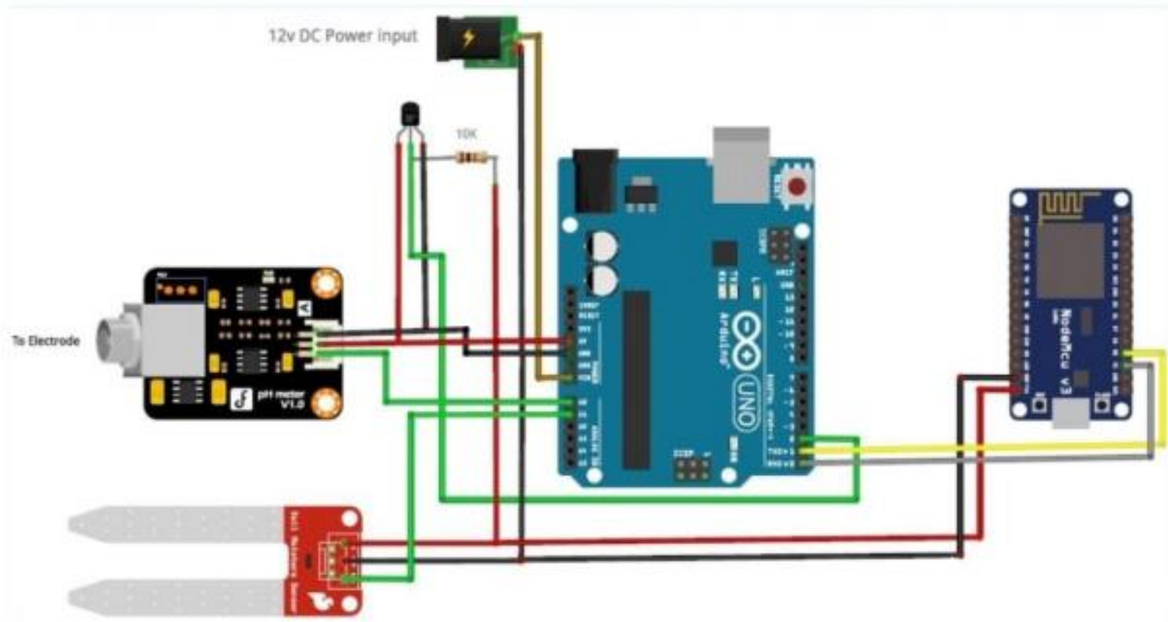
Use Case

Smart water management using IoT (Internet of Things) technology can be applied to various use cases to improve water resource efficiency, reduce waste, and ensure water quality. In agriculture, water is a precious resource, and its efficient use is critical for crop yield and environmental sustainability. Traditional irrigation methods often lead to water wastage due to over-irrigation or inadequate monitoring and control of water usage. This can result in both water and energy waste and can harm the environment. Implement a smart irrigation system using IoT technology to optimize water usage in agriculture. These sensors are placed in the soil to measure the moisture level. They provide real-time data on the soil's water content. These gather information on weather conditions, including temperature, humidity, wind speed, and precipitation forecasts. This data helps adjust irrigation schedules based on weather patterns. Collects data from the sensors and sends it to the central control system. It can also receive commands for controlling irrigation. An IoT-based platform that processes data from sensors and weather stations. It uses algorithms to make decisions on when and how much to irrigate. The system ensures that crops receive the right amount of water, reducing over-irrigation and water wastage. By optimizing irrigation, the system can also reduce energy consumption, as it requires less power to pump and distribute water. Proper irrigation leads to healthier crops and higher yields. Reduced water usage and more sustainable farming practices help preserve water resources and protect the environment. Farmers can monitor and control the irrigation system remotely, saving time and resources. Over time, the system collects valuable data that can be used for long-term planning and analysis to further improve water management practices.

Block Diagram



Schematic Diagram



Code

Since the C Code Libraries are available in Wokwi .

C Coding

```
#define BLYNK_TEMPLATE_ID "TMPL3g8NOyuBL"
#define BLYNK_TEMPLATE_NAME "WATER MANAGEMENT"
#define BLYNK_AUTH_TOKEN "keuIMDdFF9hQOue1x8ntz3OuNTInBbD0"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
//Your wifi credentials
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();
// Update state
  Blynk.virtualWrite(V1, value);
}

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
```

```
{
  // Change Web Link Button message to "Congratulations!"
  Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
  Blynk.setProperty(V3, "onImageUrl",  "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
  Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}


// This function sends Arduino's uptime ev   ery second to Virtual Pin 2.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V2, millis() / 1000);
}
#define trigerpin 16
#define echopin 17
#define buzzer 5
float distance,a, duration;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(trigerpin, OUTPUT);
  pinMode(echopin, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(115200);
}
```

```
void loop(){
  digitalWrite(trigerpin, LOW);
  delay(2);
  digitalWrite(trigerpin, HIGH);
  delay(10);
  digitalWrite(trigerpin, LOW);
  duration=pulseIn(echopin,HIGH);
  a=duration/2*0.034;
  Serial.print("distance (in cm)=");
  Serial.println(a);
  Blynk.virtualWrite(V1,a);
  Blynk.virtualWrite(V3, 4);
  if(a>10&&a<30){
    Blynk.virtualWrite(V3, 3);
    Serial.println("level :3");
  }else if(a>30&&a<50){
    Blynk.virtualWrite(V3, 2);
    Serial.println("level :2");
  }else if(a>50&&a<70){
    Blynk.virtualWrite(V3, 1);
    Serial.println("level :1");
  }else if(a>70){
    Blynk.virtualWrite(V3, 0);
    Serial.println("level :0");
  }

if (a<30){
  digitalWrite(buzzer, HIGH);
  delay(1000);
}
```

```
else{
  digitalWrite(buzzer,LOW);
  delay(500);
}
}
```

Working Principle :

1. Data Collection and IoT Devices:

Set up IoT devices (sensors, flow meters) to monitor water parameters such as water level, quality, and flow rate.

2. Data Processing and Analysis:

Develop software for data processing, analytics, and decision-making algorithms on the server.

Use Python, Node.js, or other server-side languages and libraries to handle data efficiently.

3. Web Development:

Create a web-based platform to monitor and manage the Smart Water Management system.

4. User Interfaces:

Design user-friendly web interfaces for different stakeholders, including administrators, maintenance personnel, and end-users.

Implement responsive web design to ensure usability on various devices.

5. Data Visualization:

Use web-based data visualization libraries such as D3.js, Chart.js, or Plotly to display real-time and historical data through interactive charts and graphs.

Display water quality, consumption trends, and equipment health.

6. Alerts and Notifications:

Set up alert mechanisms to notify stakeholders via web notifications or email about critical water parameters, equipment issues, or consumption anomalies.

7. Remote Monitoring:

Implement remote monitoring of the water management system through a web dashboard, enabling stakeholders to check system status and make adjustments as needed.

8. User Authentication and Security:

Implement secure user authentication mechanisms to control access to system data. Use HTTPS for secure data transmission and ensure that user data remains private.

9. Database Management:

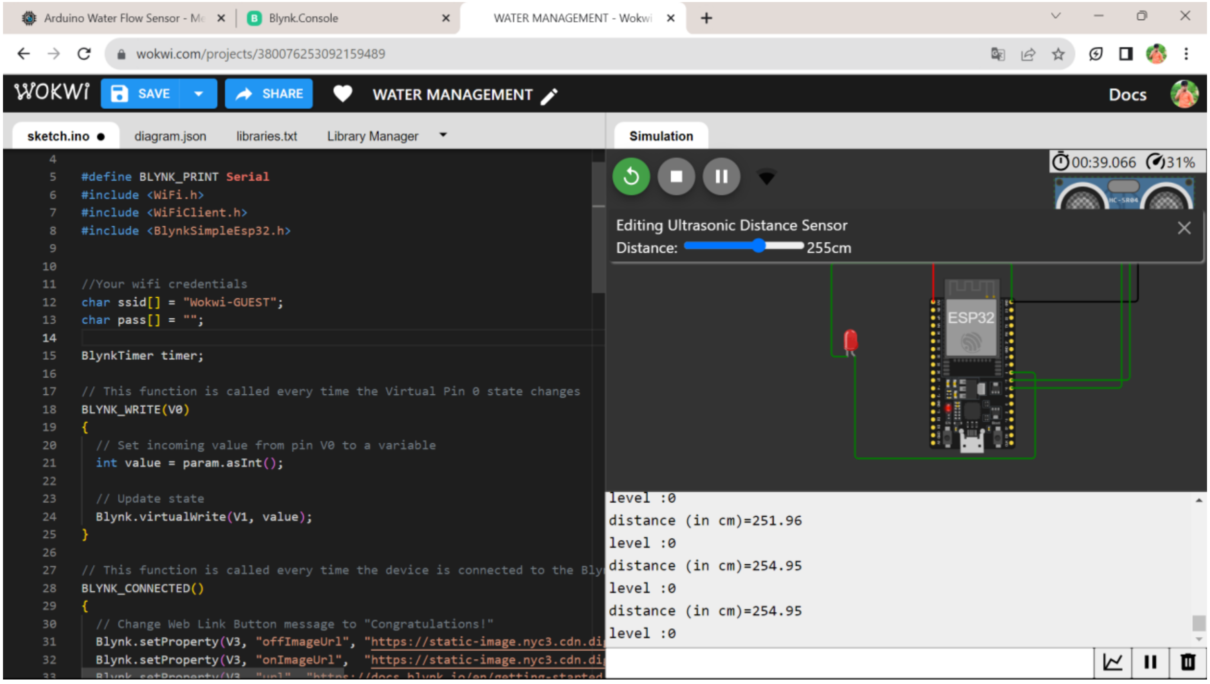
Set up a database system (e.g., MySQL, MongoDB) to store historical data for analysis and reporting.

10. Mobile App Integration :

Develop mobile applications for Android and iOS platforms to enable on-the-go monitoring and control of the Smart Water Management system.

Utilize web technologies, like React Native or Flutter, to build cross-platform mobile apps.

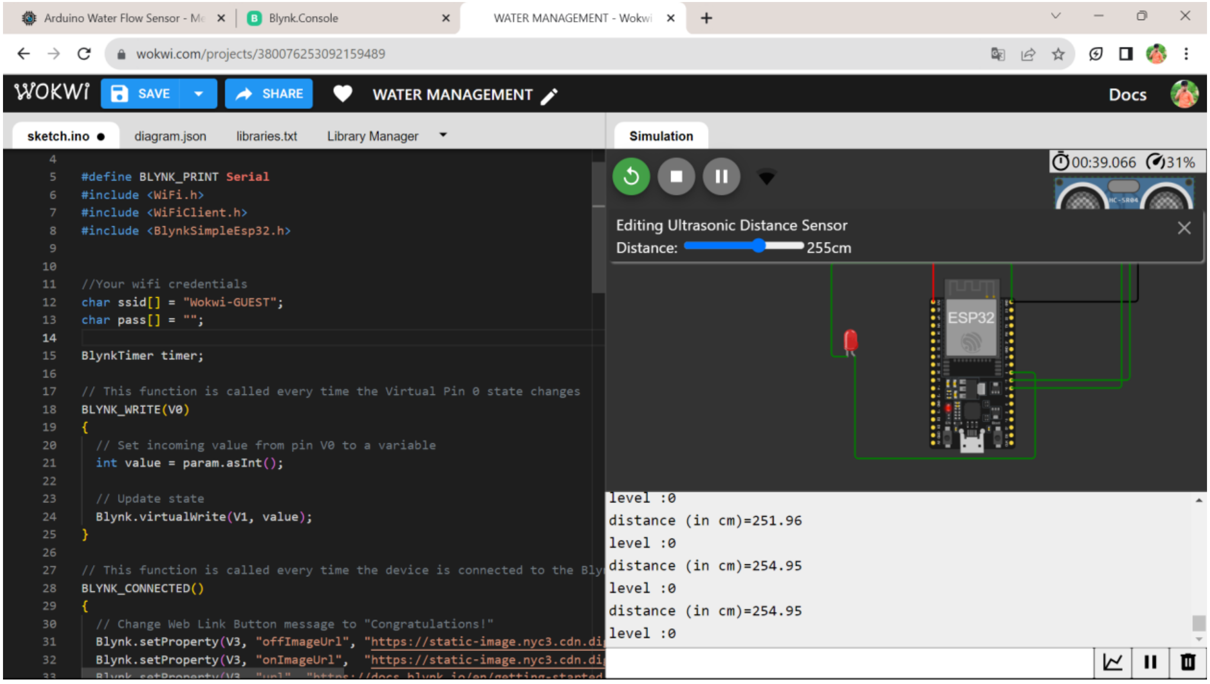
Simulation Using ESP 32



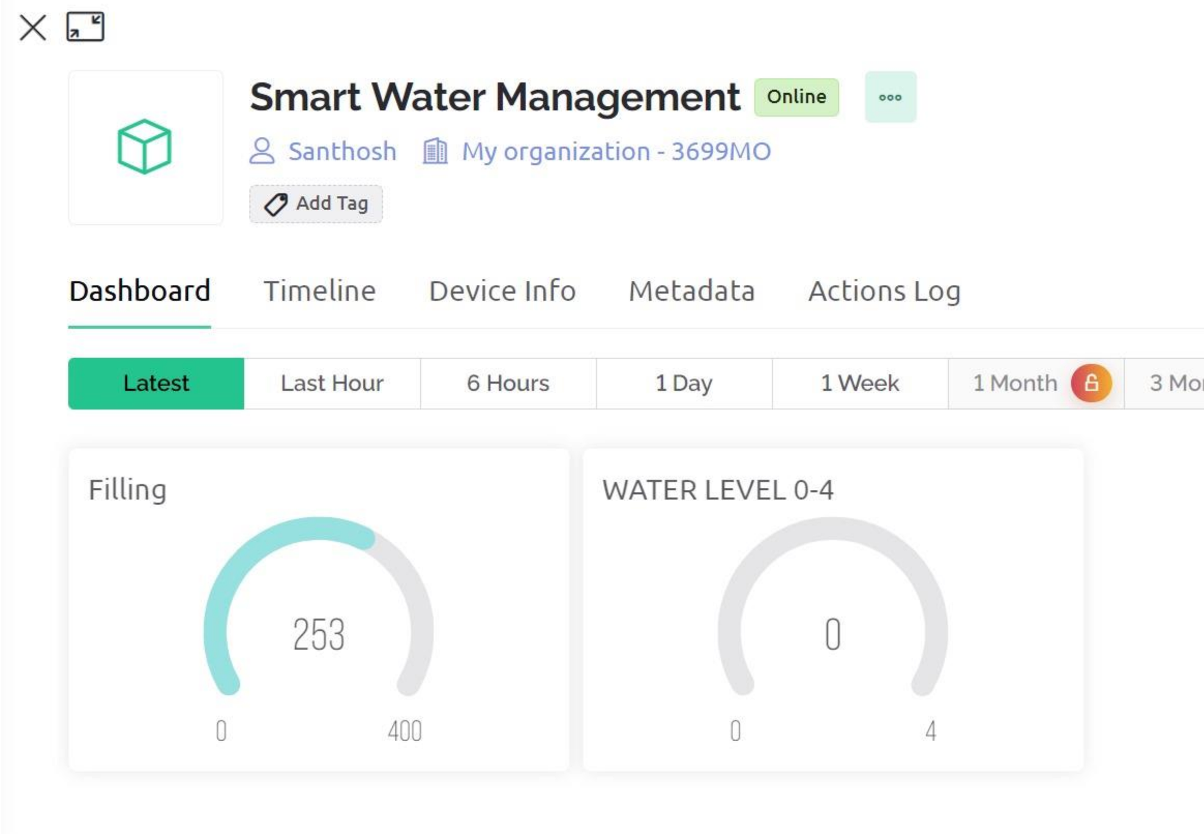
The Blue LED indicated the functioning of the Fountain and the Red LED indicates the functioning of refill motor which fills the reservoir.

We have used Blynk app to get

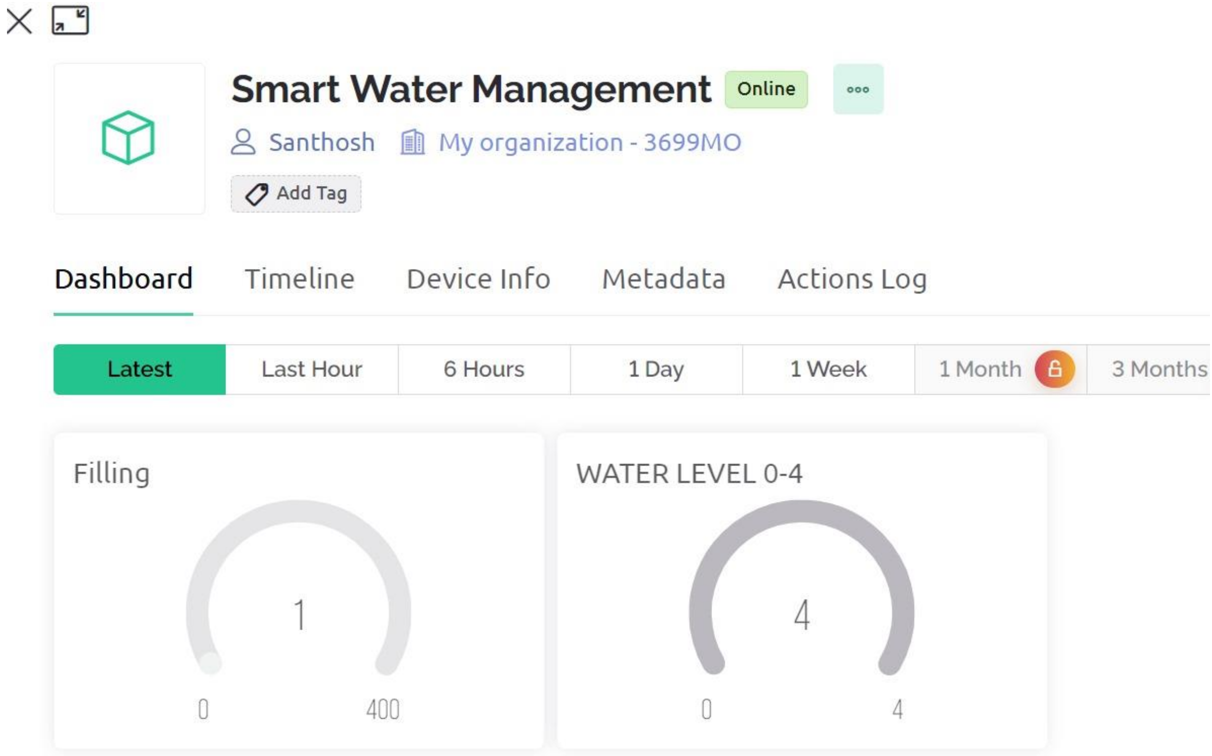
Using C/C++ & BLYNK



Arduino Water Flow Sensor - M
Blynk.Console
WATER MANAGEMENT - Wokwi
wokwi.com/projects/380076253092159489
WOKWI
SAVE
SHARE
WATER MANAGEMENT
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:39.066
31%
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
//Your wifi credentials
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
BlynkTimer timer;
// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
// Set incoming value from pin V0 to a variable
int value = param.asInt();
// Update state
Blynk.virtualWrite(V1, value);
}
// This function is called every time the device is connected to the Bly
BLYNK_CONNECTED()
{
// Change Web Link Button message to "Congratulations!"
Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.di
Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.di
Editing Ultrasonic Distance Sensor
Distance:
255cm
ESP32
level :0
distance (in cm)=251.96
level :0
distance (in cm)=254.95
level :0
distance (in cm)=254.95
level :0

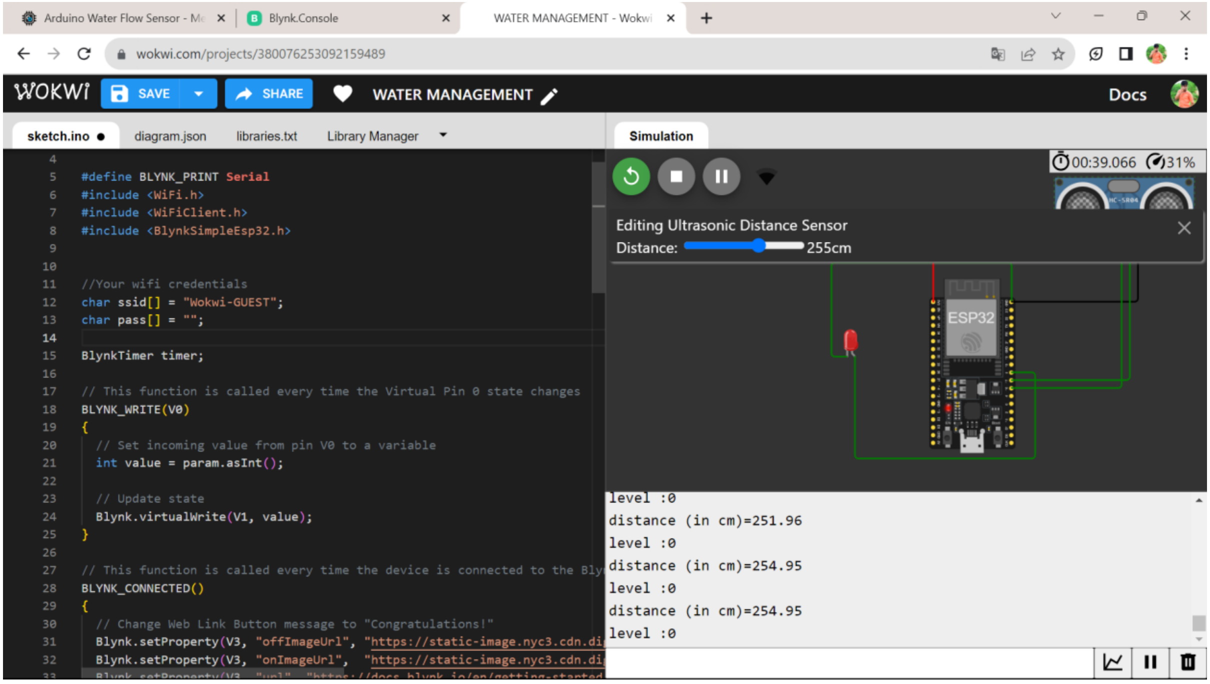


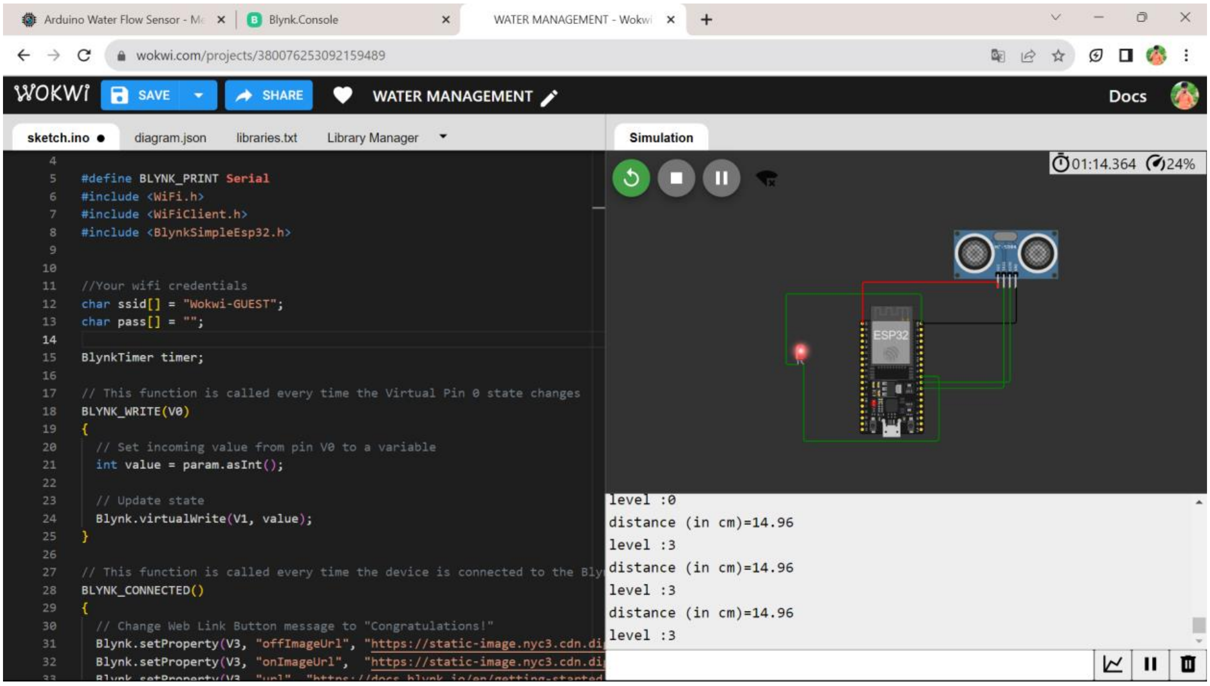
Smart Water Management
Online
Santhosh
My organization - 3699MO
Add Tag
Dashboard
Timeline
Device Info
Metadata
Actions Log
Latest
Last Hour
6 Hours
1 Day
1 Week
1 Month
Filling
253
0
400
WATER LEVEL 0-4
0
0
4



The Blue LED indicated the functioning of the Fountain and the Red LED indicates the functioning of refill motor which fills the reservoir.

We have used Blynk app to get live





The Above processes can be monitored and controlled using BLYNK

Advantages:

1. Resource Conservation: Smart water management can lead to more efficient use of water resources, reducing wastage and promoting sustainability.

2. Real-Time Monitoring: Smart systems allow real-time monitoring of water infrastructure, enabling quick detection of leaks or issues.

3. Data-Driven Decision Making: Data analytics can help water utilities make informed decisions, optimize distribution, and plan for future needs.

4. Improved Customer Engagement: Consumers can access data about their water usage, encouraging water conservation.

5. Environmental Benefits: Reduced water wastage positively impacts ecosystems and energy consumption.

6. Cost Savings: Lower operational costs, reduced water loss, and improved infrastructure management can result in financial savings.

Disadvantages:

1. Costly Implementation: Setting up smart water systems can be expensive, including the installation of sensors and data infrastructure.

2. Vulnerability to Cyberattacks: Smart systems can be targets for cyberattacks, potentially compromising water supply and data security.

3. Technical Challenges: Smart water systems require ongoing maintenance and expertise to function effectively.

4. Privacy Concerns: Collecting and sharing data on water usage raises privacy issues, and consumers may be wary of data misuse.

5. Inequity: The benefits of smart water management may not reach all communities, potentially exacerbating existing disparities in water access.

6. Reliance on Technology: Over-reliance on technology can lead to issues when systems fail or during power outages.

Conclusion

Smart water management is the use of digital technologies to improve the efficiency and sustainability of water management. It involves the use of sensors, data analytics, and artificial intelligence to monitor and control water systems in real time. smart water management can also have a broader impact on the environment and society. Smart water management is a vital tool for addressing the global water crisis. By investing in smart water management systems, we can improve the efficiency and sustainability of water management and ensure that everyone has access to clean water.