WATER MANAGEMENT FOR SMART CITIES

Batch 5 | Oct 30, 2024

Submitted To:- L&T EduTech

Submitted By:- Danishbir Sidhu(2310996771)
Ashutosh Sharma(2310996773)
Karamjit Chechi(2310996774)
Brahmveer Singh(2310996775)

Overview

Aim

Design and develop a Water Management System prototype for smart cities that:

- Tracks water usage in residential and commercial areas.
- Detects leaks and ensures efficient water distribution in real-time.
- Provides data analytics and user access through a cloud platform.

Problem Statement

The increasing population and urbanization have intensified the demand for efficient water management. Current water distribution systems often face issues such as wastage through leaks, inefficient usage, and lack of real-time data monitoring. This project aims to address these problems by building a smart water management solution that allows city authorities and consumers to monitor and manage water usage effectively, ensuring the prevention of leaks and optimized distribution.

Scope of Solution

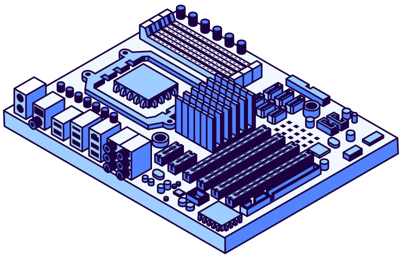
The system will be able to:

- Monitor water usage across various regions (residential/commercial) in real-time.
- Detect and flag anomalies like excessive usage or leaks.
- Manage distribution to avoid wastage and ensure efficient use of available resources.
- Provide cloud-based access for users and city authorities for monitoring and decision-making.
- Integrate with IoT sensors and cloud services for scalable and efficient implementation.

“Required Components To Develop Solution.”

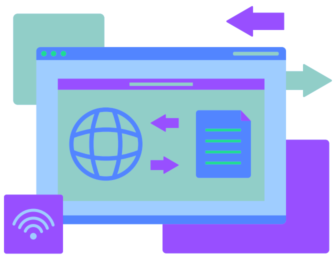
1. Hardware Components:

- **Microcontroller:** Arduino Uno or ESP32
- **Water flow sensors:** To measure water usage
- **Water level sensors:** For tanks and pipelines to monitor water levels
- **Leak detection sensors**
- **Wi-Fi module:** ESP8266 for Arduino or built-in Wi-Fi for ESP32
- **Solenoid valves:** To control water flow
- **Power supply:** For the microcontroller and sensors



2. Software Components:

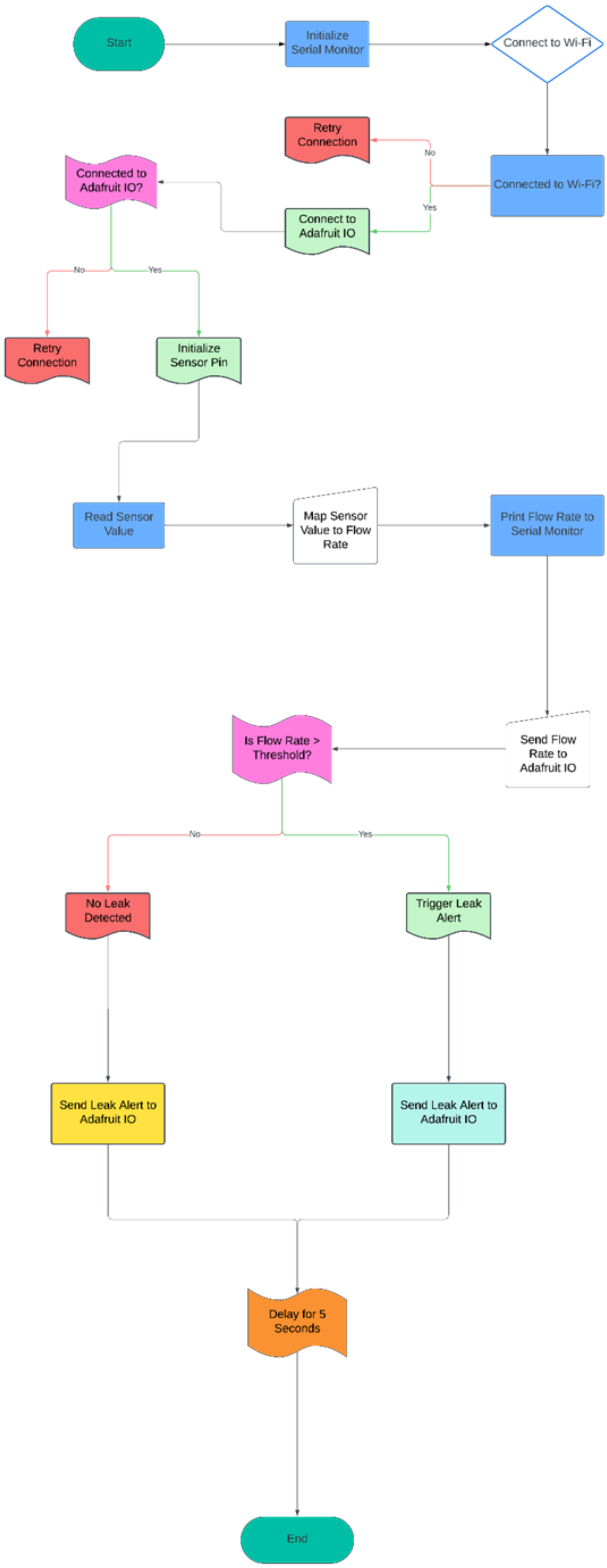
- **Arduino IDE:** For programming the microcontroller
- **Cloud Service:** AWS IoT, Google Cloud, or Azure IoT
- **Data visualization tools:** Power BI, Grafana, or Google Data Studio
- **API & Connectivity:** MQTT, HTTP, or WebSocket for cloud communication



3. Cloud Environment:

- **Cloud Storage:** To store historical water usage data
- **Cloud Processing:** For real-time monitoring and analytics
- **User Interface:** Web or mobile-based application for access to water consumption data and system alerts.

Flowchart

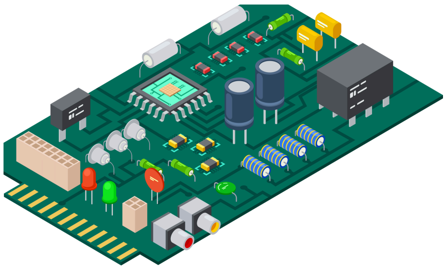


Start
Initialize Serial Monitor
Connect to Wi-Fi
Retry Connection
No
Connected to Wi-Fi?
Yes
Connected to Adafruit IO?
Connect to Adafruit IO
No
Yes
Retry Connection
Initialize Sensor Pin
Read Sensor Value
Map Sensor Value to Flow Rate
Print Flow Rate to Serial Monitor
Is Flow Rate > Threshold?
Send Flow Rate to Adafruit IO
No
Yes
No Leak Detected
Trigger Leak Alert
Send Leak Alert to Adafruit IO
Send Leak Alert to Adafruit IO
Delay for 5 Seconds
End

Hardware Kit

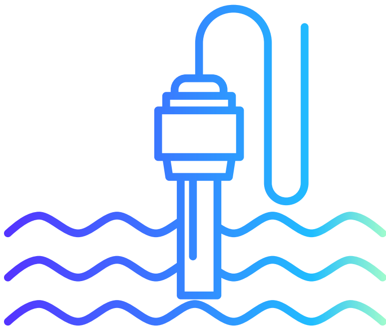
1.Microcontroller Board

Arduino Uno: Using the Arduino Uno as we are focusing on local monitoring and basic functionalities.



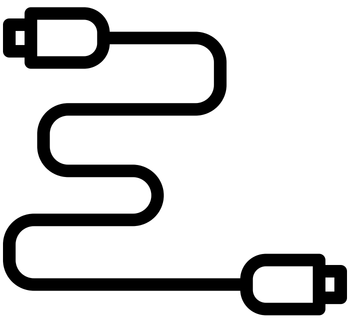
2.Float Sensor

Water Flow Sensor: This is ideal for measuring the actual water flow.



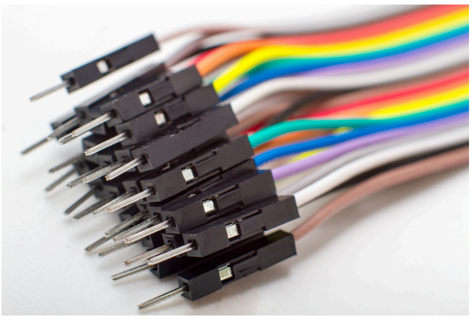
3.Power Supply

USB Power Supply: For powering the Arduino Uno or ESP board. You can use a power bank or a USB wall adapter.



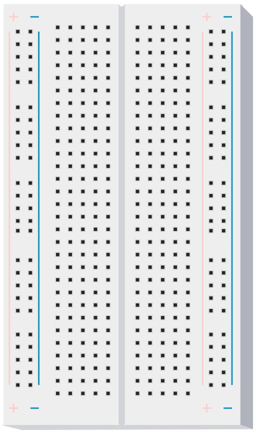
4.Connecting Wires

Jumper Wires: For connecting components on a breadboard.



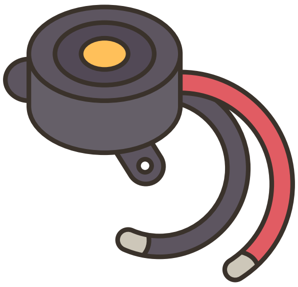
5.Breadboard

A **breadboard** is useful for making connections without soldering.



6.Buzzer

A buzzer can be used for alerting if a leak is detected.



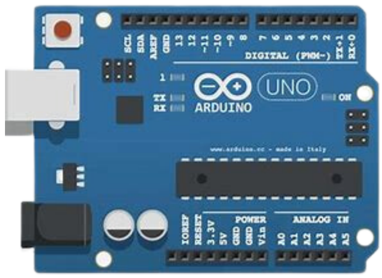
7.Laptop/PC

For programming the Arduino/ESP board and monitoring data.

“Reason For Required Components.”

1.Arduino Uno

Versatile and widely available. Alternatives like Raspberry Pi would be overkill for this simple prototype. Has sufficient number of slots for a small prototype.



2.Float Sensor

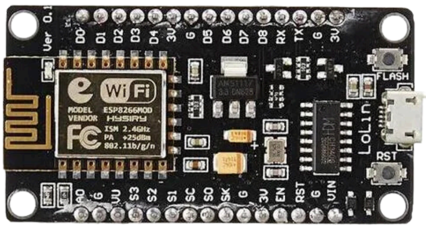
Simple and reliable for detecting water presence. Alternatives like ultrasonic sensors could be more precise but are more complex and expensive.

3.Buzzer

Simple and effective for local alerts. Alternatives like LEDs would be less noticeable in a water management scenario.

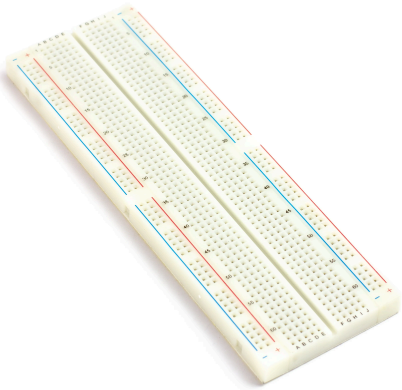
4.ESP8266 WiFi Module

Inexpensive and widely supported for IoT applications. It's more suitable for this project than alternatives like Ethernet shields due to its wireless capability and lower power consumption.



5.Breadboard

They provide a flexible and temporary connection method, ideal for prototyping and testing



Code

```
// Define the pins for the float sensor and the buzzer
int floatSensorPin = 2;
int buzzerPin = 3;

void setup() {
  // Set the float sensor pin as input
  pinMode(floatSensorPin, INPUT);

  // Set the buzzer pin as output
  pinMode(buzzerPin, OUTPUT);

  // Start the serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the state of the float sensor
  int floatSensorState = digitalRead(floatSensorPin);

  // Check if the float sensor is triggered
  if (floatSensorState == HIGH) {
    // Turn on the buzzer
    digitalWrite(buzzerPin, HIGH);
```

```
    Serial.println("Float sensor triggered!");

    delay(3000);

  } else {

    // Turn off the buzzer

    digitalWrite(buzzerPin, LOW);

    Serial.println("Float sensor not triggered.");

  }

}
```

Conclusion

In this project, we successfully developed a prototype of a **Water Management System** using an Arduino, a float sensor, and a buzzer. The system is designed to monitor the water level in a tank and sound an alert when the water level decreases below a critical threshold. By doing this, we can ensure timely interventions, preventing water shortages and promoting efficient water usage in smart city applications.

The project demonstrated how to integrate basic hardware components and Arduino programming to create a real-time monitoring system. The float sensor was used to detect water levels, and the buzzer provided an audible alert in case of low water. This prototype offers a scalable solution that can be further expanded by integrating additional features such as real-time data logging, cloud access for remote monitoring, and automated water distribution.

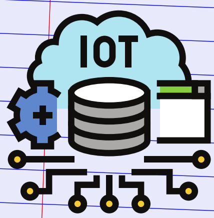
This system highlights the potential for simple, cost-effective solutions to address water management issues in modern urban environments, where resource efficiency is crucial. The knowledge gained from this project can be applied to larger, more complex systems that aim to optimize water distribution and prevent resource wastage in smart cities.

GitHub Repository Link

https://github.com/AshutoshSharma026/AIIOT-Batch-5/tree/c8ea229f1eaec28532262053c569386150c35be4

Thank

You



CHITKARA
UNIVERSITY