Smart Water Fountains
IOT_PHASE-05

Project Objectives:

1. Create Smart Water Fountains that can dispense clean and safe drinking water.
2. Implement a monitoring system to track water usage and quality in real-time.
3. Minimize water wastage by optimizing fountain operation.
4. Ensure continuous availability of water to the public.
5. Implement user-friendly features for easy access and use.

Understanding the Problem:

The problem statement revolves around designing and implementing Smart Water Fountains. These fountains aim to address several key issues:

Clean and Safe Drinking Water:

The primary purpose of these fountains is to provide access to clean and safe drinking water to the public. Ensuring water quality is paramount to prevent health risks.

Water Conservation:

In addition to providing clean water, these fountains must be equipped with technology to minimize water wastage. This involves tracking water usage and implementing efficient dispensing mechanisms.

Continuous Availability:

Smart Water Fountains should be available to the public at all times. This requires a monitoring system to detect issues such as low water levels or malfunctions.

Proposed Solution:

To solve the problem, we propose the following solution:

1. Smart Dispensing Mechanism:

Implement a sensor-based system that dispenses water when a user approaches.

Utilize ultraviolet (UV) or other suitable technology to ensure water quality.

Incorporate a touchless user interface for easy and hygienic access.

2. Real-time Monitoring:

Install water quality sensors to monitor water quality continuously. Use flow sensors to track water usage in real-time. Implement a central monitoring system that collects data from all fountains.

3. Smart Control System:

Optimize fountain operation based on real-time data to reduce wastage. Send alerts to maintenance teams when issues are detected.

4. Water Source and Storage:

Connect the fountains to a clean and reliable water source (e.g., municipal water supply). Include water storage tanks with automatic refilling mechanisms to ensure uninterrupted service during water supply interruptions.

5. User-friendly Features:

Design a user interface that is intuitive and accessible to people of all ages and abilities. Provide information about water quality and source for user confidence. Ensure the fountains are ADA-compliant for accessibility.

6. Maintenance and Support:

Establish a maintenance schedule for regular checks and cleaning. Provide user support through a dedicated helpline or mobile app for reporting issues.

7. Sustainability:

Implement energy-efficient components, such as LED lighting and low-power sensors. Promote the use of reusable water bottles to reduce plastic waste.

Project Implementation Steps:

Requirements Gathering: Gather detailed requirements from stakeholders, including local authorities, public health officials, and potential users.

Design: Develop a detailed design plan that includes the fountain's physical structure, the placement of sensors, water source connections, and user interface.

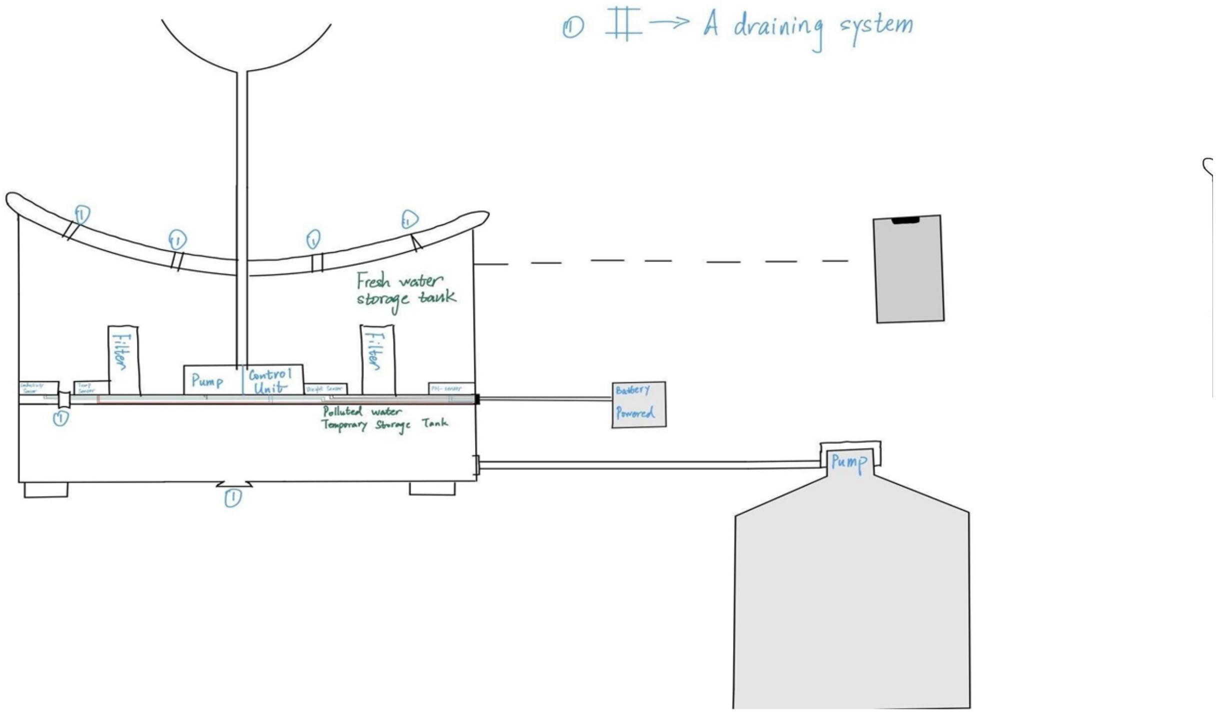
Prototyping: Build a prototype fountain to test the smart dispensing mechanism and monitoring system.

Full-scale Implementation: Deploy Smart Water Fountains at selected locations.

Testing and Optimization: Continuously monitor and test the fountains in real-world scenarios. Optimize the control algorithms for efficiency and user experience.

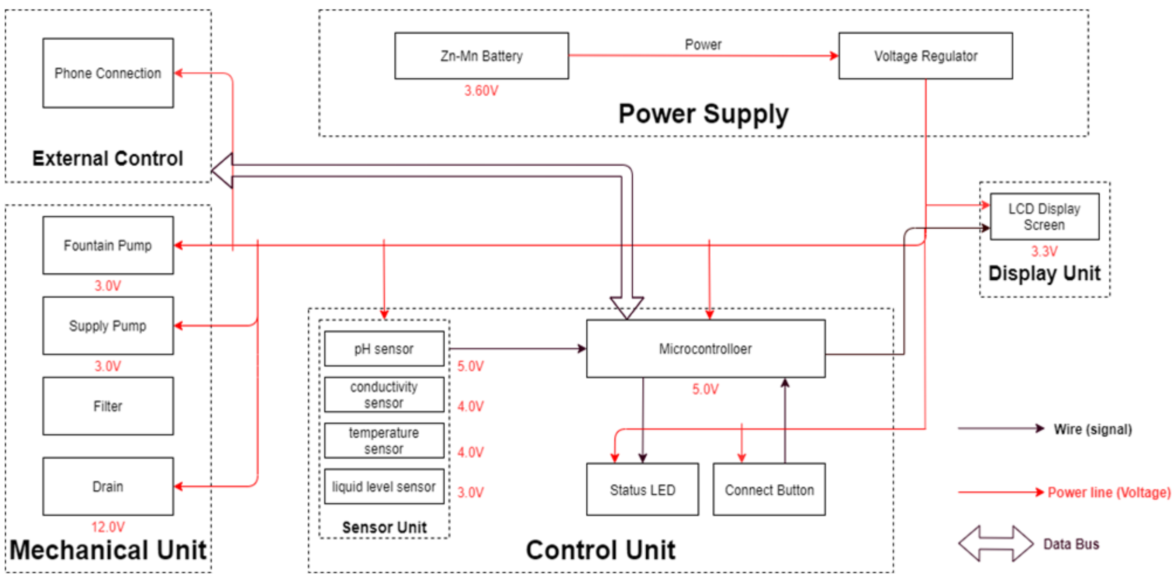
PhysicPhysical Design:

A pictorial representation of the project that puts the solution in context. Not necessarily restricted to your design. Include other external systems relevant to your project (e.g. if your solution connects to a phone via Bluetooth, draw a dotted line between your device and the phone).



Design:

The block diagram below is a general design of our solution. We divide our design into four modules, including Power Supply, Control Unit, External Control, and Mechanical Unit. Details of each unit is presented in the diagram and described in the next section.



Creating a complete Python script for IoT sensors to send real-time water fountain status data to an IoT platform requires knowledge of the specific IoT platform you are using, the sensor hardware, and the communication protocol you want to implement (e.g., MQTT or HTTP).

Below, I'll provide a general Python script template to help you get started. You'll need to adapt this script to your specific hardware and IoT platform.

I'll use the Paho MQTT library to send data to an MQTT broker. Please adjust the script according to your actual setup.

Python code:

import time

import random # For generating dummy data (you should replace this with real sensor

data) import paho.mqtt.client as mqtt

 # MQTT Broker settings (replace with your IoT platform

details) MQTT_BROKER_HOST =

"your-broker-hostname" MQTT_BROKER_PORT =

1883

MQTT_TOPIC =

"water-fountain-status"

MQTT_USERNAME =

"your-username"

MQTT_PASSWORD =

"your-password"

 # Initialize the MQTT client client = mqtt.Client()

 # Connect to the MQTT broker client.username_pw_set(MQTT_USERNAME, MQTT_PASSWORD) client.connect(MQTT_BROKER_HOST, MQTT_BROKER_PORT)# Function to simulate sensor data (replace with real sensor data acquisition) def read_sensor_data():

 # Replace this with actual code to read sensor data

flow_rate = random.uniform(0.1, 10.0) # Example: Random flow rate between 0.1 and 10.0 liters per minute

pressure = random.uniform(20.0, 100.0) # Example: Random pressure between 20.0 and
100.0 PSI

return {"flow_rate": flow_rate, "pressure": pressure}

 # Main loop to publish

sensor data while True:

sensor_data = read_sensor_data()

client.publish(MQTT_TOPIC,

payload=str(sensor_data)) print(f"Published

data: {sensor_data}")

time.sleep(5) # Adjust the frequency of data transmission as needed

Here's a brief explanation of the code:

Import necessary libraries, including paho.mqtt.client for MQTT communication.

Set your MQTT broker settings (host, port, topic, username, and password). Replace these values with your IoT platform details.

Initialise the MQTT client and connect to the broker. Create a function (read_sensor_data) that simulates sensor data. In practice, replace this with actual code to read data from your flow rate and pressure sensors.

Sensor Unit:

This block contains the four sensors. The data acquired from the sensors will be transmitted to the control unit. Control unit will then have some logic designed to send corresponding signals to control other blocks of the water fountain. At the same time, the display screen on the water fountain will display the readings along with the determined water quality level and remaining water quantity. For the PH-value sensor, temperature sensor and conductivity sensor, values will be retrieved and calculated to determine the overall water quality level. When poor water quality is determined, the water replacement procedures will take place. The weight sensor readings will be used to determine the amount of fresh water left in the water tank.

Temperature Sensor:

A water-proof temperature sensor is going to be used. Part number from sparkfun is: DS18B20 [6]. This temperature sensor is compatible with a relatively wide range of power supply from 3.0V to 5.5V. The measured temperature ranges from -55 to +125 celsius degrees. Between -10 to + 85 degrees, the accuracy is up to +-0.5 degrees. This sensor can fulfill all requirements needed for this project.

PH-sensor:

PH value is a valued indicator of water quality. This PH-sensor[7] works with 5V voltage, which is also compatible with the temperature sensor. It can 6measure the PH value from 0 to 14 with an accuracy of +- 0.1 at the temperature of 25 degrees.

Conductivity sensor:

Conductivity sensor is also part of the water quality assessment. The input voltage is from 3.0 to 5.0V. The error is small, +-5%F.S. The measurement value ranges from 0 to 20 ms/cm which is enough for water quality monitoring. [8]

Liquid Level Sensor:

This sensor [9] is responsible for reflecting how much freshwater is left in the water tank. When the water level is low, fresh water will be pumped to the water tank to ensure the water fountain keeps running with freshwater. This sensor is 0.5 Watts. For water level from 0 to 9 inches, the corresponding sensor outputs readings from 0 to 1.6. From that, the quantity of freshwater left can be determined.

i. Continue building the project by developing the water fountain status platform.
ii. Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water fountain status.
iii. Design the platform to receive and display real-time water fountain data, including water flow rate and malfunction alerts.

HTML: Create the structure of your web page.

```
<!DOCTYPE html>
<html>
<head>
   <title>Smart Water Fountain Dashboard</title>
   <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
   <header>
      <h1>Water Fountain Status</h1>
   </header>
   <div class="status">
      <h2>Water Flow Rate</h2>
      <div id="flowRate">Loading...</div>
   </div>
   <div class="status">
      <h2>Malfunction Alerts</h2>
      <div id="malfunctionAlerts">Loading...</div>
   </div>

   <script src="script.js"></script>
</body>
</html>
```

CSS:

```
body {
```

```css
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: #f0f0f0;
}

header {
    background-color: #333;
    color: #fff;
    padding: 10px;
}

.status {
    display: inline-block;
    margin: 20px;
    padding: 20px;
    background-color: #fff;
    border: 1px solid #ddd;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
}
h2 {
    color: #333;
}
#flowRate {
    font-size: 24px;
}
#malfunctionAlerts {
    font-size: 18px;
}
```

JAVASCRIPT:

```
function updateData() {
    // Simulate fetching data from your water fountain sensors or API
    const flowRateData = Math.random() * 10; // Replace with actual data
    const malfunctionAlerts = Math.floor(Math.random() * 2); // Replace with actual data


    // Update the HTML elements with the retrieved data
    document.getElementById("flowRate").innerText = flowRateData.toFixed(2) + " GPM"; /
Display with 2 decimal places
    document.getElementById("malfunctionAlerts").innerText = malfunctionAlerts === 1 ?
"Yes" : "No"; // Display Yes/No
}

// Fetch and update data every 5 seconds (adjust as needed)
setInterval(updateData, 5000);

// Initial data update
updateData();
```

This project focuses on the development of a smart water fountain status platform using web development technologies, including HTML, CSS, and JavaScript. The platform is designed to display real-time water fountain data, such as water flow rate and malfunction alerts. The web-based dashboard provides a user-friendly interface for monitoring the status of the water fountain. The HTML structure defines the layout, while CSS is used for styling the elements. JavaScript fetches and updates the real-time data, simulating data retrieval from sensors or an API. This example serves as a foundation for creating a comprehensive smart water fountain system, which can be expanded to include features like data storage, authentication, and integration with actual sensor data sources.

Risk Analysis:

Control Unit Block:

One of the most challenging points in this project is the precise control of the control unit between different blocks. To react accurately and promptly based on the results from the sensors is the key. The control unit needs to accommodate the mechanical and the electrical part so that the pumps, draining system can work collaboratively smoothly. From acquiring the data from sensors, analyzing the data, communicating and displaying the data to users, and then sending signals to activate the corresponding actions(drain or add fresh water), these are all to be performed by the control unit. Thus, it is the block that brings the greatest risk.

We will divide all the overall control unit functions into three parts: data retrieving, data manipulation, data delivering. Data retrieving is the logic used to read data from all sensors. Necessary algorithm is to be written to ensure successful and accurate data acquisition.

Data manipulation is the process of calculating the water quality levels, and the formula to integrate all the data to produce a credible result. The data delivering is used to connect the control unit to the screen, displaying the necessary information as described above. This part will also be responsible for building the connection between the water fountain and the users' phones through WIFI.

Mechanical Unit Block:

This is very challenging and extremely important. As most of the components will be exposed to water. Sensors, pumps, filters, draining system motors are all to be placed in the water tank. This means that we need to ensure no water can leak into the electrical-related mechanical parts. This puts pressure on the design and also the implementation. In addition, the motor-controlled valves used to drain the polluted water need to be firm when closed.

Otherwise the fresh water will be leaking to the polluted water storage and the water consumption will be uncontrollable.

To achieve those points, we will make sure the designs are carefully implemented. The actual building process for the container should be proved before placing the electronic parts in.

Ethics and Safety

Mechanical Unit Block

I-1 of IEEE Code of Ethics:

Quoted from IEEE Code of Ethics[11]: "To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment.”

We will carefully choose the materials used to build the container. Non-toxic are sure to be used. We will prefer using reusable materials. In addition to that, the users can choose to buy reusable bottles of water for the freshwater supply for the water fountain. Those universal water bottles are safe and reusable. [12] A special connector will be designed and the universal connection is to be used. After the water in the bottle is used up, this reusable bottle can be recycled and reused. This is the most environmentally-friendly solution and complies with the IEEE Code of Ethics #I-1. It not only improves the practicality, convenience, and reduces the future cost when using the water fountain.

II of IEEE Code Of Ethics:

Quoted from [11]: “II. To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.”

As mentioned in the 3.2, the mechanical unit involves electronic components that are physically placed in the water tank. The consequence can be serious if the leakproofness is not performed properly. To maintain a safe, convenient using experience, we will be responsible for testing and ensuring all containers meet the demand. These actions must be taken to ensure the safety of using the water fountain and protect the others.

I-6 of IEEE Code Of Ethics:

Quoted from [11]: “to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.”

All team members involved in the development of the water fountain have completed “Laboratory Safety training” and have gained required and necessary knowledge in dealing with emergency situations. In case of accidents, proper reaction will be made to ensure the safety of people and property to the largest extent.