SMART WATER FOUNTAINS

NAME: Sneha Snigdha Mallick D

REG NO: 723721104103

PHASE-1 DOCUMENT SUBMISSION

PROJECT DEFINITION:

The project aims to enhance public water fountains by implementing IoT sensors to control water flow and detect malfunctions. The primary objective is to provide real-time information about water fountain status to residents through a public platform. This project includes defining objectives, designing the IoT sensor system, developing the water fountain status platform, and integrating them using IoT technology and Python.

PROJECT OBJECTIVES:

1. Real-time Water Fountain Monitoring: Implement IoT sensors to continuously monitor water fountain status in real-time.

2. Efficient Water Usage: Optimize water usage by controlling water flow based on demand.

3. Malfunction Detection: Detect and report malfunctions such as leaks, low water levels, or pump failures.

4. Resident Awareness: Provide residents with easy access to information about the nearest available water fountain.

DESIGN THINKING:

1. IoT Sensor Design:

- Selection of Sensors: Choose appropriate IoT sensors, including flow rate sensors and pressure sensors, to collect relevant data.

- Sensor Placement: Determine the optimal locations for sensor deployment on each water fountain.

- Connectivity: Ensure sensors are connected to a central data hub for data transmission.

2. Real-Time Water Fountain Status Platform:

- Mobile App Interface: Design a user-friendly mobile app interface to display real-time information.

Features:

- Map View: Show the locations of public water fountains on a map.

- Fountain Status: Indicate whether a fountain is operational, under maintenance, or offline.

- Water Flow Control: Allow users to request increased water flow when needed.

- Malfunction Alerts: Notify users and maintenance teams of any detected malfunctions.

3. Integration Approach:

- IoT Communication: Determine the communication protocols (e.g., MQTT) for transmitting sensor data to the platform.

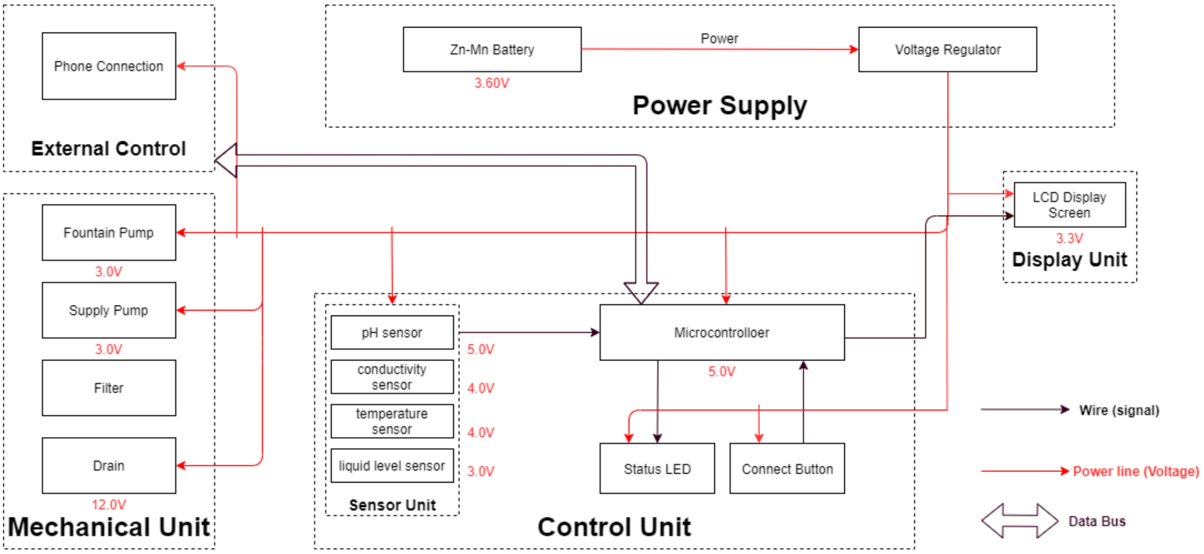
- Data Processing: Develop data processing algorithms to analyze sensor data for status updates and malfunction detection.

- Mobile App Integration: Ensure the mobile app can receive real-time updates from the IoT sensors.

DESIGN PROCESSING:

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.



Block Diagram of Smart Water Fountain

1. Power Supply Unit

- The Zn-Mn battery must be able to continuously support the functioning of the circuit, display unit, and the mechanical unit.

Requirement: Commercial batteries will be used to maintain a continuous 3.60V power supply for at least 24 hours. If the chosen battery is not powerful enough, 120V power outlets will be considered.

- The integrated circuit will regulate the power supply for each module to maintain their functionality. This chip must be able to handle the maximum voltage supplied by the battery (3.60V ± 0.5V) while ensuring the voltage at each module does not exceed their limit.

2. Mechanical Unit

The fountain pump [14] must maintain a continuous water supply through the fountain mechanism. The pump must work 24 hours a day, 7 days a week unless the user manually turns off the power supply

.

3. Supply Pump

The supply pump must function when a low water level alert is raised. While no water supply is requested, the pump must prevent water flow between the main supply and the fountain.

-The filter must maintain the water quality through controlling the pH value and conductivity of the water.

-The drain [13] must be able to hold and release water in the fountain. When water in the fountain should be replaced, the faucet should automatically drain the fountain once instruction is received from the integrated circuit.

4. Control Unit

-When the weight sensor reports a weight less than the minimum weight setting, the control unit will send an alert signal to the user and then control the water supply unit to refill the water fountain with a certain amount of water.

- Computes the water quality with data transferred from the three sensors in the water quality module and sends the result in terms of “Good”, “Average” or “Bad” to the user.

- If the water quality is “Bad”, the control unit will control the drain module to drain the water in the fountain and then control the water supply to refill.

- Water quality result is sent to the user with wireless connection and screen display as described above in the display unit.

MODEL SELECTION:

1. Type of Fountain: Determine if it's a basic fountain with on/off functionality or if you want more advanced features like customizable water patterns, synchronized music, or smart control via an app.

2. Power Source: Consider whether the fountain will be powered by electricity or solar energy, as this may influence your model choice.

3. Size and Design: Choose a model that fits the size and aesthetic of your space.

4. Materials Ensure: the fountain is made of durable, weather-resistant materials if it will be outdoors.

5. Pump Capacity: Check the pump's flow rate and head height to ensure it meets your desired water flow and height.

6. Water Filtration: If necessary, select a model with built-in filtration to keep the water clean.

MODEL TRAINING:

1. Define Objectives: Clearly outline what you want the smart water fountain to do. For example, detect motion to turn on the fountain, adjust water flow, or respond to voice commands.

2. Data Collection: Collect data relevant to your objectives. For motion detection, you might need video or sensor data. For voice commands, you'll need audio data.

3. Data Labeling: Annotate the collected data to indicate the correct actions or responses. For example, label video frames where motion is detected.

4. Feature Engineering: Extract relevant features from the data. This could involve techniques like image processing for video data or speech recognition for audio data.

5. Model Selection: Choose an appropriate machine learning or deep learning model based on your data and objectives. For example, you might use a convolutional neural network (CNN) for image analysis or a natural language processing (NLP) model for voice commands.

6. Data Splitting: Divide your data into training, validation, and test sets to train and evaluate your model's performance.

7. Model Training: Train your chosen model on the training data, adjusting parameters as needed to optimize performance.

8. Testing: Evaluate the model's performance on the test data to ensure it meets your objectives.

MODEL EVALUATION:

1. Accuracy: Measure how often the fountain correctly responds to inputs or triggers. For example, if it's supposed to turn on when it detects motion, calculate the percentage of correct detections.

2. Response Time: Assess how quickly the fountain responds to triggers or commands. Fast response times are often critical for user satisfaction.

3. Energy Efficiency: Evaluate the energy consumption of your smart fountain. Ensure that it doesn't drain power excessively when in standby mode.

4. Robustness: Test the model's performance under different conditions, such as varying light levels (for motion detection) or noisy environments.

5. User Feedback: Collect feedback from users to gauge their satisfaction with the fountain's behavior and responsiveness.

6. Reliability: Measure the model's uptime and check if it encounters frequent failures or errors.

7. Security and Privacy: Assess the model's vulnerability to security threats or data privacy issues, ensuring that user data is protected.

8. Scalability: If you plan to deploy multiple smart fountains, evaluate how easily the model can scale to manage multiple devices simultaneously.

CONCLUSION:

In phase-1,we have established a clear understanding of our goal: Incorporating machine learning and data-driven decision-making can further enhance the functionality and responsiveness of smart water fountains. However, it's essential to carefully plan, evaluate, and maintain these systems to provide a seamless and enjoyable experience for users while maximizing efficiency and sustainability.