# Project Proposal - Introduction to IoT
# **Water Quality Monitoring System**
# Team ID - 18

## Introduction

1) **Explain the problem the project aims to solve or the opportunity it seeks to capitalize on**

Access to clean and safe drinking water is paramount for maintaining our health and well-being, especially within the confines of our homes. However, water pollution remains a pressing concern globally, with contaminants such as industrial waste, sewage, and pollutants infiltrating water sources. This threat extends to the water stored in tanks within our homes, which can become susceptible to contamination due to various factors like improper storage or aging infrastructure.

Traditional methods of water quality monitoring, often reliant on periodic sampling and laboratory analysis, present significant drawbacks when it comes to timely detection of contamination in home water tanks. Moreover, manual monitoring processes are labor-intensive, time-consuming, and lack scalability, rendering them impractical for regular use by homeowners.

In response to these challenges, our project on a smart water quality monitoring system aims to revolutionize the way we ensure the safety of the water in our home tanks. By leveraging real-time monitoring and analysis of water quality parameters such as turbidity, temperature,Total Dissolved Solids (TDS), and pH levels, our system acts as a vigilant guardian against potential contamination threats. It provides homeowners with timely alerts and enables swift action to maintain water quality, whether it involves stopping the water supply, altering the direction of water flow for further purification, or initiating other necessary measures to safeguard the health of household members.By detecting changes in water quality parameters such as turbidity, TDS or pH levels, the system acts as a warning system for potential contamination and thus enables prompt action to be taken.Through this initiative, we strive to empower homeowners with the means to proactively safeguard the quality of their drinking water.

### 2) Describe the project's objectives and deliverables.

The objectives of our project are :

1) **Real Time Monitoring**: Implementing a system that can continuously monitor pH, temperature, TDS, and turbidity levels of drinking water and update and analyze the data.
2) **Data Collection and Analysis**: Collecting data from the sensors and analyzing it to detect any abnormalities in water quality parameters and store them in cloud for further usage.
3) **Alert System**: An integrated alert system that can notify the users in real-time when water quality parameters exceed predefined thresholds.
4) **Control Mechanisms:** Integrate valve actuators to enable automatic control of water supply in response to detected contamination, ensuring the safety of drinking water by preventing and changing the distribution of contaminated water.
5) **User-Friendly Interface:** Develop a user-friendly interface, such as a dynamic user web-page to display real-time water quality data and alerts for the users to easily monitor and manage the system.
6) **Data Logging**: Implementing a data logging system to store all the water quality data for analysis.
7) **Failure Analysis:** Implementing error detection mechanisms to identify anomalies or deviations in sensor readings, communication failures, or hardware malfunctions by using 2 sets of sensors.

Deliverables:

1) **Hardware Setup**:
    a) Installation of pH, temperature, TDS, and turbidity sensors in the water system, along with an ESP32 microcontroller.
    b) Integration of solenoid valve actuators to control water flow based on sensor data and predefined thresholds.
    c) Additional hardware setup to redirect water flow to another container for further purification when water quality parameters exceed threshold levels.
2) **Software Development**:
    a) Development of firmware for the ESP32 microcontroller to collect, process, and analyze data from sensors in real-time.
    b) Development of an alert system utilizing the ESP32's built-in Wi-Fi or Bluetooth capabilities to establish internet connectivity, enabling real-time alerts to be sent when water quality issues are detected.
    c) Implementation of a user-friendly interface, such as a dynamic user web-page for users to remotely monitor water quality data and manage the system.
    d) Incorporation of control algorithms to automate actions such as stopping water flow and redirecting it to another container for purification based on sensor readings.
3) **System Integration**:
    a) Integration of sensor data with the software platform and database for real-time monitoring and analysis.
    b) Establishment of communication protocols between sensors, actuators, and the ESP32 microcontroller to ensure seamless data exchange and control.
    c) Testing and validation of the entire system to ensure reliability, accuracy, and efficiency in monitoring and managing water quality.
4) **Deployment and Support**:
    a) Deployment of the complete system in the user's environment, ensuring proper installation and configuration.
    b) Continuous monitoring and optimization of the system to enhance performance and reliability over time.

# Hardware Requirements

## 1) List the required hardware components such as sensors, actuators, controllers, and other devices.

**The required hardware components for the project include:**

- ESP32 Microcontroller (1x)
- Temperature sensor (2x)
- TDS Sensor (2x)
- Turbidity sensor (2x)
- pH sensor (1x) – If possible because of cost constraints
- Relay Module (2x)
- Flyback Diode (2x)
- Solenoid valve actuator (2x)
- LCD Screen (1x)-if possible
- Breadboard
- Connecting wires

Minimum Sensor, actuator requirements - 2 Temperature sensors, 2 TDS, 2 Turbidity sensors, 2 Relay module, 2 Solenoid valve actuator, 2 Flyback/Relay diode

## 2) Specify the quantities and series of each component that is required.

| Device | Series/Model | Product Link | Github library links | Expected Cost |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Temperature sensor | DS18B20 Waterproof Temperature Sensor | https://www.electronicscomp.com/ds18b20-water-proof-temperature-sensor-probe-india?gad_source=4&gclid=Cj0KCQjwzZmwBhD8ARIsAH4v1gXg-8q3_CRSPpH9mEy02V4TRHNoT8acWC7v0mfGdhKlpo7bMDYdS5saAk5sEALw_wcB | https://www.arduino.cc/reference/en/libraries/ds18b20/ .....or..... https://github.com/RobTillaart/DS18B20_RT (For one sensor per esp32) | 57 * 2 |
| TDS sensor (Total Dissolved Solids) | Gravity: Analog TDS Sensor/ Meter for Arduino .....or...... CQRobot TDS (Total Dissolved Solids) Meter Sensor for Raspberry Pi and Arduino ...or.... Analog TDS Sensor Module Normal Quality | https://www.dfrobot.com/product-1662.html ....or..... https://www.cqrobot.com/index.php?route=product/product&product_id=1122 ...or... https://robu.in/product/analog-tds-sensor-module-normal-quality/ | https://github.com/DFRobot/GravityTDS (For the gravity tds sensor) ...or... https://www.arduino.cc/reference/en/libraries/cqrobottds/ (for cqrobot tds sensor) | 950 * 2 or $8 * 2 or 657 * 2 (Atleast 1 sensor required if not 2) |
| Turbidity sensor | Gravity: Analog Turbidity Sensor for Arduino .....or...... Turbidity Sensor with Module | https://www.dfrobot.com/product-1394.html ......or....... https://www.electronicscomp.com/turbidity-sensor-with-module?gad_source=1&gclid=Cj0KCQjwzZmwBhD8ARIsAH4v1gX | https://github.com/niclabs/GravityTurbidity (For gravity turbidity sensor) ....or..... https://github.com/duyhuynh/Turbidity_Sensor | 800 * 2 or 477 * 2 |

| | | | | |
|---|---|---|---|---|
| | | LUkQnouliG-HOuX9nOJjd5ps3ekSo4DeYMHjtUARrWODicK30Y54aAku3EALw_wcB | | |
| Solenoid valve actuator | 24V DC 1/2 Electric Solenoid Water Air Valve Switch (Normally Closed) | https://robu.in/product/24v-dc-1-2%E2%80%B3-electric-solenoid-water-air-valve-switch-normally-closed/?gad_source=1&gclid=Cj0KCQjwzZmwBhD8ARIsAH4v1gVrQJ1YCrqTSX3Ve7r5qY5_kBUXJOdL5WWuBuFCBTGoOKSPIcPUiw0aAoGKEALw_wcB | Not Required | 227 * 2 |
| Relay Module | 1 Channel 5V Relay Module without Optocoupler | https://www.electronicscomp.com/1-channel-5v-relay-module-without-optocoupler?gad_source=1&gclid=Cj0KCQjwzZmwBhD8ARIsAH4v1gUC1Fr1cr2BMBEMMEU_oQm6QwGJA57SQRstoHvUVa9vbNagC_BqdMEaAva8EALw_wcB | https://github.com/me-no-dev/ESPAsyncWebServer , https://github.com/me-no-dev/AsyncTCP/ , https://randomnerdtutorials.com/esp32-relay-module-ac-web-server/ | 33 * 2 |
| Flyback Diode | | https://www.amazon.com/Ltvystore-100PCS-10Value-1N4001-1N5819-Asso | Not required | Quantity - 2, Price - Negligible |

| | | | | |
|---|---|---|---|---|
| | | rted/dp/B01LW6N0AD/ref=as_li_ss_tl?_encoding=UTF8&psc=1&refRID=7CXJ02F64088F1ZZ18Z3&linkCode=sl1&tag=tinkeranbuild-20&linkId=eba5a67ed9be0d9154a134255275d4fb | | |
| LCD Screen | | https://www.electronicscomp.com/1602-blue-lcd-display-with-i2c-interface?gad_source=1&gclid=Cj0KCQjwzZmwBhD8ARIsAH4v1gUEAn2b3iTlU5N7JYLd3i4_KxgvtKkquXnIontPCx_eWJFoTqSrAOwaAkJEEALw_wcB | https://github.com/johnrickman/LiquidCrystal_I2C , https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/ | 159 * 1 |
| pH sensor (high cost) | Analog PH Sensor Kit for Arduino | https://www.graylogix.in/product/ph-sensor-kit-for-water | https://how2electronics.com/diy-iot-water-ph-meter-using-ph-sensor-esp32/ | 1200 * 1 (If possible) |

## 3) Explain why each component is needed and how it will contribute to the project's goals

1. **ESP32 Microcontroller**:
    - **Explanation**: The microcontroller controls the entire system, processes sensor data, and triggers actions.

- **Working**: It acts as the brain of the system, receiving inputs from sensors, executing programmed logic, and sending commands to actuators based on predefined conditions.
- **Application**: The microcontroller integrates all components of the system, allowing for real-time monitoring and control of water quality parameters. It enables automated responses to sensor data, such as activating the solenoid valve actuator or sending alerts.

2. **Temperature Sensor**:
   - **Explanation**: The temperature sensor monitors water temperature, which can affect water quality and microbial growth.
   - **Working**: It measures the resistance change of a semiconductor element with temperature. As temperature increases, the resistance of the element changes, allowing the sensor to determine the temperature.
   - **Application**: Monitoring temperature is important as it affects various chemical and biological processes in water. Additionally, extreme temperatures can indicate issues with the water supply system.
3. **TDS Sensor**:
   - **Explanation**: It measures the total dissolved solids in water, providing an indication of its mineral content.
   - **Working**: It measures the electrical conductivity of water, which correlates with the TDS level. The sensor applies a voltage across two electrodes and measures the resulting current flow.
   - **Application**: Monitoring TDS levels helps in assessing the overall mineral content of water. Elevated TDS levels may indicate the presence of harmful substances or excessive mineralization.
4. **Turbidity Sensor**:
   - **Explanation**: It detects suspended particles in water, indicating its clarity and purity.
   - **Working**: It works by emitting light into the water and measuring the amount of light scattered. Higher turbidity levels indicate a greater amount of suspended particles.

- **Application**: Monitoring turbidity helps in assessing the overall cleanliness and purity of the water. Elevated turbidity levels can suggest the presence of contaminants or sedimentation.

5. **pH Sensor**:
    - **Explanation:** It measures the acidity or alkalinity of water and is crucial for determining water quality as pH levels can indicate if the water is acidic, neutral, or alkaline.
    - **Working**: It consists of an electrode that generates a voltage proportional to the hydrogen ion concentration in the water. This voltage is then converted into a pH value.
    - **Application**: By monitoring pH levels, we can detect if the water is becoming too acidic or basic, which could indicate contamination or other issues affecting water quality.
6. **Solenoid Valve Actuator**:
    - **Explanation**: It is used to control water flow based on sensor data and threshold conditions.
    - **Working**: It operates by energizing a coil to create a magnetic field, which opens or closes the valve. In our project, it allows for the regulation of water flow through the pipe based on the sensor readings.
    - **Application**: It enables us to take immediate action based on water quality data. For example, if the sensor detects poor water quality, the actuator can close the valve to prevent contaminated water from flowing further.
7. **Relay Module:**
    - **Explanation**: It serves as a switch to control the solenoid valve actuators based on sensor data and predefined thresholds.
    - **Working**: It receives signals from the microcontroller and triggers the opening or closing of the solenoid valves accordingly by controlling the voltage supply.
    - **Application**: By integrating the relay module, the system can automate actions such as stopping water flow in response to detected contamination.
8. **LCD Screen:**
    - **Explanation**: It provides a visual interface for displaying real-time water quality data and alerts to users.

- **Working**: It receives data from the microcontroller and displays information such as sensor readings and alert notifications.
- **Application**: It enhances user interaction with the monitoring system by providing feedback on water quality parameters, enabling users to stay informed and take appropriate actions as needed.

**9. Alert System: Using Wi-Fi and Internet Connectivity OR Bluetooth**

a. Connect the ESP32 to a Wi-Fi network using its Wi-Fi module or establish a Bluetooth connection between the ESP32 and a nearby device.
b. Use protocols such as HTTP or MQTT to send messages to a web-page.
c. Implement an alerting mechanism in your software code that triggers when water quality parameters exceed predefined thresholds.
d. Upon triggering the alert, use the Wi-Fi connection to send an HTTP POST request or publish a message via MQTT to a designated endpoint or topic..

## 4) Mention the details of the hardware features that you intend to implement.

First of all we will have to ensure that all the sensors are waterproof and can work under different conditions to allow them to work efficiently and without any damage.Also, here we are using ESP-32 as microcontroller,so we have to ensure that all sensors are compatible with ESP-32 and can be used with Thinkspeak.

**pH Sensor:**

- **Functioning:** The pH sensor measures the acidity or alkalinity of water by detecting the concentration of hydrogen ions. It consists of a glass electrode that generates a voltage proportional to the pH level of the solution it's immersed in.
- **Interface:** Connect the pH sensor's electrode to the ESP32's analog input pin.
- **Data Acquisition:** Use the ESP32's ADC to convert the analog voltage output from the pH sensor into pH values.
- **Calibration:** Calibrate the pH sensor by immersing it in known pH buffer solutions and adjusting calibration settings in the software to ensure accurate pH measurements.

- **Feedback Mechanism:** Implement software algorithms to compensate for variations in pH sensor readings, ensuring accurate and consistent measurements over time.

**Turbidity Sensor:**

- **Functioning:** The turbidity sensor measures the amount of suspended particles in water by analyzing the light scattered or absorbed by the particles in the water sample. It typically utilizes a light source and a photodetector to measure the intensity of light passing through the water.
- **Interface:** Connect the turbidity sensor's output to the ESP32's GPIO or analog input pins, depending on the sensor's interface type.
- **Data Acquisition:** Read the digital or analog signals from the turbidity sensor using GPIO pins or ADC to obtain turbidity values.
- **Calibration:** Calibrate the turbidity sensor by correlating its output readings with standardized turbidity solutions of known concentrations.
- **Feedback Mechanism:** Develop algorithms to adjust turbidity sensor readings based on calibration data and compensate for environmental factors affecting turbidity measurements.

**Temperature Sensor:**

- **Functioning:** The temperature sensor measures the temperature of water by detecting changes in electrical resistance, voltage, or frequency in response to temperature variations. It typically utilizes a thermistor or a digital temperature sensor.
- **Interface:** Connect the temperature sensor to the ESP32's GPIO pins or analog input pins, depending on the sensor type.
- **Data Acquisition:** Read the digital or analog temperature values from the sensor using GPIO pins or ADC.
- **Calibration:** Calibrate the temperature sensor by comparing its output readings with known temperature standards.

- **Feedback Mechanism:** Implement temperature compensation algorithms to adjust temperature sensor readings based on calibration results and maintain accuracy across varying temperatures.

**TDS Sensor:**

- **Functioning:** The TDS sensor measures the total dissolved solids (TDS) in water by measuring its electrical conductivity. It detects the conductance of ions in the water sample and correlates it to the concentration of dissolved solids.
- **Interface:** Connect the TDS sensor to the ESP32's GPIO pins or digital interface for data communication.
- **Data Acquisition:** Read digital signals from the TDS sensor using GPIO pins to obtain TDS values.
- **Calibration:** Calibrate the TDS sensor by comparing its output readings with standardized conductivity solutions of known concentrations.
- **Feedback Mechanism:** Develop algorithms to adjust TDS sensor readings based on calibration data and compensate for variations in water conductivity.

**Solenoid Valve Actuator:**

- **Functioning:** The solenoid valve actuator controls the flow of water in the system based on input signals from the microcontroller. When energized, the solenoid creates a magnetic field that moves a plunger, allowing water to flow through the valve. When de-energized, the solenoid closes the valve, stopping the flow of water.
- **Interface:** Connect the solenoid valve actuator to the ESP32's GPIO pins for control signals.
- **Data Acquisition:** No data acquisition required for the actuator; it receives control signals from the ESP32.
- **Calibration:** Calibrate the actuator's control parameters, such as opening and closing times, to ensure precise valve operation.
- **Feedback Mechanism:** Implement feedback mechanisms to monitor valve position or flow rates and adjust control signals as necessary for desired water flow control.

**Microcontroller (ESP32):**

- **Functioning:** The ESP32 serves as the brain of the system, coordinating the operation of sensors, actuators, and communication with external devices. It reads sensor data, processes it, and triggers actions based on predefined conditions. Additionally, it handles communication with cloud platforms or other external systems.
- **Interface:** Utilize the ESP32's GPIO pins to interface with sensors, actuators, and other peripheral devices.
- **Data Acquisition:** Read sensor data using ADC or digital interfaces and send control signals to actuators.
- **Calibration:** Implement calibration routines in software to calibrate sensor readings and actuator control parameters.
- **Feedback Mechanism:** Develop feedback loops in the software to monitor sensor readings, adjust control signals, and provide real-time feedback to users through the web interface or other communication channels.

# Data Collection Plan

### 1) Explain how you are planning to collect the data

Our data collection plan is very straightforward and will proceed in these following steps:-

1) **Data Collection :**
    a) Sensors will be installed in the water body which can range from a water tank installed in your house or some lake or pond ( our main focus is for household application).
    b) Sensors will measure important parameters like how clear the water is (turbidity), the amount of dissolved substances (TDS), the acidity or alkalinity (pH), and the temperature of the water. Sensors will monitor water continuously.

2) **Regular Checking :**

a) The Microcontroller installed would take data from sensors in a regular interval of time which can be every hour or every day according to the use case.
b) This ensures that we have the up to date information about the water quality in our reservoir.
c) Frequency of this checking procedure would be set up adequately because Lower sampling frequency may lead to delayed detection of water quality issues, while higher frequency may increase power consumption and data storage requirements.

**3) Preprocessing Data :**

a) We will do preprocessing of data before sending it to cloud service.
b) We will do Data Validation like checking range (like ph is from 0 to 14),error handling like detecting sensor malfunction or missing some data etc.
c) By performing data validation before sending data to the cloud, you can ensure that only reliable and accurate sensor data is transmitted, minimizing the risk of transmitting erroneous or inconsistent data.
d) This helps maintain data integrity and reliability.

**4) Data Transmission:**

a) After processing  the data is ready for transmission. For now we are planning to use Thingspeak as our cloud service for storing the data.
b) We can use communication protocols like MQTT or HTTPS.
c) For our implementation of a table top model we are considering to use HTTPS as it is fast and secure and we will use ThingSpeak's built-in encryption via HTTPS to protect our data during transmission.

**5) Real Time Monitoring:**

a) The cloud service offers a dashboard interface for monitoring water quality parameters in real time.

b) The dashboard will show sensor readings trends over time and any deviations from acceptable levels allowing authorities  to easily check the status of water quality.

c) Real time monitoring capabilities enable action in case of unusual conditions or contamination events reducing potential health risks and ensuring the safety of the water supply.

**6) Alerting Mechanism:**

a) An alert system will be incorporated into the platform to inform personnel or authorities about significant changes in water quality parameters.

b) Set thresholds for ranges of each parameter will trigger alerts when surpassed indicating possible issues with water quality or emerging patterns that need attention.

c) We will display alerts on the user web-page to ensure a timely response and appropriate measures by those in charge.

**7) Calibration and Maintenance:**

a) Ensuring the accuracy and reliability of collected data requires calibration and maintenance of sensors.

b) Calibration involves verifying sensor accuracy against known standards and adjusting settings as necessary.

c) Routine maintenance tasks, like cleaning sensors, replacing worn out components and updating firmware or software are carried out to uphold optimal sensor performance and reduce downtime.

d) Utilizing Redundant Sensors: Employing redundant sensors for each type enhances reliability and accuracy by cross-referencing measurements. This redundancy mitigates potential errors from individual sensor variations, thereby ensuring more robust data collection and analysis.

e) Calibrations for each sensor are written in hardware features.+

## 2) What are the possible outcomes you would like to infer from them?

a) **Identification of Contaminated Sources :** By keeping an eye on factors like total dissolved solids (TDS) the system is able to pick up on abrupt increases or changes that could signal the existence of impurities in the water source. Continuous monitoring gives households the ability to pinpoint causes of pollution like pipe rust, chemical leaks or bacterial development leading to prompt action to reduce health hazards.

b) **Assessment of Water Treatment Effectiveness :** Comparing the pH levels and turbidity of water before. After undergoing treatment processes, like filtration and disinfection enables households to evaluate the efficiency of their water treatment systems. This comparison helps households confirm that contaminants are effectively removed and that water quality standards are being upheld.

c) **Early Detection of Plumbing Issue :** Keeping an eye on the temperature and pH levels can assist in spotting problems, in home plumbing like corrosion or mineral buildup. Unexpected shifts in temperature or pH could signal issues, with the water heater's performance, pipe condition or water softening systems prompting homeowners to address them to prevent any damage.

d) **Personalized Health Recommendation and Emergency Alerts :** Ensuring that the pH level and TDS of household drinking water are regularly checked can offer insights into its safety. By examining water quality data alongside health profiles and dietary choices households can tailor their water consumption recommendations. This could involve adjusting pH levels for hydration or choosing water filters that meet filtration requirements.Real time monitoring systems and alert features give households a heads up on water quality concerns like chemical leaks or bacterial contamination. Immediate notifications via email or SMS empower homeowners to act whether by switching to water or seeking help from local authorities to safeguard themselves and their loved ones.

## Conclusion

In conclusion, the proposed Water Quality Monitoring System for IoT offers a proactive solution to the pressing issue of ensuring safe and clean drinking water within households. By leveraging real-time monitoring and analysis of key water quality parameters such as pH, temperature, TDS, and turbidity, the system provides homeowners with a reliable tool to detect and address potential contamination threats promptly and in real time.

The project's objectives, including real-time monitoring, data collection and analysis, an integrated alert system, and control mechanisms, aim to empower users with the means to safeguard their drinking water quality effectively. The user-friendly interface and data logging features enhance usability and provide valuable insights into water quality trends and potential issues.It helps users to take fast and efficient steps to ensure good water quality.

The hardware components, including sensors, actuators, and the ESP32 microcontroller, are carefully selected to ensure accurate and reliable data collection and control. The alerting mechanism, whether through Wi-Fi and internet connectivity or Bluetooth, ensures timely notifications to users, enabling them to take immediate action when needed.

Overall, the proposed Water Quality Monitoring System for IoT is a comprehensive and innovative solution that addresses the challenges of traditional water quality monitoring methods. It has the potential to significantly improve the safety and quality of drinking water in households, ultimately contributing to the health and well-being of individuals and communities.