DEVELOPMENT OF A WATER QUALITY TESTING SYSTEM USING SENSORS, MICROCONTROLLER AND LCD DISPLAY

BY

OKORO MARVELLOUS
ENG2006325

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN

BENIN CITY

OCTOBER, 2025.

DEVELOPMENT OF A WATER QUALITY TESTING SYSTEM USING SENSORS, MICROCONTROLLER AND LCD DISPLAY

BY

OKORO MARVELLOUS
ENG2006325

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING, FACULTY OF ENGINEERING, UNIVERSITY OF BENIN, BENIN CITY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING (B.ENG) DEGREE IN COMPUTER ENGINEERING.

OCTOBER, 2025.

CERTIFICATION

This project was completed by Okoro Marvellous from the Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City, and is hereby certified.

ENGR PROF S.T APEH Date

(Project Supervisor)

Engr. Dr. I. A. Edeoghon Date

(Head of Department)

DEDICATION

Firstly, I dedicate this work to God Almighty, our Creator and sustainer, whose unwavering guidance, boundless grace, and infinite wisdom have been our strength and inspiration throughout every step of this journey. May this work bring honour to His name and serve as a testament to His faithfulness and love.

This project is also dedicated to my beloved parents, Mr. and Mrs. Owere Clifford, whose love, sacrifices, guidance, and unwavering support have made this journey possible. Their encouragement has been my greatest motivation, and I am deeply grateful for everything they have done for me.

This project reflects the dedication, patience, and guidance of those who have stood by me through every step, and for their constant support, I am forever grateful.

ACKNOWLEDGEMENT

I sincerely express my profound gratitude to Almighty God for His guidance, wisdom, and strength throughout the duration of this project. By His grace, this work was successfully completed despite the challenges encountered.

I am sincerely grateful to my project supervisor, Prof. Simon T. Apeh, for his/her invaluable guidance, support, and mentorship throughout this project. His/her insightful feedback, encouragement, and patience greatly enhanced the quality of my work and helped me navigate the challenges I encountered during the research and writing process.

I would like to express my sincere appreciation to the Department of Computer Engineering and all the lecturers for their guidance, support, and encouragement throughout my academic journey. Their dedication to teaching, constructive feedback, and willingness to share their knowledge greatly contributed to the successful completion of this project.

I also wish to express my sincere appreciation to Aero Contractors Company of Nigeria Limited for providing me with the opportunity to undertake my industrial training. My heartfelt gratitude goes to my IT supervisors, Mr. Dami and Mr. Tope, whose guidance, mentorship, and technical support greatly enhanced my practical knowledge and contributed significantly to the successful completion of this project.

I am deeply grateful to my parents, Mr. and Mrs. Owere Clifford, for their unwavering love and support throughout my academic journey. I would also like to specially acknowledge my siblings, Francis Goodnews Naomi and Francis Chinedu, for their constant encouragement, motivation, and for always being there to support me through the challenges of this project.

I would like to sincerely appreciate my friends for their encouragement, support, and companionship throughout this project. I am especially grateful to my project group members, ANAGA IFEANYI-CHUKWU. J and ZACK O. JENNIFER for their teamwork, collaboration, and dedication, which were invaluable to the successful completion of this work. I am grateful to all others whose contributions are not specifically mentioned here.

ABSTRACT

This project presents the design and implementation of an Internet of Things (IoT) based water quality monitoring system that enables real-time assessment of key water parameters using low-cost sensors and microcontroller technology. The system employs an ESP32 microcontroller integrated with pH, turbidity, Total Dissolved Solids (TDS), and temperature sensors to measure the chemical and physical characteristics of water. The sensed data are processed, displayed locally on a 20×4 Liquid Crystal Display (LCD), and transmitted wirelessly to a custom web interface for remote visualization and storage.

The proposed system addresses the limitations of traditional laboratory-based water testing methods, which are typically expensive, time-consuming, and prone to human error. By leveraging IoT connectivity, the system provides continuous, accurate, and cost-effective monitoring of water quality, suitable for domestic, educational, and small-scale environmental applications. Experimental results show that the developed prototype effectively measures and reports water parameters with acceptable precision and reliability.

Overall, the system demonstrates a practical approach to automating water quality analysis, promoting accessibility, sustainability, and public health through real-time digital monitoring.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

IoT	internet of things
GPIO	General Purpose Input/Output
TDS	Total Dissolved Solids
PH	Potential of Hydrogen
ESP32	Espressif 32-bit Microcontroller
LCD	Liquid Crystal Display
DO	Dissolved Oxygen
CSV	Comma-Separated Values
Wi-Fi	Wireless Fidelity
ORP	Oxidation-Reduction Potential
NTU	Nephelometric Turbidity Unit
DC	Direct Current
OTA	Over-The-Air (programming/update)
CSSS	Cascading Style Sheets
HTML	HyperText Markup Language
SQL	Structured Query Language
EC	Electrical Conductivity
HTTP	Hypertext Transfer Protocol
DAC	Digital-to-Analog Converter
ADC	Analog-to-Digital Converter
WSN	Wireless Sensor Network
WQSM	Water Quality Monitoring System
MHZ	Megahertz
SRAM	Static Random Access Memory

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF STUDY

Water quality is essential for human health, industrial applications, and maintaining environmental balance. Conventional methods that use laboratory chemical analysis are typically characterized by delays, high manual workload, and a greater likelihood of human mistakes (Sugiharto et al., 2023). Recent advances in embedded systems and the Internet of Things (IoT) have enabled the automation of real-time water quality monitoring using sensors to measure key parameters, including pH, turbidity, and Total Dissolved Solids (TDS). Microcontrollers, such as ESP32, equipped with built-in Wi-Fi, enable compact, low-cost, and remotely accessible monitoring solutions (Sugiharto et al., 2023).

Even with ongoing technological improvements, many communities, particularly rural and low-income ones still face barriers because available monitoring tools are expensive, bulky, and demand regular servicing. Detecting pollutants like heavy metals often requires sophisticated optical or chemical equipment that is not affordable for widespread use. Additionally, conventional testing procedures do not support real-time communication of results, causing delays in identifying and addressing contamination.

To address these limitations, this work proposes an IoT-enabled solution that uses electronic (non-chemical) sensors to track pH, turbidity, and TDS levels in water. The goal is to deliver an accessible and real-time monitoring platform that enhances water safety and promotes healthier living conditions.

1.2 PROBLEM STATEMENT

Every living organism on Earth depends on water for survival. The human body itself is composed of more than 60% water (Cheng & Yu, 2003). Despite the fact that nearly 70% of the planet is covered by water, only a small proportion is fresh and safe enough to support essential activities including domestic consumption, agriculture, industrial applications, recreation, and fisheries. (Taru & Karwankar, 2017).

Existing monitoring solutions remain impractical due to their high cost, complexity, and technical limitations, as they are laboratory-focused. Many current systems rely on expensive sensors, lack real-time data transmission, and are too complicated for use in rural communities with limited resources. To address these issues, the project proposes an affordable IoT-driven solution for monitoring water quality, offering real-time and easily interpretable data without the need for costly laboratory procedures. The objective is to develop a user-friendly platform that equips communities with the tools to consistently monitor and verify the safety of their water supplies.

1.3 AIMS AND OBJECTIVES

A key objective of this project is to implement a system capable of continuously monitoring drinking water quality, employing sensors, microcontrollers, and an LCD for instant, remote feedback.

The following are pursued to achieve the aim:

1. To implement an inexpensive system capable of measuring water quality through pH, turbidity, and TDS sensors.
2. To interface the sensors with a microcontroller (ESP32) for real-time data collection.
3. To include a real-time LCD for immediate on-site water quality feedback
4. To develop a web or mobile-based platform for displaying live water quality readings

5. To examine how accurately the system can monitor the main parameters that define water quality

1.4 SCOPE OF STUDY

This study focuses on the design and implementation of an IoT-based Water Quality Monitoring System capable of measuring and analyzing essential water parameters in real-time. The system will employ sensors to detect key chemical properties, such as pH, Turbidity, and Total Dissolved Solids (TDS), which serve as significant indicators of water quality.

The sensed data will be processed by a microcontroller (ESP32) and displayed on an LCD screen, enabling immediate local visualization of the measured parameters. In addition, the system will feature Internet of Things (IoT) integration, enabling the transmission of collected data to an online cloud platform, such as ThingSpeak, for continuous remote monitoring, analysis, and storage. Through this interconnection, users will be able to observe real-time variations in water quality from any location with internet access.

The scope of this study covers hardware design, sensor calibration, software development, IoT data transmission, and cloud-based data visualization. However, it excludes large-scale water treatment or industrial purification processes, focusing primarily on monitoring and data reporting for household, educational, or small-scale environmental applications.

1.5 JUSTIFICATION OF STUDY

Monitoring various water quality parameters provides a clear understanding of the factors essential for maintaining a healthy life and preventing water wastage. The use of the Internet of Things (IoT) enables real-time monitoring and control of water quality through seamless data integration and analysis (AlMetwally et al., 2020a). However, traditional water quality

monitoring methods are often expensive, time-consuming, and require frequent manual sampling and laboratory analysis. Therefore, this study aims to develop an IoT-based water quality monitoring system that measures key parameters, such as pH and Total Dissolved Solids (TDS), in real time, displays them on an LCD screen, and transmits the data to an online platform, such as a customized webpage, for remote access and continuous analysis. This approach provides a cost-effective, efficient, and scalable solution for maintaining and managing water quality in both domestic and environmental applications.

CHAPTER TWO

LITERATURE REVIEW

2.1 A BRIEF OVERVIEW OF WATER QUALITY

The conventional method of testing water quality is to manually gather water samples and send them to the lab for testing and analysis. This method is time-consuming, inefficient, and not economical (Sugiharto et al., 2023). Water quality plays a vital role in maintaining environmental integrity, protecting public health, and supporting sustainable ecosystems. It encompasses the chemical, physical, and biological properties of water, which determine its appropriateness for different purposes such as domestic consumption, agricultural activities, industrial processes, and the survival of aquatic organisms. Over the years, there has been a growing need to test water quality. This is due to the ever-increasing number of water pollutants, ranging from heavy metals such as Mercury (Hg), lead (Pb), and Arsenic (As) to Salinity. Heavy chemicals and salinity are said not only to be detrimental to water quality, but also to be factors, along with Turbidity, pH value, concentration of dissolved minerals, and amount of dissolved oxygen, that contribute to it.

The objective of the review is to study various conventional and modern methods of monitoring water quality to identify their strengths and weaknesses. The methods include the Internet of Things (IoT) (Zainurin et al., 2022).

2.2 ROLE OF IOT IN WATER QUALITY TESTING

IoT offers significant benefits for water quality by providing real-time data for timely decisions on water usage and treatment, especially in water-scarce areas. It also helps identify and track pollution sources, which is crucial for protecting public health and the environment (Miller et al., 2023).

Continuous monitoring of different water quality parameters is essential for understanding the requirements for healthy living and minimizing water wastage. The application of the Internet of Things (IoT) technology facilitates real-time observation and management of water quality conditions. The proposed system employs IoT-based sensors to monitor key water quality parameters, including pH, temperature, and turbidity, particularly for domestic applications. (AlMetwally et al., 2020).

2.3 BENEFITS OF TESTING WATER QUALITY

Water is one of the most essential natural resources given to humanity. However, rapid societal development and numerous human activities have accelerated contamination and degraded water resources. Testing water quality is essential as it ensures safe drinking water and protects human health by detecting harmful contaminants early. Additionally, it contributes to environmental protection, preserves the balance of aquatic ecosystems, assists farmers in adopting safe irrigation methods, enables industries to meet environmental standards, and encourages the sustainable and efficient utilization of water resources.

Water is used for drinking, agricultural work, industrial operations, and cleaning. Water needs to be clean and safe because contaminated water can endanger human and aquaculture ecosystems.

2.4. WATER QUALITY BY DEFINITION

Water is one of the most essential natural resources for human survival and environmental sustainability. Due to its limited availability and its vital role in supporting biological processes, economic development, and other growth-related activities, the proper management of water resources is of great importance. Water quality describes the chemical, physical, biological, and radiological properties of water, which influence its appropriateness for different applications, including domestic consumption, agricultural use, industrial

operations, and the maintenance of healthy ecosystems. These characteristics collectively serve as indicators used to evaluate the overall condition and quality of water.

Table 2.1: Representation of the characteristics of water.

Chemical Properties	**Biological Properties**	**Physical Properties**
Dissolved oxygen (DO), chemical oxygen demand (COD)	Bacteria	Turbidity
Biological Oxygen demand	Algae	
Ph level	Viruses	Temperature
Ammonia		Color
Salinity		Taste and Odor
Harness		Suspended
Organic Compounds		Solid
Metals		Metal

S/N	Parameter	Quality Range	Units
1	pH	6.5–8.5	pH
2	Electrical Conductivity	500–1000	μS/cm
3	Turbidity	0–5	NTU
4	ORP	650–800	mV
5	Temperature	–	°C
6	Free Residual Chlorine	0.2–2	mg/L
7	Dissolved Oxygen	–	mg/L
8	Nitrates	<10	mg/L

Fig. 2.1: Water quality parameters proposed by WHO and South Africa, DWA (Sithole et al., 2019)

2.5 SENSORS BY DEFINITION

A **sensor** is an electronic device that detects and measures physical, chemical, or biological properties (such as temperature, light, pressure, motion, or chemical concentrations and converts them into a readable signal (usually electrical or digital). Sensors are essential in automation, monitoring, and control systems across various industries.

2.6 ROLE OF SENSORS IN WATER QUALITY TESTING

The Internet of Things (IoT) is a transformative technology that is reshaping our world by deploying trillions of sensors and actuators to create intelligent, interconnected environments. (Sehrawat and Gill, 2019). It is recommended that the water pH be between 6.5 and 8.5.

Sensors play an essential role in automating applications by measuring and processing data to detect changes in physical conditions. Whenever there is a change in any physical condition that a sensor is designed to detect, it produces a measurable response (Sehrawat & Gill, 2019).

Sensors play a crucial role in water monitoring by continuously measuring pH levels, turbidity, and TDS to detect contamination.

2.6.1 CHALLENGES OF TRADITIONAL WATER QUALITY MONITORING METHODS

Traditional methods of water quality monitoring primarily rely on manual sampling and laboratory-based chemical analysis to determine key parameters such as pH, turbidity, dissolved oxygen, and total dissolved solids (TDS). While these methods are often accurate and standardized, they face several challenges that limit their efficiency, accessibility, and practicality, especially in developing regions and rural communities.

One major challenge is the time-consuming nature of traditional testing. Water samples are typically collected from the field and transported to laboratories for analysis, which introduces delays between sampling and obtaining results. This lag can lead to inaccurate assessments, particularly when water quality fluctuates rapidly due to pollution or environmental changes. The delay also hinders timely intervention in cases of contamination or sudden deterioration in water quality.

Another limitation is the high operational cost associated with laboratory equipment, chemical reagents, and skilled personnel. These resources are often scarce or unavailable in low-income and rural areas, making continuous and widespread water monitoring impractical. Furthermore, the maintenance and calibration of laboratory instruments add to the overall cost burden, reducing the sustainability of such methods.

Traditional monitoring techniques also face challenges in scalability and coverage. Manual sampling can only cover limited geographical areas, and the frequency of testing is often reduced due to resource constraints. As a result, water bodies such as rivers, wells, and

reservoirs may go unmonitored for extended periods, increasing the risk of undetected contamination.

Additionally, traditional methods are prone to human error and sample degradation. Improper handling, storage, or delays during transportation can alter the chemical composition of samples, leading to inaccurate readings. Inconsistent sampling procedures and variations in laboratory protocols may further compromise data reliability and comparability.

Finally, traditional approaches generally lack real-time data acquisition and remote accessibility. The absence of automation means that water quality data cannot be monitored continuously or transmitted wirelessly. This limitation restricts the ability of authorities and users to make timely decisions or implement early warning systems against contamination events.

In summary, traditional water quality monitoring methods, though reliable for laboratory precision, are constrained by high cost, delayed results, limited scalability, and lack of real-time data. These challenges have motivated the development of automated, sensor-based, and IoT-enabled systems, which offer continuous, low-cost, and remote water quality monitoring for improved public health and environmental management.

2.7 META-ANALYSIS TABLE.

Table 2.2: META Analysis Table

S/N	Title	Author	Background	Objective	Methodology	Result	Research Gap
1	Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas	(Bogdan et al., 2023a)	Water quality impacts health; there is a need for affordable monitoring in rural areas.	Develop a low-cost IoT system to monitor rural water quality	Arduino-based system with Bluetooth & sensors (TDS, pH, temp, turbidity), mobile app interface	All but one water source was safe (TDS > 500 ppm in one)	Limited Bluetooth range, no cloud integration, sensor calibration not discussed
2	Real-Time Water Quality Monitoring with Chemical Sensors	(Yaroshenko et al., 2020a)	Need for rapid detection of water contamination incidents	Critically assess real-time chemical sensor technologies for field use	Review of polymer and microwave-based sensors tested in real-life scenarios	Identifies technologies with high sensitivity, selectivity, and field stability	Challenges in integration, long-term stability, and mass production remain
3	IoT-Based Smart Water Quality Monitoring: Recent Techniques, Trends, and Challenges for Domestic Applications	(Jan et al., 2021a)	Traditional WQM methods are manual, slow, and lack real-time feedback; WSN-based methods suffer from energy, security, and communication issues.	Survey state-of-the-art IoT-WQMS for domestic use and propose an efficient system design.	Critical review of WQM parameters, sensors, IoT systems, empirical metric-based evaluation, and design recommendations	Identifies the pros and cons of current systems; emphasizes the potential of IoT for smart homes/offices.	Challenges in energy efficiency, data security, communication range, and integration persist.
4	Design and Development of Smart Water Quality Monitoring System Using IoT	(Sanya et al., 2022)	Traditional water monitoring lacked real-time sensors, resulting in reduced accuracy, lower efficiency, and manual data collection.	Propose a real-time IoT platform using LoRa for improved water-quality monitoring.	Five-part system: sensor devices, IoT board, LoRa gateway, cloud server, user domain; data analyzed via MATLAB	Provides real-time data access, improves accuracy, and enables supervision of device status via the cloud	The text lacks a comparison to existing systems and a discussion on long-term deployment challenges and maintenance.

2.8 RESEARCH GAPS

The development of the proposed Internet of Things (IoT) based water quality monitoring system is built upon existing research that leverages embedded systems and smart sensing to move beyond traditional, laboratory-based water testing. The following section reviews key works summarized in the Meta Analysis Table (Table 2.1) to establish the state-of-the-art, identify successful strategies, and define the critical research gaps that the current project is designed to address.

1. Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas (Bogdan et al., 2023b).

This study was designed to create a low-cost IoT-based solution for monitoring water quality in rural communities where traditional water testing methods are not readily accessible. Its emphasis on affordability and suitability for underserved areas closely corresponds with the goals of the present research project

The authors implemented an Arduino-based system that incorporated sensors for Total Dissolved Solids (TDS), pH, temperature, and turbidity[5]. Data communication was handled through Bluetooth, and the system provided an interface via a mobile application. The results indicated that the majority of tested water sources were safe, although one source had a TDS level exceeding 500ppm This successfully demonstrated the feasibility of using low-cost hardware for basic field-level assessment.

Despite its low cost approach this particular system presented limitations in terms of usage as a comprehensive, long term monitoring solution. Short Bluetooth range connectivity has been highlighted as an issue in this particular system, and the system lacked real time remote monitoring for long term trend analysis. This project facilitates the use of Wi-Fi connectivity

and also integration of a customized webpage for visual display of the ESP32 for data transmission.

2. Real-Time Water Quality Monitoring with Chemical Sensors (Yaroshenko et al., 2020b).

The goal of this work was to critically assess existing real-time chemical sensor technologies suitable for rapid detection of water contamination incidents in field-use scenarios. This review focused on the technical efficacy of specialized sensor types.

The methodology involved a thorough review of various advanced chemical sensing technologies, including polymer and microwave-based sensors. The review successfully identified several technologies demonstrating high sensitivity and selectivity, along with satisfactory field stability for critical contamination detection.

The method of data collection has been highlighted as an issue: manual collection and analysis in the lab. This project facilitates continuous measurement and transmission of water quality data, allowing immediate detection of contamination.

3. IoT Based Smart Water Quality Monitoring: Recent Techniques, Trends and Challenges for Domestic Applications (Jan et al., 2021b).

This work performed a comprehensive survey of state-of-the-art IoT Water Quality Monitoring Systems (IoT-WQMS) for domestic applications. The goal was to propose a more efficient system design by evaluating the strengths and weaknesses of current deployments.

The methodology centered on a critical review of water quality parameters, sensors, and existing IoT system architectures. The review highlighted that traditional Water Quality Monitoring (WQM) is slow and lacks real-time feedback, while earlier Wireless Sensor Network (WSN)-based systems struggled with energy efficiency and security. The paper

successfully identified the strong potential of IoT-WQMS for enhancing monitoring in smart homes and offices.

This aim aim to solve problems such as energy efficiency by incorporating ESP32 Microcontroller known for its lower power consumption and integrated Wi-Fi capabilities, directly addressing concerns about energy and communication range.

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION/OVERVIEW

This study was designed to create a low-cost IoT-based solution for monitoring water quality in rural communities where traditional water testing methods are not readily accessible. Its emphasis on affordability and suitability for underserved areas closely corresponds with the goals of the present research project.

The research concentrates on the development of a real-time water quality monitoring system capable of measuring essential parameters such as pH, turbidity, and Total Dissolved Solids (TDS). These indicators are crucial in assessing water purity and determining its suitability for human consumption as well as its safe application within the environment.

To accomplish this, the system integrates multiple electronic sensors connected to an ESP32 microcontroller, which functions as the central processing unit. The ESP32 acquires analogue signals from the sensors, converts them into meaningful digital data, and presents the measured values on an LCD for on-site monitoring. Furthermore, the system incorporates Internet of Things (IoT) technology by transmitting the processed information to a cloud-based platform, ThingSpeak, through Wi-Fi communication. This approach allows users to remotely monitor, visualise, and store water quality information over extended periods using graphical interfaces and trend-based analysis.

The method also involves hardware circuit design, software algorithm design, and sensor calibration to ensure accuracy and reliability. The integration provides continuous and automated monitoring, eliminating the drawbacks of conventional water testing methods, which tend to be manual, time-consuming, and costly.

In conclusion, this chapter outlines the systematic procedures employed in the design, implementation, and assessment of an IoT-based water quality monitoring system. The developed system delivers real-time data insights, supports environmental sustainability, and improves the effectiveness of water resource management practices.

3.1 SYSTEM ARCHITECTURE

The architecture of the proposed water quality monitoring system is structured to continuously monitor and evaluate parameters including pH, Total Dissolved Solids (TDS), turbidity, and temperature in real time. The system comprises several key components, namely the ESP32 microcontroller, pH sensor, TDS sensor, temperature sensor, turbidity sensor, and an LCD display unit.

The ESP32 microcontroller functions as the central processing unit of the system. It acquires analog signals from the various sensors, converts them into digital data, processes the obtained measurements, and presents the results on the LCD screen. In addition, the ESP32 utilizes its built-in Wi-Fi capability to wirelessly transmit the processed information to a cloud-based IoT platform, thereby facilitating remote monitoring, data storage, and further analysis.

The pH sensor measures the hydrogen ion concentration in the water, which determines its acidity or alkalinity. It provides an analog output proportional to the pH, which is read and calibrated by the ESP32.

The TDS sensor is used to measure the concentration of dissolved substances in water, including minerals, salts, and other impurities. It operates by assessing the electrical conductivity of the water sample and converting this measurement into Total Dissolved Solids (TDS) values, typically expressed in parts per million (ppm).

The temperature sensor is included to provide the capability for thermal compensation on both pH and TDS readings since the temperature of water directly influences sensor accuracy. The ESP32 automatically compensates and corrects reading values using this data.

An LCD display (20×4) is interfaced with the ESP32 via I²C communication pins to display real-time readings of the pH, temperature, and TDS values. This enables users to visually monitor the water quality status locally without needing an external device. In use, the sensors are immersed in water sample to be monitored to continuously collect data. The ESP32 processes the input, displays the values on the LCD, and transmits the readings to a web or cellular IoT platform for remote viewing and data storage. The combination of sensing, processing, and communication units offers a small, efficient, and dependable platform for real-time water quality monitoring. The web platform enables data extraction in the form of CSV, which enables user to have a record of all monitored water parameters. This feature is necessary in identifying trends, detecting anomalies, and making informed decisions on water treatment based on extracted data.

3.2. SYSTEM FLOW CHART

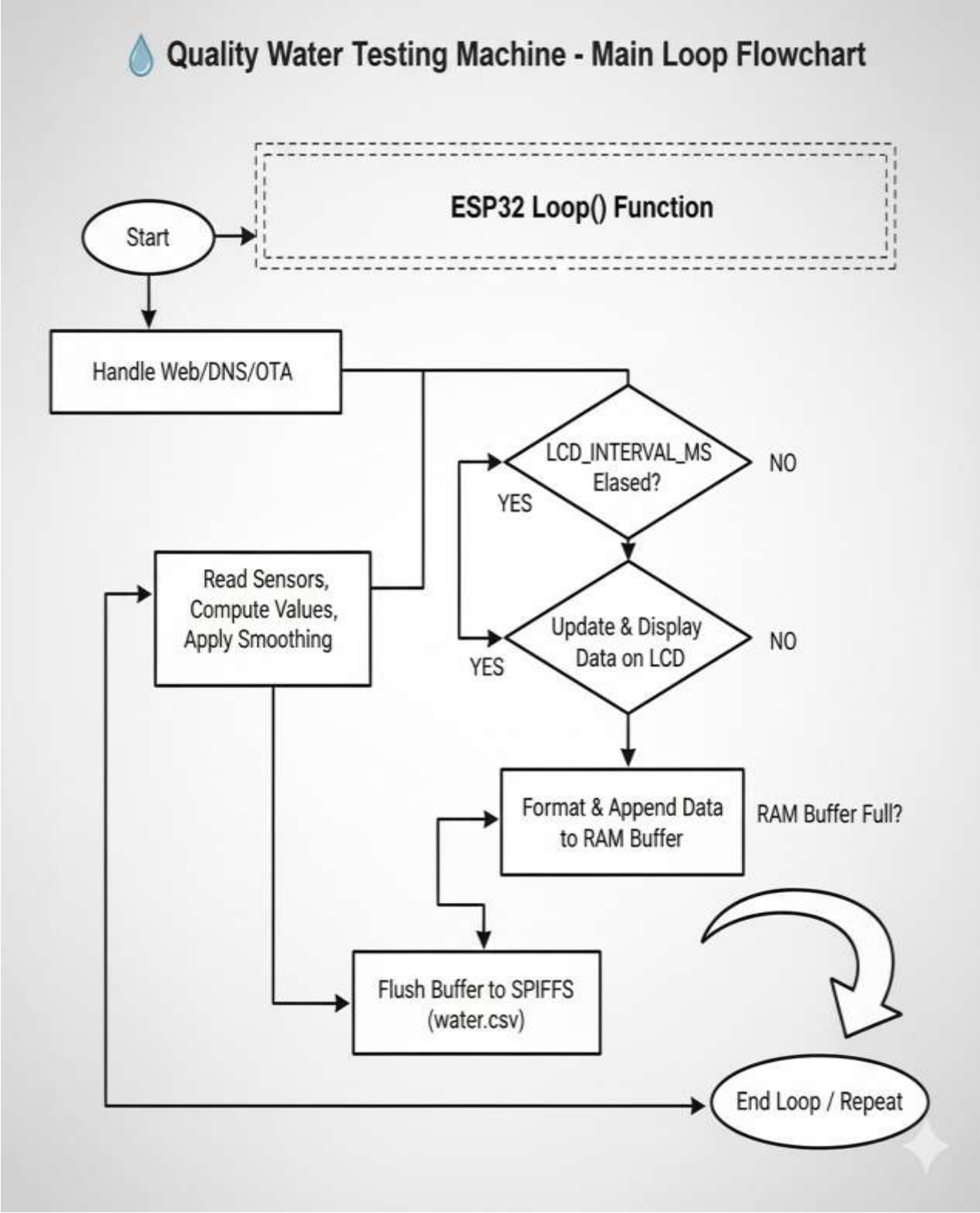


Fig 3.1: System flow chart

3.3. SYSTEM BLOCK DIAGRAM

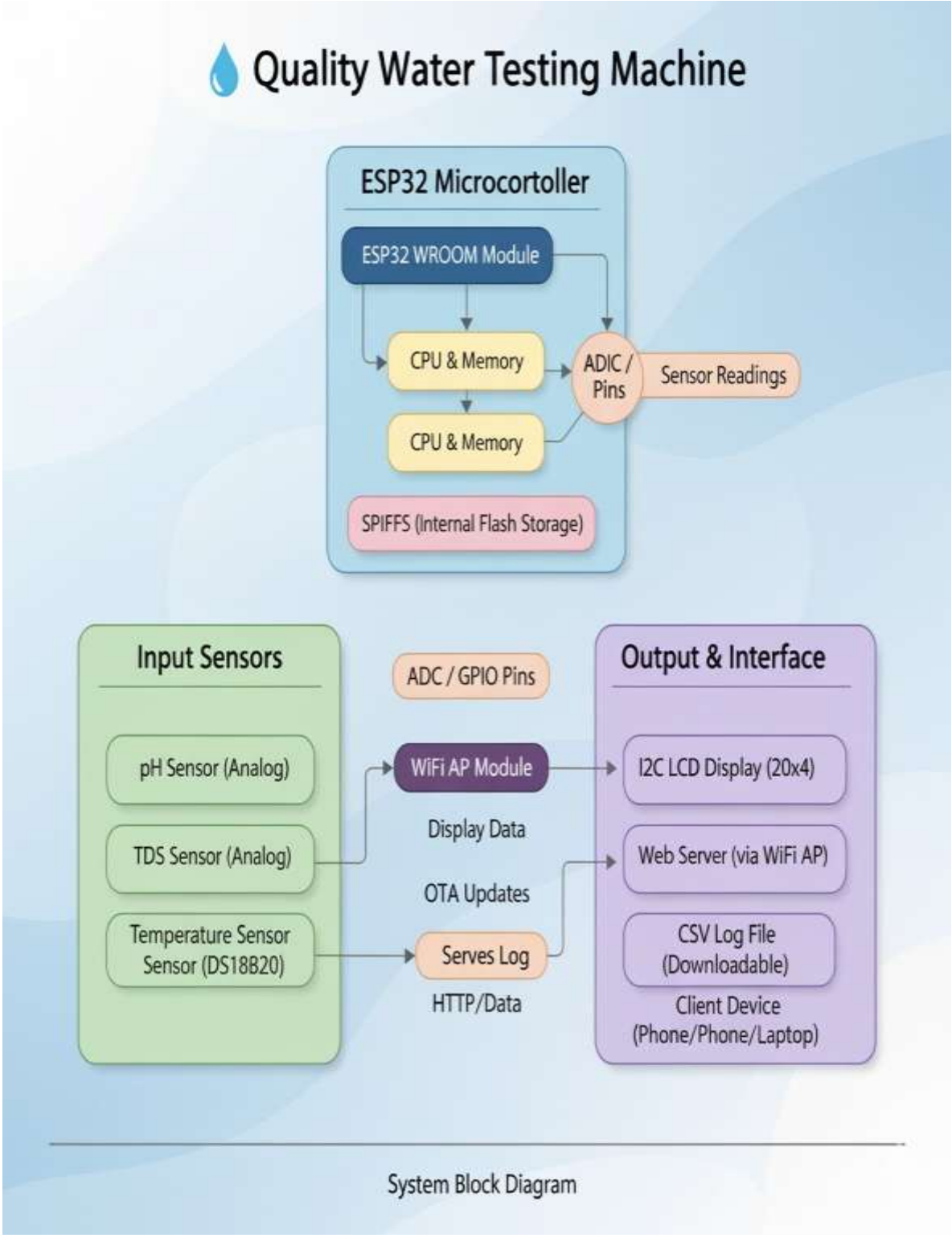


Fig 3.2: System block Diagram

3.4. HARDWARE DESIGN AND COMPONENTS

The system consists of various hardware components, which are;

3.4.1. Microcontroller (ESP32)

The ESP32 microcontroller serves as the central controller of the water quality monitoring system. It is employed to sense sensor values, process values, and send them to the internet for remote observation. The ESP32 possesses several analogue-to-digital converter (ADC) pins that capture analogue output from sensors such as the pH sensor and TDS sensor, converting them into digital values for computation.

Additionally, the ESP32 controls the LCD, displaying real-time visualisation of observed parameters like pH level, temperature, and total dissolved solids. The ESP32 connects the system with the cloud platform (ThingSpeak) or any IoT dashboard through its onboard Wi-Fi module, offering transparent data transfer and online monitoring.

In the design and implementation of this project, the choice of the microcontroller fell on the ESP32 instead of the traditional Arduino board. The primary reason for this was the ESP32's superior characteristics and compatibility with the Internet of Things (IoT) applications. The ESP32 has a two-way connection in the form of Wi-Fi and Bluetooth, which wipes off the need for electrical modules that are usually necessary with the Arduino. The use of the wireless feature allows for the easy transfer of data to the cloud platforms like ThingSpeak and this in turn supports real-time monitoring and remote access.

To be precise, the ESP32 is not only more potent in terms of performance but also provides a larger memory compared to the Arduino Uno. The dual-core 32-bit processor of the ESP32, which can go as fast as 240 MHz, and its huge Flash and SRAM memory together permit the ESP32 to receive inputs from several sensors and perform complicated tasks at the same time without making the system slow. On the other hand, the low-speed Arduino Uno, which is

clocked at only 16 MHz, has limitations as regards multitasking and hence is less productive for the projects that have the requirement of sensor reading and network communication at the same time.

Another plus point is that the ESP32 has more input/output pins with many selectable functionalities like ADC and DAC conversions, PWM, capacitive touch sensing, plus serials. The latter features have made the ESP32 really compatible with different sensor connections and peripherals without needing extra external circuits. Moreover, it has the capability to run under power-saving conditions such as deep sleep that are very important for the energy-efficient battery-operated applications, more so in remote or rural areas where power is cut off very often.

One more significant benefit is the fact that it has better support for IoT communication protocols, such as HTTP, HTTPS, and MQTT, which makes integration with online dashboards and cloud services more accessible. This turns it into the perfect option for environmental monitoring systems that work continuously such as water quality analysis, which needs the transmission of the data in real-time. On the other hand, even though the ESP32 has great capabilities, it is still affordable, giving you a high performance and functionality level at a price that is almost the same or even lower than that of the Arduino, provided that the cost of external communication modules is considered.

In summary, the ESP32 is a water quality monitoring system solution that is more compact, efficient, and scalable, giving the reliability of data acquisition, processing, and wireless communication while keeping the power consumption low and the prices affordable. These benefits make it the preferred option for the creation of smart, connected, and real-time IoT-based applications.

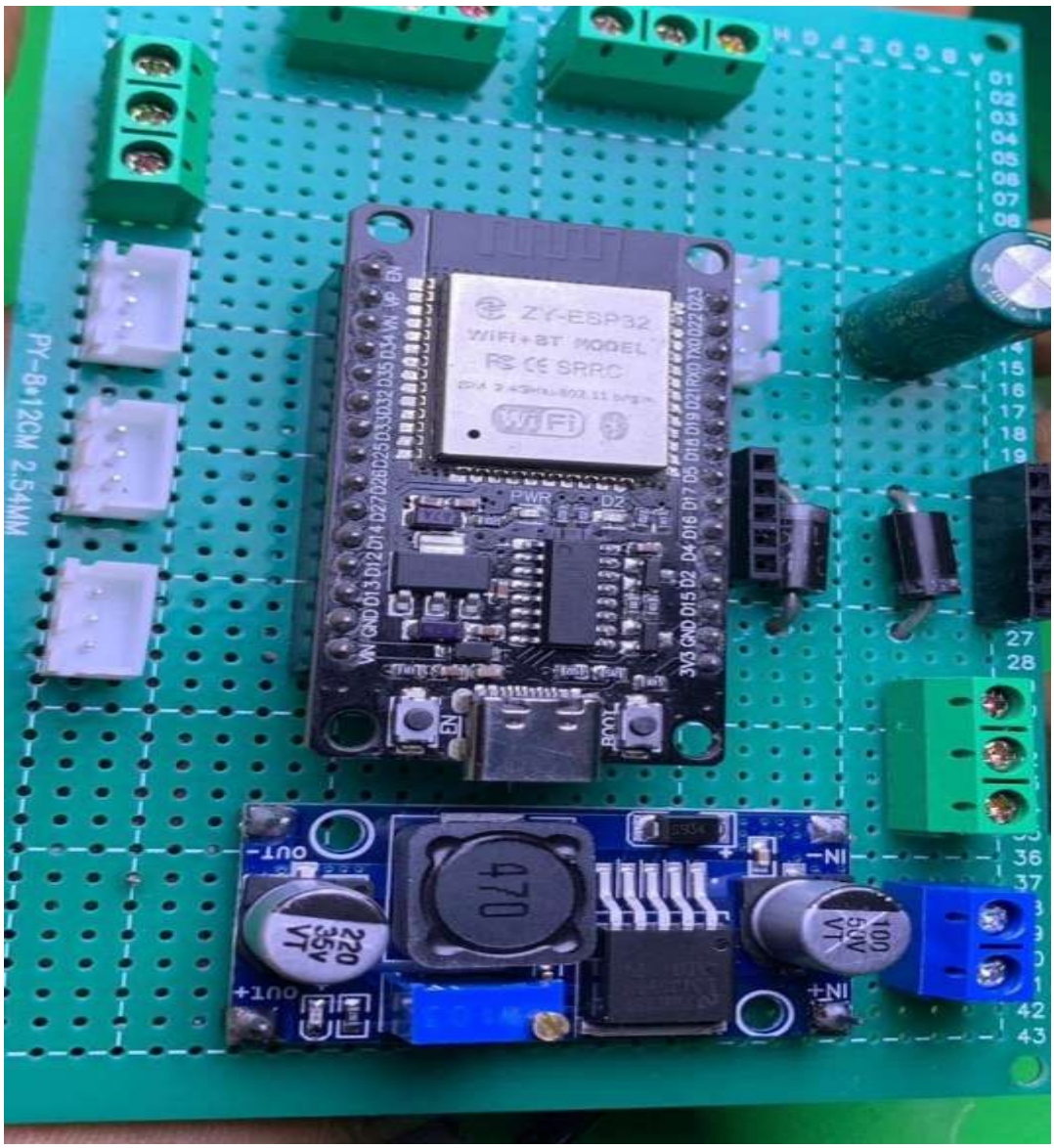


Fig 3.3: ESP32 Microcontroller mounted on the veroboard.

Fig 3.4: The Engineering Projects. (2024, March). ESP32 development board pinout. The Engineering Projects.

https://images.theengineeringprojects.com/image/webp/2024/03/esp32pinout.jpg.webp?ssl=1

The above figure represents the pin diagram and configuration of the ESP32 microcontroller

3.4.2. TDS SENSORS

Total dissolved solids (TDS) refer to the concentration of inorganic salts and trace quantities of organic substances dissolved in water. The major components commonly include cations such as calcium, magnesium, sodium, and potassium, as well as anions including carbonate, bicarbonate, chloride, sulfate, and nitrate . TDS levels in water are typically expressed in milligrams per litre (mg/L), which is numerically equivalent to parts per million (ppm). Generally, a higher TDS value indicates that more dissolved solids are present in the water, making it less pure. Thus, the TDS level serves as an important indicator of water cleanliness.

Key specifications:

- Input Voltage: 3.3 ~ 5.5V
- Output Voltage: 0 ~ 2.3V
- Working Current: 3 ~ 6mA
- TDS Measurement Range: 0 ~ 1000ppm
- TDS Measurement Accuracy: ± 10% F.S. (25 ℃)
- Module Size: 42 * 32mm
- Module Interface: PH2.0-3P
- Electrode Interface: XH2.54-2P

Fig 3.5: TDS sensor

3.4.3. TURBIDITY SENSORS

Traditionally, turbidity measurement requires collecting water samples on-site and analyzing them in a laboratory, a process that is both time-consuming and expensive (Mukhopadhyay & Mason, 2013). Turbidity is a measure of water's cloudiness, caused by suspended particles like silt, clay, algae, and other organic matter that scatter light. The unit of measurement is the Nephelometric Turbidity Units (NTU).

Key Specifications:

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time: <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
    - Analog output: 0-4.5V
- Operating Temperature: 5℃~90℃
- Storage Temperature: -10℃~90℃
- Weight: 30g

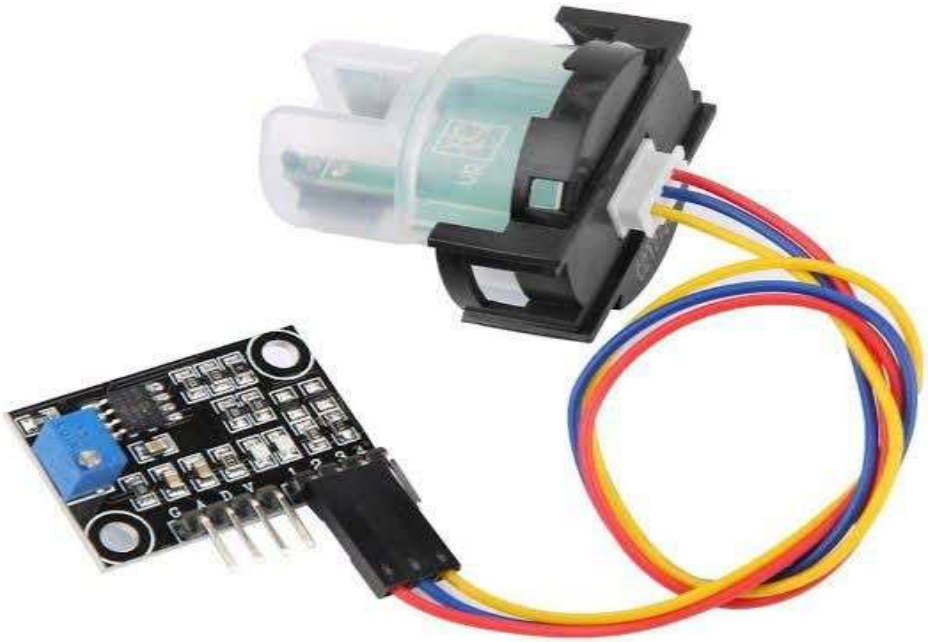


Fig 3.6: Turbidity Sensors

3.4.4. PH SENSOR

pH is a parameter used to determine the degree of acidity or alkalinity of a solution. It is mathematically expressed as the negative logarithm of the hydrogen ion concentration present in the solution. The pH scale ranges from 0 to 14 and operates on a logarithmic basis, reflecting variations in hydrogen ion concentration. Values below 7 indicate acidic conditions, whereas values above 7 represent alkaline or basic conditions.

$$pH = -\log(H+)$$

A low pH level in water increases its potential for corrosion, which can indirectly affect health by causing higher ingestion of metals from pipes and reducing the effectiveness of

disinfection. For chlorine to disinfect water efficiently, the pH should be maintained below 8 (Bogdan et al., 2023b).

Fig 3.7: Ph sensor

3.4.5. POWER SUPPLY (LITHIUM RECHARGEABLE BATTERY)

This is the important component that provides power to the system. It is made up of 6 lithium battery fussed together with an added integration of Battery Management system (BMS) board. The Power management section is responsible for providing a stable and regulated supply of voltage to all components, ensuring efficient and safe operation of the IoT based water quality monitoring system.

The 3S BMS board (as shown in *Figure 3.6*) manages and protects the three series-connected 18650 lithium-ion cells that form the system's main power source. It performs essential battery protection functions, including overcharge, over-discharge, short-circuit, and overcurrent protection. The BMS also balances the voltage levels of individual cells during charging and discharging to enhance battery efficiency and prolong its lifespan. The labels on

the board (0V, 4.2V, 8.4V, and 12.6V) correspond to the individual cell connections and the total pack voltage, respectively.

The battery pack provides a total nominal voltage of approximately 11.1V (3.7V × 3), which is then regulated by the DC-DC buck converter module visible beside the ESP32. This converter steps down the battery voltage to a stable 5V or 3.3V, depending on the requirements of the ESP32 board and other components such as sensors and the LCD display. The adjustable voltage regulator ensures that fluctuations in battery output do not affect the system's operation.

In addition, filter capacitors are connected across the input and output terminals of the power lines to minimize electrical noise and voltage ripples. This improves the stability of sensor readings and enhances communication.

Overall, this configuration ensures an efficient, safe, and consistent power supply for the entire system, with the BMS board providing intelligent battery protection and the DC converter guaranteeing regulated voltage delivery to all modules.

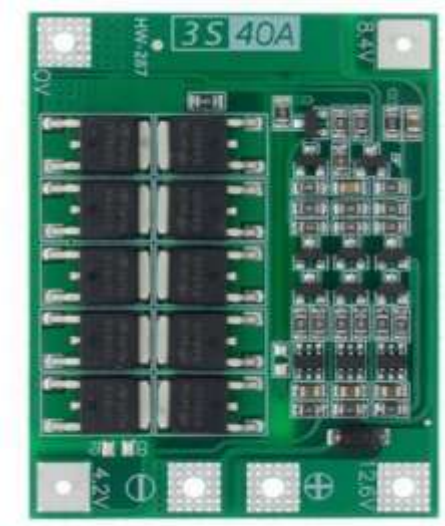


Fig 3.8: BMS Board

Fig 3.9: Lithium Battery

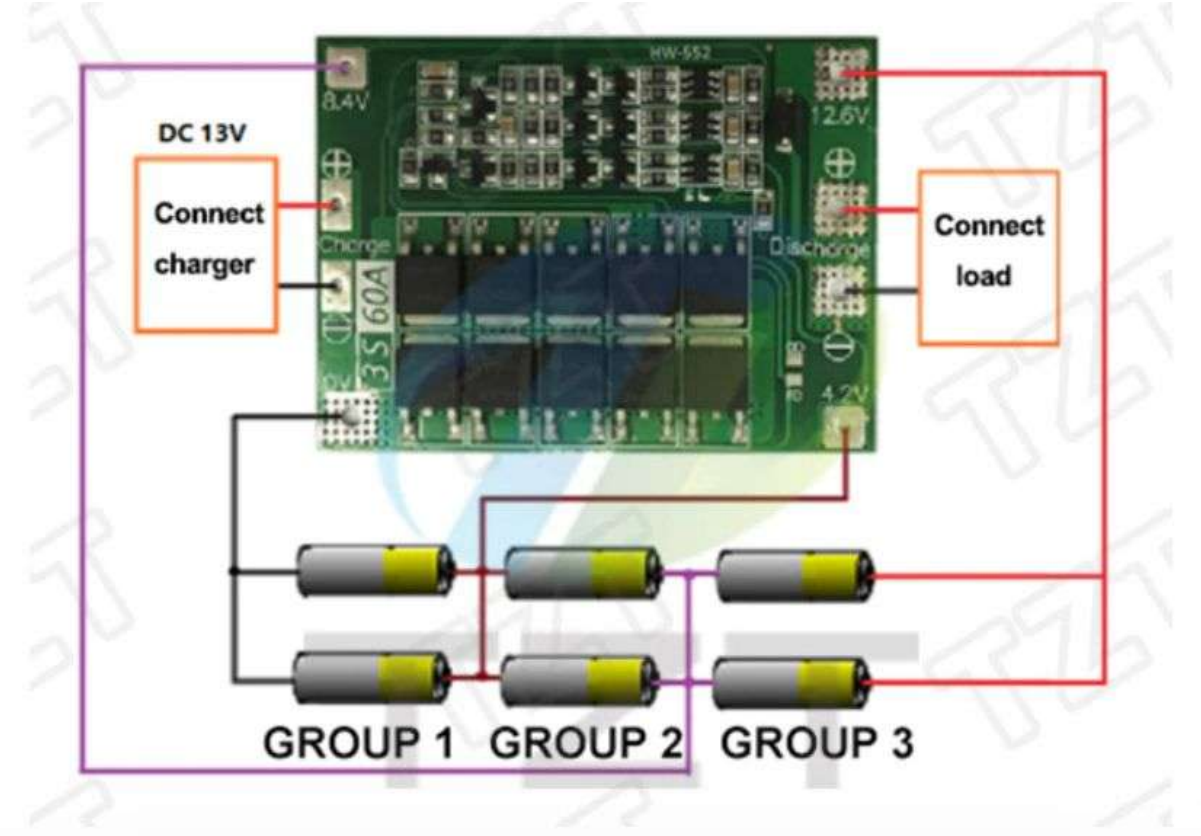


Fig 3.10: Configuration Schematic with the lithium battery.

The battery arrangement shown in the figure consists of six lithium-ion cells connected through a 3S 40A (or 60A) Battery Management System (BMS) board. The configuration follows a 3S2P connection, meaning that the batteries are grouped into three sets connected in series, with each group containing two cells connected in parallel.

Within each group, the two batteries are linked in parallel, allowing their capacity (mAh) to combine while maintaining the same voltage as a single cell (approximately 3.7 V nominal and 4.2 V when fully charged). The three groups are then connected in series to increase the overall output voltage of the pack. As a result, the battery pack provides a nominal voltage of about 11.1 V (3.7 V × 3) and a total capacity that is twice that of a single cell.

The BMS board helps to keep the battery pack healthy by making sure each group of cells charges and discharges evenly. It protects the batteries from issues like overcharging, deep

discharge, and excessive current, while keeping the voltage stable across all three groups. This setup makes the power supply reliable and efficient, making it well-suited for IoT-based monitoring systems and other electronic projects.

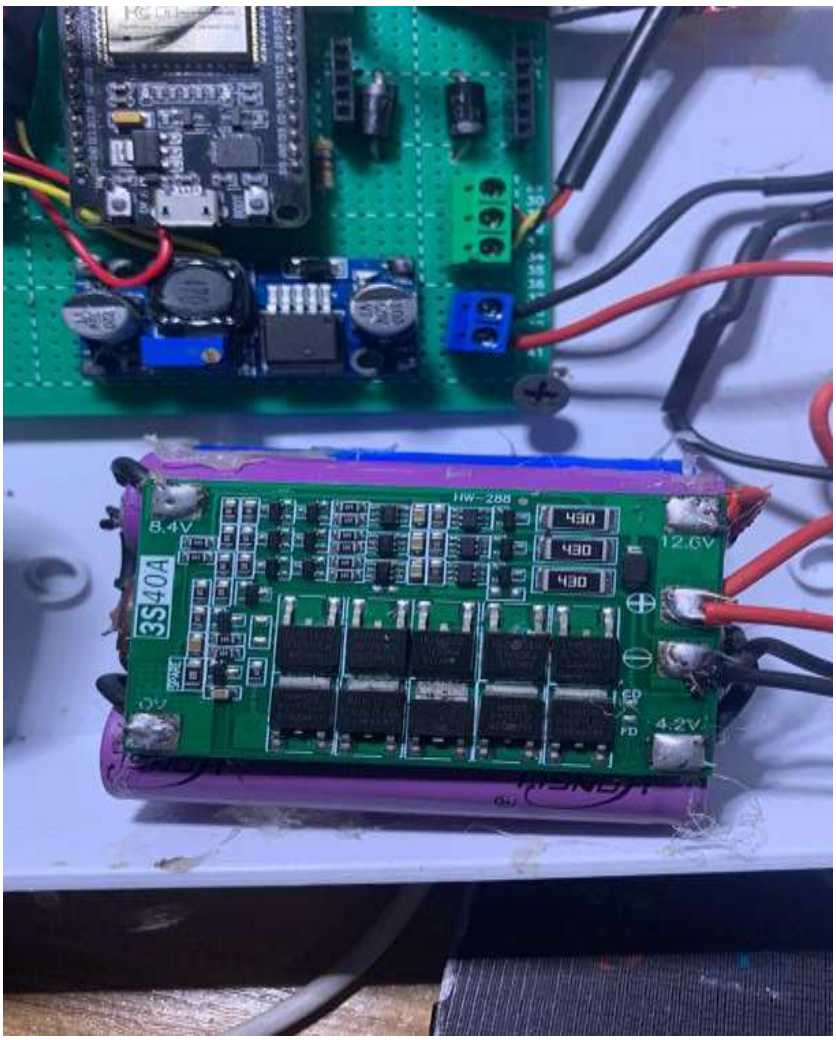


Fig 3.11: Implementation of the BMS with the Lithium battery

3.4.6. LCD DISPLAY

The LCD (Liquid Crystal Display) module is used to provide real-time visual feedback of sensor data such as pH, temperature, turbidity, and TDS. It acts as an output component that allows the user to easily monitor water quality parameters on-site. The LCD (Liquid Crystal Display) module is used to provide real-time visual feedback of sensor data such as pH,

temperature, turbidity, and TDS. It acts as an output component that allows the user to easily monitor water quality parameters on-site.

The LCD is interfaced with the ESP32 microcontroller using the I²C communication protocol, which reduces the number of connection pins required compared to a parallel interface. This setup simplifies circuit wiring and ensures stable data transmission between the microcontroller and the display.

The LCD is interfaced with the ESP32 microcontroller using the I²C communication protocol, which reduces the number of connection pins required compared to a parallel interface. This setup simplifies circuit wiring and ensures stable data transmission between the microcontroller and the display.

Fig 3.12: 20 x 4 LCD display

Fig 3.13: I²C Communication Protocol

The figure above represents the I²C communication protocol, Its primary function is to reduce the number of wires needed to connect a standard character LCD (20x4 display) to a microcontroller (in this case an ESP32 Microcontroller).

3.4.7. TEMPERATURE SENSOR

Temperature sensor is one of the key parts of water quality monitoring system. It quantifies the thermal conditions of water which has a direct and indirect effect on many other water quality parameters that include pH, dissolved oxygen (DO), turbidity, and electrical conductivity (EC). A digital temperature sensor (DS18B20) is a device employed in this project to measure the temperature of water in real time and in greater accuracy.

The sensor works by converting changes in voltage or resistance depending on temperature as digital signals that are processed by the ESP32 microcontroller. The obtained temperature is then utilized and used as an independent parameter as well as a correction factor to other sensors, particularly the pH and TDS sensors, the readings of which are temperature-dependent.

Fig 3.14: Temperature sensor.

Correlation between temperature and water quality.

Among the most vital environmental parameters that impact the chemical, physical and biological properties of water is water temperature. Its fluctuation gives a good data of the general health and quality of water and the ecosystem. The following are the essential relations:

- pH Variation:

PH of water is usually decreasing with the rise of temperature since the ionization of the water molecules is heightened by heat. Thus, pH readings can be achieved with temperature compensation.

- Dissolved Oxygen (DO):

A higher rise in temperatures of water reduces oxygen solubility. Hot water contains low oxygen compared to cold water, which may have adverse consequences on the aquatic lives. Water temperature is therefore one of the indicators of low water quality.

- TDS and Conductivity:

As temperature rises, the electrical conductivity of water also rises as ions move more quickly at higher temperatures. Therefore, when temperatures are high, TDS values can be higher when the values are not compensated.

- Microbial Activity:

Temperature has an effect on the speed of biological and chemical reactions in water. An increase in temperatures enhances the rate of decomposition and growth of microbes, and this may cause water contamination**.**

3.4.8 LM2596 BULK CONVERTER

The LM2596 buck converter is a DC–DC step-down voltage regulator designed to provide a stable and controlled output voltage for electronic systems. In this study, the converter is employed to reduce a higher input voltage of 12V to lower voltage levels required for the proper operation of the ESP32 microcontroller and the water quality sensing units.

The LM2596 is suggested for this project because it provides a stable and efficient power supply to all components of the water quality monitoring system. It ensures reliable voltage regulation and minimizes power loss during operation, which is essential for maintaining consistent performance across the ESP32 microcontroller and the various water quality sensors. Its ability to step down the input voltage from 12V to 5V makes it ideal for powering the ESP32 and sensors without generating excessive heat or causing instability.

Additionally, the LM2596 module is compact, cost-effective, and easy to integrate onto the system's circuit board, making it well-suited for embedded applications like this IoT-based water quality monitoring system. Its long-term stability, low energy consumption, and high conversion efficiency make it particularly advantageous for continuous environmental

monitoring, where reliability and power efficiency are critical. This ensures that the system remains functional over extended periods, even in remote or resource-limited areas where stable power sources may not always be available.

Key Specifications:

- Input Voltage: 4V – 40V DC
- Output Voltage: Adjustable (1.23V – 37V DC)
- Output Current: Up to 3A
- Conversion Efficiency: Up to 92%
- Switching Frequency: 150 kHz

Fig 3.15: Lm2596 IC (Bulk Converter)

3.5. CIRCUIT DESIGN AND ASSEMBLY

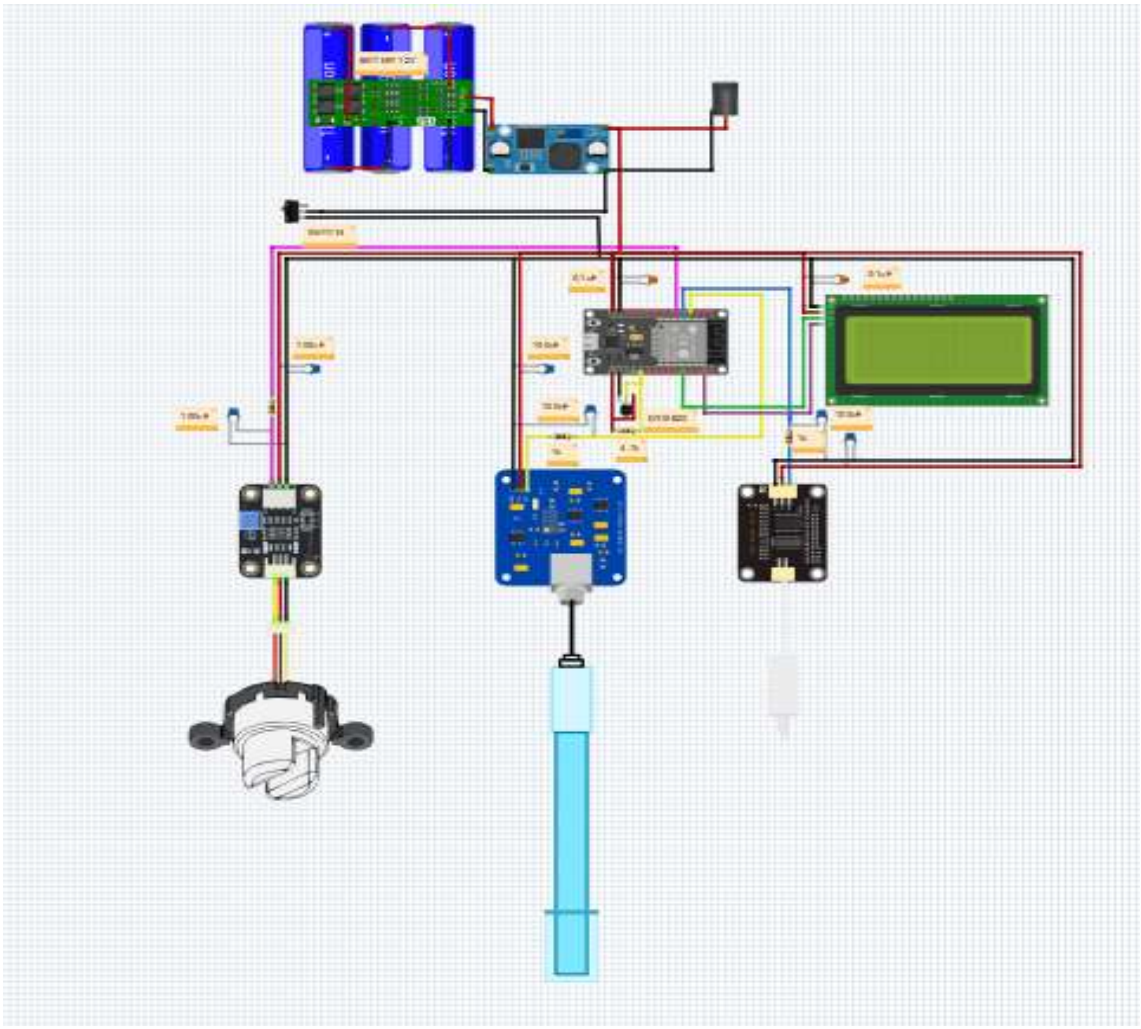


Fig 3.16: Circuit Design

3.5.1 SUMMARY OF HARDWARE SPECIFICATIONS

Table 3.1: Summary of Hardware Specifications.

Component	Model	Key Specification	Function
Microcontroller	ESP32	Dual-core, Wi-Fi enabled	Central processing
TDS Sensor	DFRobot TDS Meter	0–1000 ppm	Water purity
pH Sensor	Analog pH Meter	0–14 pH	Acidity level
Turbidity Sensor	SEN0189	0–4.5V analog output	Water clarity
Temperature Sensor	DS18B20	-55°C to +125°C	Temperature compensation

3.6 SOFTWARE DESIGN AND PROGRAMMING

The software design execution involved programming the microcontroller of ESP32 using the C/C++ language with the Arduino IDE. The code acts for all system functions such as data acquisition, data processing, data presentation, and wireless communication.

In the development process, several libraries were utilized not only to ease but also to improve the overall system with the following features:

- **WiFi.h** – to set up and keep the Wi-Fi link going between the ESP32 and the local network.
- **HTTPClient.h** – to take care of the communication of data to the custom web server through the use of HTTP POST requests.

- **LiquidCrystal_I2C.h** – to manage the 20×4 LCD so that the measured parameters can be displayed in real-time.

```
#include <WiFi.h>
#include <WebServer.h>
#include <DNSServer.h>
#include <LiquidCrystal_I2C.h>
#include <FS.h>
#include <SPIFFS.h>
#include <SD.h>
#include <HardwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoOTA.h>
```

Fig:3.17: Arduino IDE Libraries

The figure above shows, the libraries used together with the Arduino IDE environment.

The program starts off with the initialization of all the sensors and the network modules, then goes on to acquire the sensor data, calibrate, and do temperature compensation. The processed data are formatted, shown on the LCD screen, and then sent to the custom web server via the built-in Wi-Fi interface of the ESP32.

The webpage was built with HTML, CSS, and MySQL to handle the reception, storage, and display of the data that has been transmitted. The server script is set to receive HTTP POST requests from the ESP32, log each reading set into a database, and update a user dashboard with real-time numbers for pH, TDS, turbidity, and temperature. Moreover, users can download historical data in CSV format through the webpage for further analysis.

3.7 CALIBRATION AND TESTING PROCEDURES

Calibration ensures the reliability and precision of sensor readings.

- **pH Sensor:** Calibrated using buffer solutions of pH 4.0, 7.0, and 10.0.
- **TDS Sensor:** Calibrated with distilled water (0 ppm) and a 500 ppm saline solution.
- **Turbidity Sensor:** Verified using formazin turbidity standards.
- **Temperature Sensor:** Compared against a digital thermometer for accuracy.

Multiple readings were averaged to reduce noise and improve stability. The calibration constants obtained were integrated into the ESP32 firmware.

3.8 WEBPAGE CONFIGURATION

A custom web server was developed using HTML, PHP, and MySQL hosted locally for testing. The server receives sensor data from ESP32 using HTTP requests, stores it in a database, and displays it on a responsive dashboard. Real-time plots of pH, TDS, turbidity, and temperature enable visual trend analysis.

3.9 SUMMARY

This chapter presented the methodology adopted for the design, programming, and calibration of the IoT-based water quality monitoring system. The proposed approach combines both hardware and software components into an integrated, cost-effective, and efficient system capable of performing real-time water quality monitoring and web-based data visualization.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULTS

4.1 INTRODUCTION

This chapter talks about the implementation of IoT water quality monitoring system, describing both the hardware and software implementation. It also covers system testing, result interpretation and performance analysis. The main goal of this chapter is to demonstrate how the proposed design was translated into a functional prototype capable of measuring, displaying and transmitting water quality data to a custom webpage for display and monitoring. The development process involved connecting the sensors (pH, TDS, Temperature and turbidity) to the ESP32 microcontroller, configuring the LCD for local data display, and linking the system to a custom webpage for real-time monitoring. The implementation also focused on ensuring accurate data acquisition, reliable interaction between the microcontroller and the cloud, and effective visualization of the measured parameters. Each stage of the implementation was carried out systematically to ensure the design meets the project's objectives of real-time measurement, display, and cloud-based monitoring of water quality parameters.

Furthermore, This chapter delivers an in-depth explanation of how the design concept was transformed into a working prototype capable of performing both sensing and communication tasks autonomously. It discusses the steps taken to assemble the hardware, develop and upload the control program using the Arduino IDE, and configure the webpage platform to receive, store, and display sensor data. In addition, it includes the testing and analysis of system performance, describing how the readings were verified, the response time of the sensors, and the accuracy of the transmitted data. The overall objective is to demonstrate that the developed system can effectively measure, process, and transmit water quality data in real

time, thereby offering a cost-effective and efficient solution for environmental monitoring and water resource management.

4.2 SYSTEM IMPLEMENTATION.

The system block diagram above depicts system implementation of the Quality Water Testing Machine. This design has been based on the ESP32 microcontroller, which is the central processing unit. With two CPUs and in-built SPIFFS flash storage, the ESP32 controls the process of data collection, processing and communication between different parts of the system.

The section of input sensors consists of four broad sensors:

A. A PH meter (analog) that measures the acidity or alkalinity of water,
B. A turbidity sensor to measure the clearness and haziness of water,
C. A TDS (Total Dissolved Solides) meter (analog) to measure the purity of water, and
D. A temperature sensor (DS18B20) to measure the temperature of the water.

These sensors transmit their values via the ADC/GPIO pins of ESP32 through which the information is than processed and logged. The Wi-Fi Access Point (AP) module that comes with the ESP32 provides the ability to connect wirelessly and allow the display and recording and updating of data remotely. It also supports Over-The-Air (OTA) updates so as to easily maintain the system and upgrade.

The output and interface component has various data display and accessibility features. The local display is a 20x4 IC LCD display where the readings are displayed, and a built-in web server has the capability to display real-time data on any Wi-Fi-capable device, including a smartphone, tablet, or laptop. Moreover, water quality logs may be downloaded and as CSV format, which means that all data that has been recorded can be stored, analyzed, or shared to conduct follow-up research.

In general, this implementation offers a strong, scalable, and end user friendly water monitoring platform that can be used to both display locally and access remotely, which is appropriate to IoT based environmental monitoring schemes.

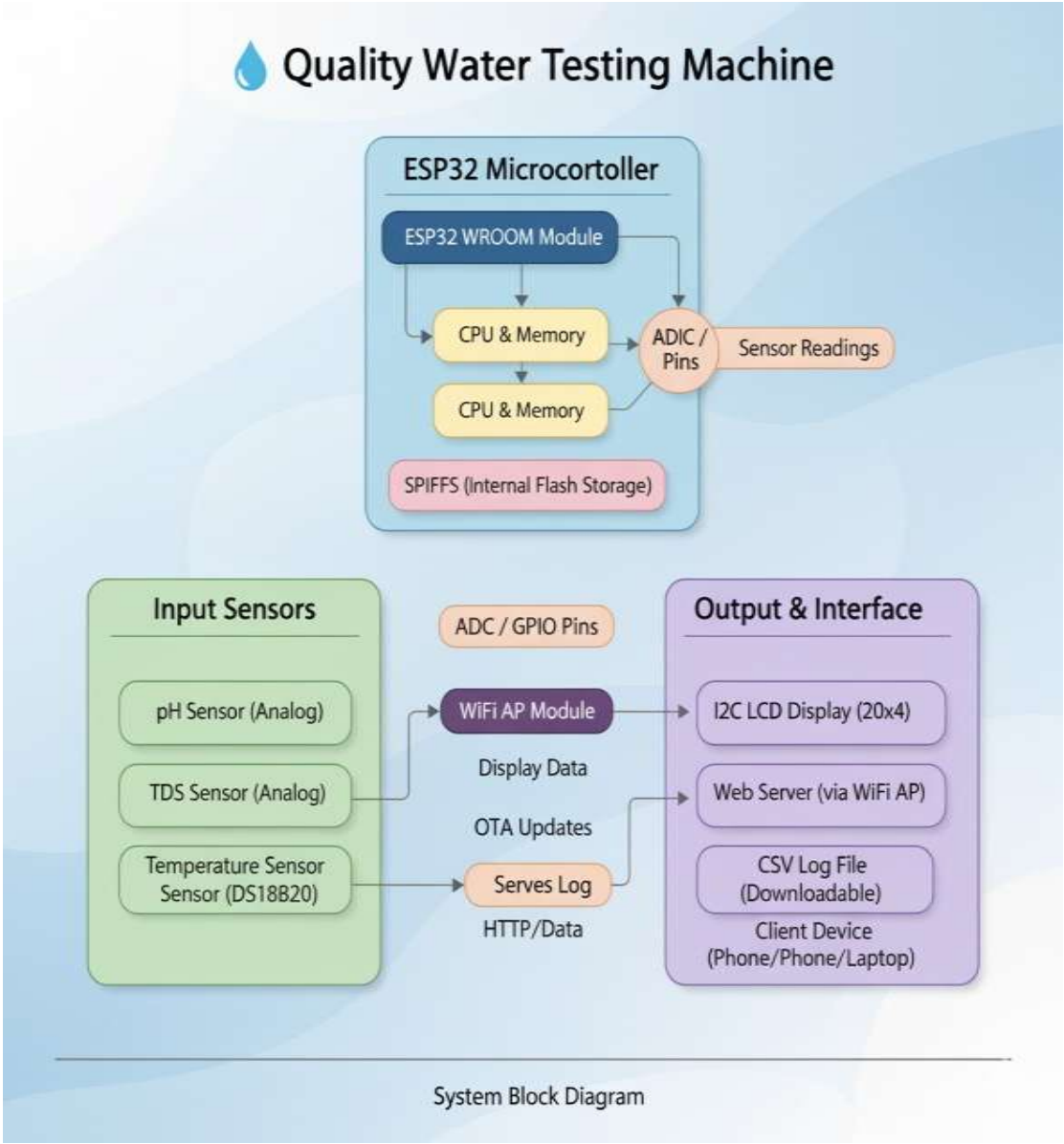


Fig: 4.1: System Block Diagram

4.3 SCALABILITY AND ADAPTABLITY

The developed IoT-based water quality monitoring system exhibits a high degree of scalability and adaptability, both in its hardware configuration and software architecture. Scalability refers to the system's ability to expand in functionality or capacity without significant redesign, while adaptability describes how effectively the system can operate under different environmental or application conditions.

The modular nature of the hardware design makes it possible to incorporate additional sensors beyond the existing pH, turbidity, and Total Dissolved Solids (TDS) sensors. This includes the feasibility of integrating other water quality sensors such as dissolved oxygen (DO), or conductivity sensors, which can further enhance the system's accuracy and comprehensiveness in assessing water conditions. The ESP32 microcontroller supports this scalability due to its multiple input/output pins, sufficient processing capability, and wireless connectivity, allowing seamless communication between new sensors and the cloud platform without overloading the system.

In terms of adaptability, the system is designed to function effectively under varying environmental and operational conditions. It can be powered from multiple sources, including rechargeable batteries and DC adapters, with the LM2596 buck converter providing voltage stability. The integration of Wi-Fi communication enables remote monitoring and data visualization through the customized webpage platform, allowing users to access real-time data from different locations. Moreover, the system can be easily deployed in diverse contexts such as households, rural communities, agricultural fields, and small industries where water quality monitoring is essential.

Overall, the system's scalability and adaptability make it a flexible, future-ready solution for water quality management. Its capacity to accommodate additional sensors like dissolved oxygen or ensures that it can evolve into a more comprehensive monitoring tool over time. This enhances its relevance for a wide range of applications while maintaining affordability, reliability, and ease of use in both urban and resource-limited settings.

4.4 COST-EFFECTIVENESS

One of the key strengths of the developed IoT-based water quality monitoring system is its cost-effectiveness, achieved through the careful selection of affordable and readily available components without compromising performance. The use of the ESP32 microcontroller significantly reduces overall system cost since it integrates both Wi-Fi and Bluetooth functionalities, eliminating the need for additional communication modules. This single-board solution supports multiple sensor inputs and cloud connectivity, offering advanced features at a lower cost compared to conventional setups that require multiple external components.

The sensors used in the system for measuring pH**,** turbidity, and Total Dissolved Solids (TDS**)** are low-cost and non-chemical, yet capable of providing sufficiently accurate readings for general water quality assessment. These sensors require minimal maintenance and calibration compared to laboratory-grade electrochemical or optical sensors, which are typically more expensive and less practical for widespread use, especially in rural or developing regions.

The LM2596 buck converter further contributes to cost-effectiveness by ensuring efficient power management, allowing the system to operate reliably on inexpensive power sources such as rechargeable batteries or low-voltage adapters. This minimizes operational costs and makes the system suitable for continuous or remote monitoring where access to grid electricity may be limited.

In addition, the use of open-source software tools such as the Arduino IDE for programming and customized webpage for cloud data visualization eliminates the need for costly proprietary software licenses. The online dashboard provides free real-time data storage and analysis, enabling effective monitoring without additional infrastructure costs.

Overall, the cost-effectiveness of this system makes it highly suitable for large-scale deployment in communities, schools, or local water management agencies. By offering a low-cost, real-time, and user-friendly solution, the system provides a practical alternative to traditional laboratory testing, promoting accessibility and sustainability in water quality monitoring, particularly in resource-constrained environments.

4.5 USER FRIENDLY INTERFACE

The developed IoT-based water quality monitoring system features a user-friendly interface designed to ensure simplicity, accessibility, and efficiency in data presentation. The interface was implemented through a customized web application built using HTML**,** CSS**,** and SQL, providing an intuitive platform for users to view and interpret real-time water quality data collected from the system.

The web interface serves as the central access point for users to monitor parameters such as pH, turbidity, and Total Dissolved Solids (TDS)**,** which are transmitted from the ESP32 microcontroller to the database. The use of HTML provides the structural layout of the webpage, defining how data and text are arranged, while CSS enhances the appearance and responsiveness of the interface, ensuring that it is visually clear and adaptable across devices such as smartphones, tablets, and computers. The integration of SQL enables efficient data storage, retrieval, and management, allowing the system to maintain a historical record of readings that users can easily access for analysis and comparison.

The webpage was designed with a clean and organized layout, featuring clearly labeled sections and real-time updating charts or tables to make interpretation straightforward even for non-technical users. This promotes usability and ensures that community members, students, or local authorities can readily understand the water quality conditions without needing specialized knowledge. The system also allows automatic data refresh and synchronization, reducing the need for manual input or configuration.

Overall, the user-friendly interface enhances the practicality of the system by bridging the gap between data acquisition and interpretation. By combining functionality with simplicity, the customized web platform ensures that the water quality monitoring process remains accessible, interactive, and efficient. This design approach aligns with the project's objective of creating a cost-effective and easy-to-use IoT-based solution that can be adopted by individuals and communities with minimal technical training.

4.6 DISCUSSION OF RESULTS

The developed IoT-based Water Quality Testing Machine was successfully tested for its ability to measure and display real-time water quality parameters, including pH, Total Dissolved Solids (TDS), and Turbidity, through a responsive web interface. The system interface, accessible via the local IP address 192.168.4.1, displayed sensor readings with corresponding visual indicators and interpretive status messages, providing users with clear and immediate feedback on water quality.

During testing, the pH level recorded was 7.17, which falls within the World Health Organization (WHO) recommended range of 6.5 to 8.5 for safe drinking water. This indicates that the tested water sample is neutral and suitable for consumption, demonstrating the accuracy and proper calibration of the pH sensor. The color-coded bar and "OK" status

indicator further confirm the system's ability to interpret sensor data effectively for user comprehension.

The TDS (Total Dissolved Solids) reading displayed a value of 0.0 ppm, which is significantly below the WHO guideline limit of ≤ 500 ppm. This suggests that the sample used was either distilled or deionized water, containing no measurable dissolved minerals or impurities. The result validates the sensor's sensitivity and responsiveness in detecting dissolved solid content within expected environmental ranges.

Similarly, the turbidity reading was recorded as 0 NTU, indicating that the water sample was clear and free from suspended particles or impurities. This shows that the turbidity sensor is capable of detecting even minor changes in water clarity and confirms its proper functionality after calibration.

The visual dashboard not only presented the numerical readings but also provided a user-friendly interface that allows the data to be downloaded in CSV format for record-keeping and further analysis. This demonstrates the successful integration of IoT functionalities, enabling both real-time monitoring and data management.

Overall, the results indicate that the system performs effectively in acquiring, processing, and displaying real-time water quality data. The readings obtained fall within acceptable standards for potable water, confirming the accuracy, reliability, and operational success of the developed IoT-based monitoring prototype. The combination of low-cost sensors and a wireless data interface proves to be a practical and efficient solution for continuous environmental monitoring.

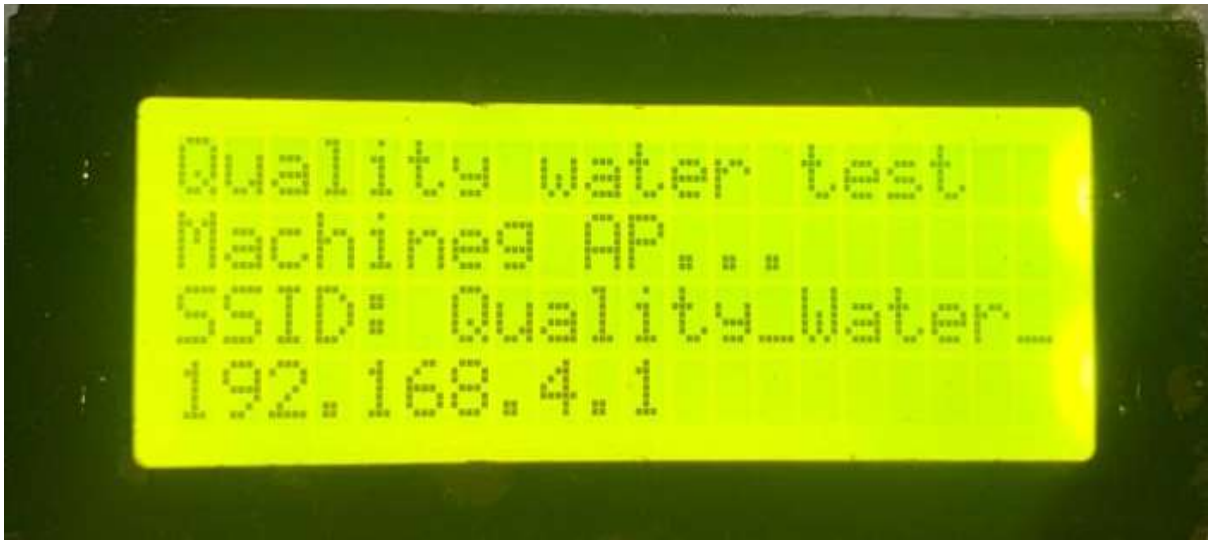


Fig 4.2: LCD Display When Turned On

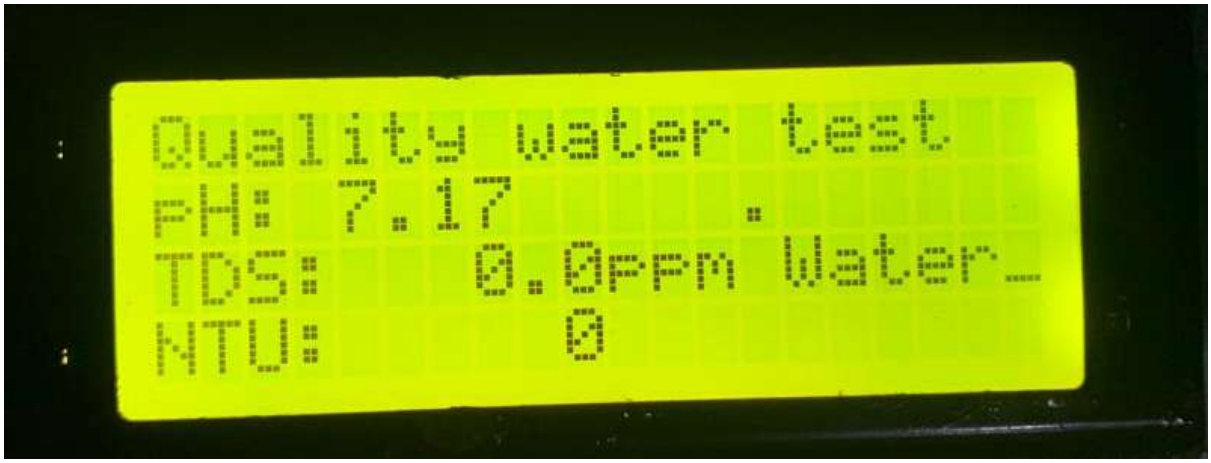


Fig 4.3: LCD Display Of Tested Water

Fig4.4: Display of tested water solution on the Webserver

Fig 4.5: Display of The Water Testing System

4.7 PERFORMANCE EVALUATION

The performance evaluation of the developed IoT-based water quality monitoring system was carried out to assess its functionality, efficiency, and reliability based on the performance of its individual components and the overall system operation. Each hardware and software component was tested to ensure accurate data acquisition, stable communication, and consistent power delivery during real-time monitoring.

The ESP32 microcontroller performed effectively as the central processing and communication unit. Its built-in Wi-Fi capability allowed seamless data transmission from the sensors to the online database without requiring external network modules. The microcontroller demonstrated efficient multitasking performance, simultaneously handling sensor readings, LCD updates, and data uploads with minimal latency. Its low power consumption and ability to maintain stable connectivity contributed to the overall energy efficiency of the system.

The sensor**s** used for measuring pH, turbidity, and Total Dissolved Solids (TDS) exhibited reliable operation within their respective calibration ranges. The pH sensor responded quickly to changes in water acidity or alkalinity, while the turbidity sensor effectively detected variations in water clarity caused by suspended particles. Similarly, the TDS sensor provided consistent readings for dissolved impurities, with results that correlated well with reference measurements. Although the sensors are low-cost types, they produced sufficiently accurate and stable outputs for non-laboratory, real-time monitoring applications.

The LM2596 buck converter ensured a stable and regulated power supply throughout the operation of the system. It efficiently stepped down the 12V input to 5V, which powered the ESP32 and sensors without overheating or voltage drops. This contributed to the system's

overall stability, allowing continuous operation over extended periods, which is essential for environmental monitoring applications.

The 20×4 LCD display performed well in providing real-time visual feedback of the measured parameters. Its clear and bright display ensured that users could easily read data in both indoor and outdoor conditions. The use of I²C communication reduced wiring complexity and enhanced the system's responsiveness.

The customized web interface, built using HTML**,** CSS**,** and SQL, also proved efficient in displaying live water quality data remotely. The interface was responsive and easily accessible across different devices, enabling users to visualize and interpret water quality conditions in real time. The SQL database successfully stored and retrieved historical readings, allowing for trend analysis and comparison over time.

Overall, the system demonstrated excellent performance in terms of accuracy, response time, data transmission reliability, and power efficiency. The integration of affordable components such as the ESP32, LM2596, and non-chemical sensors achieved a balance between cost and performance. The combination of stable hardware operation and a user-friendly web interface confirms that the system meets its design objectives of providing a low-cost**,** reliable**,** and real-time IoT-based solution for water quality monitoring

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

This project successfully designed and implemented an IoT-based water quality monitoring system capable of monitoring essential water quality parameters in real time. The developed system incorporated sensors for the measurement of pH, turbidity, total dissolved solids (TDS), and temperature, which are important parameters used in assessing water quality and determining its suitability for domestic and environmental applications.

The integration of the ESP32 microcontroller served as the core control unit, efficiently acquiring sensor data, processing it, displaying the results on a 20×4 LCD, and transmitting the readings to a custom web server for online monitoring. The web interface enabled users to visualize real-time data and historical trends, providing a flexible and accessible means of water quality evaluation.

Through a combination of hardware design, embedded programming, and cloud integration, the system demonstrated high reliability, stability, and accuracy during testing.

Compared to traditional laboratory-based water analysis, the developed system offers a low-cost, portable, and automated solution suitable for household, educational, and community-level applications. It successfully bridges the gap between manual testing and modern digital monitoring by leveraging IoT technology for real-time environmental data collection and decision-making.

Overall, the project achieved its objectives of designing, implementing, and validating a reliable IoT-based system for monitoring water quality. It demonstrates how embedded systems and wireless communication can be combined to create affordable and scalable solutions for public health and environmental sustainability.

5.2 RECOMMENDATIONS

In light of the successful design and implementation of this system, the following recommendations can be made to improve its functionality and potential for large-scale deployment.

5.2.1 Enhanced Sensor Calibration:

Regular calibration should be performed to maintain measurement accuracy, particularly when the device is used in varying water sources or environmental conditions.

5.2.2 Integration of Additional Sensors:

Future versions can include sensors for dissolved oxygen (DO), electrical conductivity (EC), or chemical contaminants such as nitrates or heavy metals to provide a more comprehensive assessment of water quality.

5.2.3 Improved Power Management:

Incorporating solar panels or advanced power-efficient modes can extend system runtime, making it more suitable for continuous field deployment in remote areas.

5.2.4 Data Security and Backup:

Implementing encryption and automatic database backups on the web server will enhance the integrity and security of collected data during online transmission and storage.

5.2.5 Mobile Application Development:

Developing a complementary mobile app that syncs with the web server will increase accessibility and user convenience, allowing users to receive alerts and notifications about water quality status in real time.

5.2.6 Industrial and Community Deployment:

The system can be adapted for larger-scale use in municipal water systems, aquaculture, and industrial wastewater management, where continuous monitoring is critical.

5.2.7 Machine Learning Integration:

With further development, machine learning algorithms can be used to analyze sensor data trends and predict potential contamination events before they occur.

REFERNCES

AlMetwally, S. A. H., Hassan, M. K., & Mourad, M. H. (2020a). Real Time Internet of Things (IoT) Based Water Quality Management System. *Procedia CIRP*, *91*, 478–485. https://doi.org/10.1016/j.procir.2020.03.107

AlMetwally, S. A. H., Hassan, M. K., & Mourad, M. H. (2020b). Real Time Internet of Things (IoT) Based Water Quality Management System. *Procedia CIRP*, *91*, 478–485. https://doi.org/10.1016/j.procir.2020.03.107

Bogdan, R., Paliuc, C., Crisan-Vida, M., Nimara, S., & Barmayoun, D. (2023a). Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas. *Sensors*, *23*(8), 3919. https://doi.org/10.3390/s23083919

Bogdan, R., Paliuc, C., Crisan-Vida, M., Nimara, S., & Barmayoun, D. (2023b). Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas. *Sensors*, *23*(8), 3919. https://doi.org/10.3390/s23083919

Cheng, Y. L., & Yu, A. W. (2003). ELECTROLYTES | Water–Electrolyte Balance. In *Encyclopedia of Food Sciences and Nutrition* (pp. 2039–2047). Elsevier. https://doi.org/10.1016/B0-12-227055-X/00395-3

Jan, F., Min-Allah, N., & Düştegör, D. (2021a). IoT Based Smart Water Quality Monitoring: Recent Techniques, Trends and Challenges for Domestic Applications. *Water*, *13*(13), 1729. https://doi.org/10.3390/w13131729

Jan, F., Min-Allah, N., & Düştegör, D. (2021b). IoT Based Smart Water Quality Monitoring: Recent Techniques, Trends and Challenges for Domestic Applications. *Water*, *13*(13), 1729. https://doi.org/10.3390/w13131729

Miller, M., Kisiel, A., Cembrowska-Lech, D., Durlik, I., & Miller, T. (2023). IoT in Water Quality Monitoring—Are We Really Here? *Sensors*, *23*(2), 960. https://doi.org/10.3390/s23020960

Mukhopadhyay, S. C., & Mason, A. (Eds.). (2013). *Smart Sensors for Real-Time Water Quality Monitoring* (Vol. 4). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-37006-9

Organization, W. H. (2002). *Guidelines for Drinking-Water Qualit: Addendum Microbiological Agents in Drinking-water* (2nd ed). World Health Organization.

Sanya, W. M., Alawi, M. A., & Eugenio, I. (2022). Design and development of Smart Water Quality Monitoring System Using IoT. *International Journal of Advances in Scientific Research and Engineering*, *08*(03), 01–13. https://doi.org/10.31695/ijasre.2022.8.3.1

Sehrawat, D., & Gill, N. S. (2019a). Smart Sensors: Analysis of Different Types of IoT Sensors. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 523–528. https://doi.org/10.1109/icoei.2019.8862778

Sehrawat, D., & Gill, N. S. (2019b). Smart Sensors: Analysis of Different Types of IoT Sensors. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 523–528. https://doi.org/10.1109/icoei.2019.8862778

Sithole, M. P. P., Nwulu, N. I., & Dogo, E. M. (2019). Dataset for a wireless sensor network based drinking-water quality monitoring and notification system. *Data in Brief*, *27*, 104813. https://doi.org/10.1016/j.dib.2019.104813

Sugiharto, W. H., Susanto, H., & Prasetijo, A. B. (2023). Real-Time Water Quality Assessment via IoT: Monitoring pH, TDS, Temperature, and Turbidity. *Ingénierie Des Systèmes d Information*, *28*(4), 823–831. https://doi.org/10.18280/isi.280403

Taru, Y. K., & Karwankar, A. (2017). Water monitoring system using arduino with labview. *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 416–419. https://doi.org/10.1109/ICCMC.2017.8282722

Yaroshenko, I., Kirsanov, D., Marjanovic, M., Lieberzeit, P. A., Korostynska, O., Mason, A., Frau, I., & Legin, A. (2020a). Real-Time Water Quality Monitoring with Chemical Sensors. *Sensors*, *20*(12), 3432. https://doi.org/10.3390/s20123432

Yaroshenko, I., Kirsanov, D., Marjanovic, M., Lieberzeit, P. A., Korostynska, O., Mason, A., Frau, I., & Legin, A. (2020b). Real-Time Water Quality Monitoring with Chemical Sensors. *Sensors*, *20*(12), 3432. https://doi.org/10.3390/s20123432

Zainurin, S. N., Wan Ismail, W. Z., Mahamud, S. N. I., Ismail, I., Jamaludin, J., Ariffin, K. N. Z., & Wan Ahmad Kamil, W. M. (2022). Advancements in Monitoring Water Quality Based on Various Sensing Methods: A Systematic Review. *International Journal of Environmental Research and Public Health, 19*(21), 14080. https://doi.org/10.3390/ijerph192114080

APPENDIX

This appendix section contains the complete source code used in the development and implementation of a water quality monitoring system.

The Arduino IDE was used as the software for programming. CSS, HTM, SQL C/C++ are the programming languages used.

LIBRARY IMPORTATION C++

```
#include <WiFi.h>
#include <WebServer.h>
#include <DNSServer.h>
#include <LiquidCrystal_I2C.h>
#include <FS.h>
#include <SPIFFS.h>
#include <SD.h>
#include <HardwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h> //added sensor (Temp sensor)
#include <ArduinoOTA.h>
```

```
/*****************************************************
 * Quality water testing Machine – corrected
 * (applies TDS/NTU dry-offset -> zero when dry,
 *  fixes FS/variable issues, uses thingspeakApiKey)
 *****************************************************/

#include <WiFi.h>
#include <WebServer.h>
#include <DNSServer.h>
#include <LiquidCrystal_I2C.h>
#include <FS.h>
#include <SPIFFS.h>
#include <SD.h>
#include <HardwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoOTA.h>

// ========= USER CONFIG =========
const char* AP_SSID     = "Quality_Water_Machine";
const char* AP_PASSWORD = "machine123"; // 8+ chars

// Storage (SD preferred, SPIFFS fallback)
#define SD_CS_PIN   5
#define MAX_RAM_BUFFERED_LINES  10

// LCD (20x4 I2C)
#define LCD_ADDR 0x27
#define LCD_COLS 20
#define LCD_ROWS 4

// Analog pins (ESP32 ADC1 only)
#define PH_PIN      34
#define TDS_PIN     35
#define TURBID_PIN  32
```

```
36
37  // DS18B20 temperature pin (for real TDS compensation)
38  #define ONEWIRE_PIN  4
39
40  // ADC settings
41  #define ADC_ATTEN ADC_11db
42  const int SAMPLES_PER_READ = 15;
43
44  // Cadence
45  const unsigned long READ_INTERVAL_MS = 800;     // read sensors
46  const unsigned long LCD_INTERVAL_MS  = 600;     // refresh LCD
47  const unsigned long LOG_INTERVAL_MS  = 15000;   // log + uplink (>= 15s ThingSpeak)
48
49  // pH model (linear: pH = slope * V + offset) – calibrate with pH 4 & 7 buffers
50  float pH_slope  = -5.70f;
51  float pH_offset = 21.34f;
52
53  // TDS compensation & factor
54  float TDS_K = 1.0f; // cell constant factor
55
56  // 🔧 Dry sensor offsets (baseline when not submerged)
57  const float OFFSET_TDS = 58.5f;   // ppm when dry (measured)
58  const float OFFSET_NTU = 3002.0f; // NTU when dry (measured)
59
60  // Smoothing (exponential)
61  const float ALPHA_PH   = 0.18f;
62  const float ALPHA_TDS  = 0.20f;
63  const float ALPHA_NTU  = 0.22f;
64  const float ALPHA_TEMP = 0.15f;
65
66  // JSON output: use null for missing values
67  #define JSON_NULLS 1
68
```

```
// SIM800L Pins/UART
HardwareSerial sim800(1);
#define SIM800_TX 17  // ESP32 TX -> SIM800L RX
#define SIM800_RX 16  // ESP32 RX -> SIM800L TX

// Airtel APN
const char APN[] = "airtelgprs.com";

// ThingSpeak
const char THINGSPEAK_HOST[] = "api.thingspeak.com";
String thingspeakApiKey = "NN34L4LOUG8K0SO7"; // replace with yours

// Optional MQTT over GPRS (no TLS). Disabled by default.
#define USE_MQTT false
const char* MQTT_HOST = "test.mosquitto.org";
const uint16_t MQTT_PORT = 1883;
const char* MQTT_CLIENTID = "water-quality-esp32";
const char* MQTT_TOPIC = "lab/water/telemetry";

// ============================

WebServer server(80);
DNSServer dnsServer;
LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);
IPAddress apIP;

OneWire oneWire(ONEWIRE_PIN);
DallasTemperature dallas(&oneWire);

String csvPath = "/water.csv";
bool useSD = false;  // set after init

// Globals (raw volts)
float gv_ph_v = NAN, gv_tds_v = NAN, gv_turb_v = NAN;

```

```
103
104  // Globals (computed & smoothed)
105  float g_ph = NAN, g_tds_ppm = NAN, g_ntu = NAN, g_tempC = 25.0f;
106
107  // Timers
108  unsigned long tLastRead=0, tLastLCD=0, tLastLog=0;
109
110  // ========== Median helper ==========
111  int medianOfArray(int *arr, int n){
112    for(int i=1;i<n;i++){int k=arr[i],j=i-1;while(j>=0&&arr[j]>k){arr[j+1]=arr[j];j--;}arr[j+1]=k;}
113    return (n%2)? arr[n/2] : (arr[n/2-1]+arr[n/2])/2;
114  }
115
116  int readMilliVoltsMedian(uint8_t pin){
117    int buf[SAMPLES_PER_READ];
118    for(int i=0;i<SAMPLES_PER_READ;i++){ buf[i]=analogReadMilliVolts(pin); delay(3); }
119    return medianOfArray(buf, SAMPLES_PER_READ);
120  }
121
122  // ========== ADC setup ==========
123  void setupADC(){
124    analogReadResolution(12);
125    analogSetAttenuation(ADC_ATTEN);
126    analogSetPinAttenuation(PH_PIN, ADC_ATTEN);
127    analogSetPinAttenuation(TDS_PIN, ADC_ATTEN);
128    analogSetPinAttenuation(TURBID_PIN, ADC_ATTEN);
129  }
```

```
130
131  // ========== Sensor math ==========
132  float computePH(float v){ return pH_slope * v + pH_offset; }
133  float computeTDSppm(float v, float tempC){
134    // Temperature compensation (linear around 25C)
135    float k = 1.0f + 0.02f * (tempC - 25.0f);
136    float vc = v / k; // compensated volts
137    // Approximate EC (uS/cm) from voltage (sensor specific; tune with your probe)
138    float ec = 133.42f*powf(vc,3) - 255.86f*powf(vc,2) + 857.39f*vc;
139    if(ec < 0) ec = 0;
140    return ec * TDS_K * 0.5f; // ppm estimate
141  }
142  float computeNTU(float v){
143    float ntu = -1120.4f*v*v + 5742.3f*v - 4352.9f;
144    if(ntu < 0) ntu = 0;
145    return ntu;
146  }
147
148  static inline float smooth(float prev, float val, float a){
149    if(isnan(prev)) return val; // initialize
150    return a*val + (1-a)*prev;
151  }
152
153  // ========== DS18B20 ==========
154  float readWaterTemp(){
155    dallas.requestTemperatures();
156    float t = dallas.getTempCByIndex(0);
157    if(t > -55.0f && t < 125.0f) return t;
158    return NAN;
159  }
```

```
}

// ========== Read sensors (non-blocking cadence managed in loop) ==========
void readSensors(){
  // Raw voltages (millivolts -> volts)
  gv_ph_v   = readMilliVoltsMedian(PH_PIN)/1000.0f;
  gv_tds_v  = readMilliVoltsMedian(TDS_PIN)/1000.0f;
  gv_turb_v = readMilliVoltsMedian(TURBID_PIN)/1000.0f;

  // Temperature
  float tC = readWaterTemp();
  if(!isnan(tC)) g_tempC = smooth(g_tempC, tC, ALPHA_TEMP);

  // Computations (raw)
  float ph_raw  = computePH(gv_ph_v);
  float tds_raw = computeTDSppm(gv_tds_v, g_tempC);
  float ntu_raw = computeNTU(gv_turb_v);

  // 🔧 Apply baseline offset correction (make dry sensors read 0)
  float tds_corr = tds_raw;
  float ntu_corr = ntu_raw;
  if(tds_corr > OFFSET_TDS) tds_corr -= OFFSET_TDS; else tds_corr = 0.0f;
  if(ntu_corr > OFFSET_NTU) ntu_corr -= OFFSET_NTU; else ntu_corr = 0.0f;

  // Smoothing (we keep ph smoothing on raw pH, tds/ntu on corrected values)
  g_ph      = smooth(g_ph, ph_raw,  ALPHA_PH);
  g_tds_ppm = smooth(g_tds_ppm, tds_corr, ALPHA_TDS);
  g_ntu     = smooth(g_ntu, ntu_corr, ALPHA_NTU);
}
```

```
// ========== Storage ==========
void ensureCSVHeader(){
  if(useSD){
    if(!SD.exists(csvPath)){
      File f = SD.open(csvPath, FILE_WRITE);
      if(f){ f.println("seconds,ph,tds_ppm,ntu,tempC"); f.close(); }
    }
  }else{
    if(!SPIFFS.exists(csvPath)){
      File f = SPIFFS.open(csvPath, FILE_WRITE);
      if(f){ f.println("seconds,ph,tds_ppm,ntu,tempC"); f.close(); }
    }
  }
}

void initFS(){
  // Try SD first
  if(SD.begin(SD_CS_PIN)){
    useSD = true;
    Serial.println("SD ready");
  }else{
    Serial.println("SD init failed. Falling back to SPIFFS.");
    useSD = false;
    if(!SPIFFS.begin(true)){
      Serial.println("SPIFFS init failed!");
    }else{
      Serial.println("SPIFFS ready");
    }
  }
  ensureCSVHeader();
}
```

```
// RAM buffer to reduce write wear
String lineBuffer;
int bufferedLines = 0;

void appendLogLine(const String& line){
  lineBuffer += line;
  bufferedLines++;
  if(bufferedLines >= MAX_RAM_BUFFERED_LINES){
    if(useSD){
      File f = SD.open(csvPath, FILE_APPEND);
      if(f){ f.print(lineBuffer); f.close(); }
    }else{
      File f = SPIFFS.open(csvPath, FILE_APPEND);
      if(f){ f.print(lineBuffer); f.close(); }
    }
    lineBuffer = "";
    bufferedLines = 0;
  }
}

void flushLog(){
  if(lineBuffer.length() == 0) return;
  if(useSD){
    File f = SD.open(csvPath, FILE_APPEND);
    if(f){ f.print(lineBuffer); f.close(); }
  }else{
    File f = SPIFFS.open(csvPath, FILE_APPEND);
    if(f){ f.print(lineBuffer); f.close(); }
  }
  lineBuffer = "";
  bufferedLines = 0;
}
```

```
}

void appendLog(){
  String line =
    String(millis()/1000) + "," +
    (isnan(g_ph)      ? String("") : String(g_ph,2)) + "," +
    (isnan(g_tds_ppm) ? String("") : String(g_tds_ppm,1)) + "," +
    (isnan(g_ntu)     ? String("") : String(g_ntu,0)) + "," +
    (isnan(g_tempC)   ? String("") : String(g_tempC,1)) + "\n";

  appendLogLine(line);
}

void handleDownload(){
  flushLog();
  if(useSD){
    if(!SD.exists(csvPath)){ server.send(404,"text/plain","No log file"); return; }
    File f = SD.open(csvPath);
    server.streamFile(f, "text/csv"); f.close();
  }else{
    if(!SPIFFS.exists(csvPath)){ server.send(404,"text/plain","No log file"); return; }
    File f = SPIFFS.open(csvPath);
    server.streamFile(f, "text/csv"); f.close();
  }
}
```

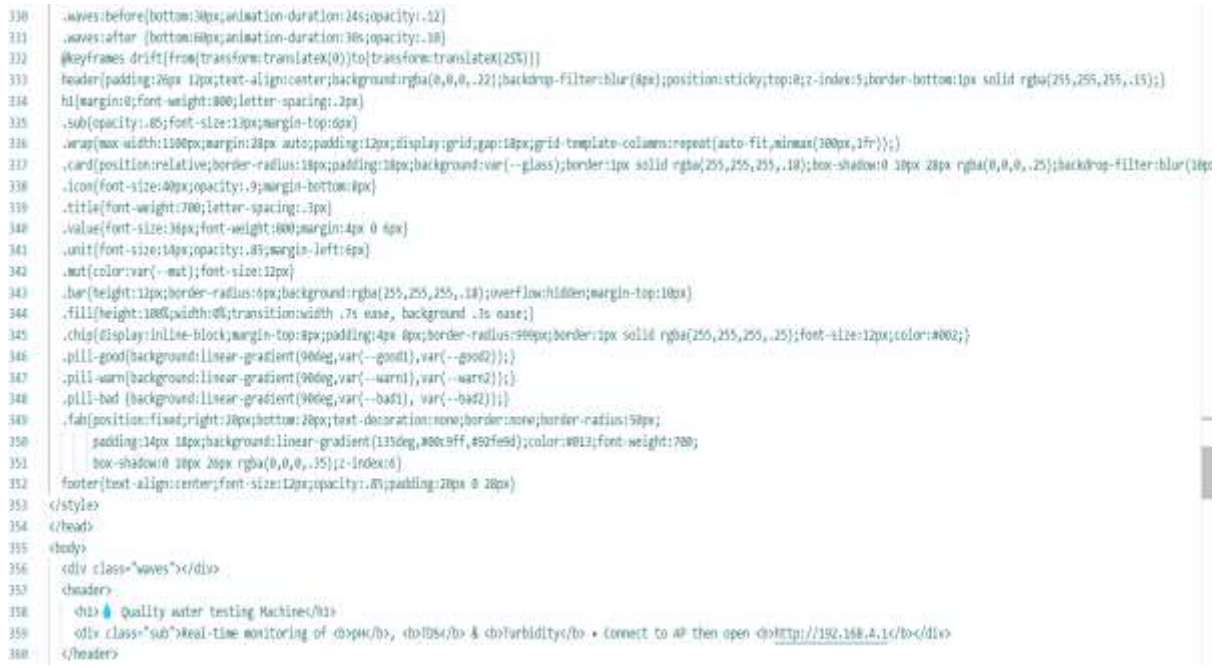
```
// ========== LCD ==========
void lcdShow(){
  lcd.setCursor(0,0); lcd.print("Quality water test ");
  lcd.setCursor(0,1);
  lcd.print("pH: ");
  if(isnan(g_ph)) lcd.print("--.--     ");
  else { char buf[12]; dtostrf(g_ph,4,2,buf); lcd.print(buf); lcd.print("     "); }

  lcd.setCursor(0,2);
  lcd.print("TDS:");
  if(isnan(g_tds_ppm)) lcd.print("---.-ppm  ");
  else { char buf[12]; dtostrf(g_tds_ppm,6,1,buf); lcd.print(buf); lcd.print("ppm "); }

  lcd.setCursor(0,3);
  lcd.print("NTU:");
  if(isnan(g_ntu)) lcd.print("---     ");
  else { char buf[12]; dtostrf(g_ntu,6,0,buf); lcd.print(buf); lcd.print("   "); }
}

// ========== Captive portal & UI ==========
void captiveRedirect() {
  String url = "http://" + apIP.toString() + "/";
  server.sendHeader("Location", url, true);
  server.send(302, "text/plain", "Redirecting to portal...");
}
```

```
// (kept HTML as you had it — omitted here for brevity in code snippet)
const char HTML[] PROGMEM = R"HTML(
<!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>Quality water testing Machine</title>
<style>
  :root{
    --bg1:#003d73; --bg2:#0077be; --glass:rgba(255,255,255,0.12);
    --good1:#2af598; --good2:#009efd;
    --warn1:#fbd72b; --warn2:#f9484a;
    --bad1:#ff6a00; --bad2:#ee0979;
    --text:#f5fbff; --mut:#bcd7f0; --chip:#0ff;
  }
  *{box-sizing:border-box;font-family:system-ui,Segoe UI,Roboto,Arial}
  body{margin:0;color:var(--text);background:linear-gradient(135deg,var(--bg1),var(--bg2));min-height:100vh;overflow-x:hidden;}
  .waves,.waves:before,.waves:after{
    position:fixed;left:-50%;right:-50%;height:140px;bottom:-20px;content:"";opacity:.18;
    background:
      radial-gradient(50px 50px at 20px 50px, rgba(255,255,255,.6), transparent 60%),
      radial-gradient(60px 60px at 120px 30px, rgba(255,255,255,.6), transparent 60%),
      radial-gradient(40px 40px at 220px 60px, rgba(255,255,255,.6), transparent 60%),
      radial-gradient(70px 70px at 340px 40px, rgba(255,255,255,.6), transparent 60%);
    filter: blur(6px);
    animation: drift 18s linear infinite;
  }
```



```
  .waves:before{bottom:30px;animation-duration:14s;opacity:.12}
  .waves:after {bottom:60px;animation-duration:30s;opacity:.18}
  @keyframes drift{from{transform:translateX(0)}to{transform:translateX(25%)}}
  header{padding:26px 12px;text-align:center;background:rgba(0,0,0,.22);backdrop-filter:blur(8px);position:sticky;top:0;z-index:5;border-bottom:1px solid rgba(255,255,255,.15);}
  h1{margin:0;font-weight:800;letter-spacing:.2px}
  .sub{opacity:.85;font-size:13px;margin-top:6px}
  .wrap{max-width:1100px;margin:28px auto;padding:12px;display:grid;gap:18px;grid-template-columns:repeat(auto-fit,minmax(300px,1fr));}
  .card{position:relative;border-radius:18px;padding:18px;background:var(--glass);border:1px solid rgba(255,255,255,.18);box-shadow:0 10px 28px rgba(0,0,0,.25);backdrop-filter:blur(10p
  .icon{font-size:40px;opacity:.9;margin-bottom:8px}
  .title{font-weight:700;letter-spacing:.3px}
  .value{font-size:36px;font-weight:800;margin:4px 0 6px}
  .unit{font-size:14px;opacity:.85;margin-left:6px}
  .mut{color:var(--mut);font-size:12px}
  .bar{height:12px;border-radius:6px;background:rgba(255,255,255,.18);overflow:hidden;margin-top:10px}
  .fill{height:100%;width:0%;transition:width .7s ease, background .3s ease;}
  .chip{display:inline-block;margin-top:8px;padding:4px 8px;border-radius:999px;border:1px solid rgba(255,255,255,.25);font-size:12px;color:#002;}
  .pill-good{background:linear-gradient(90deg,var(--good1),var(--good2));}
  .pill-warn{background:linear-gradient(90deg,var(--warn1),var(--warn2));}
  .pill-bad {background:linear-gradient(90deg,var(--bad1), var(--bad2));}
  .fab{position:fixed;right:18px;bottom:28px;text-decoration:none;border:none;border-radius:50px;
      padding:14px 18px;background:linear-gradient(135deg,#00c9ff,#92fe9d);color:#013;font-weight:700;
      box-shadow:0 10px 26px rgba(0,0,0,.35);z-index:6}
  footer{text-align:center;font-size:12px;opacity:.8%;padding:20px 0 28px}
</style>
</head>
<body>
  <div class="waves"></div>
  <header>
    <h1>💧 Quality water testing Machine</h1>
    <div class="sub">Real-time monitoring of <b>pH</b>, <b>TDS</b> & <b>Turbidity</b> • Connect to AP then open <b>http://192.168.4.1</b></div>
  </header>
```

```
<div class="wrap">
  <div class="card" id="phCard">
    <div class="icon">🧪</div>
    <div class="title">pH Level</div>
    <div class="value"><span id="ph">--</span><span class="unit">pH</span></div>
    <div class="bar"><div id="phBar" class="fill"></div></div>
    <div class="mut">Ideal range for drinking water: 6.5 – 8.5</div>
    <span id="phPill" class="chip pill-good" style="display:none">OK</span>
  </div>

  <div class="card" id="tdsCard">
    <div class="icon">🧊</div>
    <div class="title">TDS (Total Dissolved Solids)</div>
    <div class="value"><span id="tds">--</span><span class="unit">ppm</span></div>
    <div class="bar"><div id="tdsBar" class="fill"></div></div>
    <div class="mut">WHO guideline ≤ 500 ppm</div>
    <span id="tdsPill" class="chip pill-good" style="display:none">OK</span>
  </div>

  <div class="card" id="ntuCard">
    <div class="icon">🌫️</div>
    <div class="title">Turbidity</div>
    <div class="value"><span id="ntu">--</span><span class="unit">NTU</span></div>
    <div class="bar"><div id="ntuBar" class="fill"></div></div>
    <div class="mut">Clear water typically &lt; 5 NTU</div>
    <span id="ntuPill" class="chip pill-good" style="display:none">OK</span>
  </div>
</div>

<a class="fab" href="/download">⬇ Download CSV Log</a>
<footer>© 2025 Quality Water Testing Machine</footer>
```

```
<script>
function clamp(x,a,b){return Math.max(a,Math.min(b,x));}
function setBar(el, pct, state){
  el.style.width = clamp(pct,0,100) + "%";
  if(state==="good")   el.style.background="linear-gradient(90deg,var(--good1),var(--good2))";
  else if(state==="warn") el.style.background="linear-gradient(90deg,var(--warn1),var(--warn2))";
  else el.style.background="linear-gradient(90deg,var(--bad1),var(--bad2))";
}
function setPill(el, state){
  el.style.display="inline-block";
  el.className = "chip " + (state==="good"?"pill-good":state==="warn"?"pill-warn":"pill-bad");
  el.textContent = state==="good" ? "OK" : (state==="warn" ? "CHECK" : "ALERT");
}
function phState(v){
  if(isNaN(v)) return "warn";
  if(v>=6.5 && v<=8.5) return "good";
  if((v>=6.0 && v<6.5) || (v>8.5 && v<=9.0)) return "warn";
  return "bad";
}
function tdsState(v){
  if(isNaN(v)) return "warn";
  if(v<=500) return "good";
  if(v<=1000) return "warn";
  return "bad";
}
function ntuState(v){
  if(isNaN(v)) return "warn";
  if(v<=5) return "good";
  if(v<=50) return "warn";
  return "bad";
}
```

```
}
async function poll(){
  try{
    const r = await fetch('/data');
    const j = await r.json();
    const ph = j.ph===null? NaN : +j.ph;
    const tds= j.tds===null? NaN : +j.tds;
    const ntu= j.ntu===null? NaN : +j.ntu;
    document.getElementById('ph').textContent  = isNaN(ph)? "--" : ph.toFixed(2);
    document.getElementById('tds').textContent = isNaN(tds)? "--" : tds.toFixed(1);
    document.getElementById('ntu').textContent = isNaN(ntu)? "--" : ntu.toFixed(0);
    setBar(document.getElementById('phBar'),  (ph/14)*100, phState(ph));
    setBar(document.getElementById('tdsBar'), clamp((isNaN(tds)?0:tds)/1000*100,0,100), tdsState(tds));
    setBar(document.getElementById('ntuBar'), clamp((isNaN(ntu)?0:ntu)/100*100,0,100), ntuState(ntu));
    setPill(document.getElementById('phPill'),  phState(ph));
    setPill(document.getElementById('tdsPill'), tdsState(tds));
    setPill(document.getElementById('ntuPill'), ntuState(ntu));
  }catch(e){}
  setTimeout(poll, 1000);
}
poll();
</script>
</body></html>
)HTML";
```

```
448
449  // ---------- Routes ----------
450  void handleRoot(){ server.send(200,"text/html",HTML); }
451  void handleData(){
452    String out = "{";
453  #if JSON_NULLS
454    out += "\"ph\":"   + (isnan(g_ph)? String("null") : String(g_ph,2)) + ",";
455    out += "\"tds\":"  + (isnan(g_tds_ppm)? String("null") : String(g_tds_ppm,1)) + ",";
456    out += "\"ntu\":"  + (isnan(g_ntu)? String("null") : String(g_ntu,0)) + ",";
457    out += "\"tempC\":"+ (isnan(g_tempC)? String("null") : String(g_tempC,1)) + ",";
458  #else
459    out += "\"ph\":"   + String(isnan(g_ph)? -1.0:g_ph, 2) + ",";
460    out += "\"tds\":"  + String(isnan(g_tds_ppm)? -1.0:g_tds_ppm, 1) + ",";
461    out += "\"ntu\":"  + String(isnan(g_ntu)? -1.0:g_ntu, 0) + ",";
462    out += "\"tempC\":"+ String(isnan(g_tempC)? -1.0:g_tempC,1) + ",";
463  #endif
464    out += "\"uptime\":"+ String(millis()/1000);
465    out += "}";
466    server.send(200,"application/json",out);
467  }
468  void handleGen204(){ captiveRedirect(); } // Android
469  void handleHotspot(){ captiveRedirect(); } // Apple
470  void handleMsNCSI(){ captiveRedirect(); }  // Windows
471  void handleFavicon(){ server.send(204); }
472  void handleRobots(){ server.send(200,"text/plain","User-agent: *\nDisallow: /\n"); }
473
```

```
174  // ---------- SIM800L Upload ----------
175  void sendToThingSpeak(float ph, float tds, float ntu) {
176    // Build GET path and send via SIM800L AT commands (blocking-ish simple sequence)
177    String path = "/update?api_key=" + thingspeakApiKey +
178                  "&field1=" + String(ph,2) +
179                  "&field2=" + String(tds,1) +
180                  "&field3=" + String(ntu,0);
181    String fullURL = "http://" + String(THINGSPEAK_HOST) + path;
182
183    sim800.println("AT+HTTPTERM"); delay(200);
184    sim800.println("AT+HTTPINIT"); delay(200);
185    sim800.println("AT+HTTPPARA=\"CID\",1"); delay(200);
186    sim800.println("AT+HTTPPARA=\"URL\",\"" + fullURL + "\""); delay(500);
187    sim800.println("AT+HTTPACTION=0"); delay(6000); // wait for GET to complete
188    sim800.println("AT+HTTPREAD"); delay(1000);
189    sim800.println("AT+HTTPTERM");
190  }
```

```
491
492 // ---------- Setup ----------
493 void setup(){
494   Serial.begin(115200);
495   delay(200);
496
497   setupADC();
498   initFS();
499
500   // LCD
501   lcd.init(); lcd.backlight(); lcd.clear();
502   lcd.setCursor(0,0); lcd.print("Quality water test");
503   lcd.setCursor(0,1); lcd.print("Starting AP...");
504
505   // AP + captive portal
506   WiFi.mode(WIFI_AP);
507   WiFi.softAP(AP_SSID, AP_PASSWORD);
508   apIP = WiFi.softAPIP();
509   Serial.print("AP IP: "); Serial.println(apIP);
510
511   lcd.setCursor(0,2); lcd.print("SSID: "); lcd.print(AP_SSID);
512   lcd.setCursor(0,3); lcd.print(apIP.toString().c_str());
513
514   dnsServer.start(53, "*", apIP);
515   server.on("/", handleRoot);
516   server.on("/index.html", handleRoot);
517   server.on("/data", handleData);
518   server.on("/download", handleDownload);
519   server.on("/generate_204", handleGen204);
520   server.on("/hotspot-detect.html", handleHotspot);
521   server.on("/ncsi.txt", handleMsNCSI);
522   server.on("/favicon.ico", handleFavicon);
523   server.on("/robots.txt", handleRobots);
524   server.onNotFound([](){ captiveRedirect(); });
525   server.begin();
```

```
526
527   // DS18B20
528   dallas.begin();
529
530   // SIM800L init
531   sim800.begin(9600, SERIAL_8N1, SIM800_RX, SIM800_TX);
532   delay(600);
533   // quick sanity ping
534   sim800.println("AT");
535
536   // OTA (works on AP network)
537   ArduinoOTA.setHostname("QualityWaterMachine");
538   ArduinoOTA.onStart([](){
539     lcd.clear(); lcd.setCursor(0,1); lcd.print("OTA updating...");
540   });
541   ArduinoOTA.onEnd([](){
542     lcd.clear(); lcd.setCursor(0,1); lcd.print("OTA done. Reboot");
543   });
544   ArduinoOTA.onError([](ota_error_t e){
545     Serial.printf("OTA Error[%u]\n", e);
546   });
547   ArduinoOTA.begin();
548 }
```

```
// ---------- Loop ----------
void loop(){
  dnsServer.processNextRequest();
  server.handleClient();
  ArduinoOTA.handle();

  unsigned long now = millis();

  if(now - tLastRead >= READ_INTERVAL_MS){
    tLastRead = now;
    readSensors();
  }
  if(now - tLastLCD >= LCD_INTERVAL_MS){
    tLastLCD = now;
    lcdShow();
  }
  if(now - tLastLog >= LOG_INTERVAL_MS){
    tLastLog = now;
    appendLog();
    flushLog();                 // make sure we persist at least every interval
    // upload current smoothed values (g_ph, g_tds_ppm, g_ntu)
    sendToThingSpeak( isnan(g_ph)? 0.0f : g_ph,
                      isnan(g_tds_ppm)? 0.0f : g_tds_ppm,
                      isnan(g_ntu)? 0.0f : g_ntu );
  }
}
```