Cape Peninsula University of Technology

Low-Cost Smart Monitoring and Control System for Water Management in Rural Wastewater Treatment.

by

Wendy Nkomo

A thesis submitted in fulfillment of the requirements for the degree.

Master of Engineering: Electrical Engineering

in the faculty of Engineering and Built Environment

at the Cape Peninsula University of Technology

Supervisor: Prof. AK Raji

Bellville
November 2024

DECLARATION

I, Wendy Nkomo, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

11-September-2024

Signed **Date**

ABSTRACT

Water pollution is a major global problem because of increased levels of drinking water contaminants. These pollutants can cause diseases that harm both humans and animals, thereby negatively impacting the environment. The study needs to focus on a system that can detect these pollutants early enough to prevent any pollution-related damage. This research aims to provide a low-cost system that is smart in monitoring and controlling the waters for real-time water quality management. As part of this study, consider temperature, turbidity, pH, electrical conductivity (EC), total dissolved solids (TDS), and rates of water flow as the key parameters measured by the sensors. Water pollution management is possible by continuous monitoring of these parameters, issuing warning signals, and recommending measures whenever pollution is present. The system monitors water quality by connecting an Arduino to sensors. The system follows simple rules: it analyses the data to identify any issues. If it detects contamination, it can automatically introduce chemicals to enhance the water's quality. Thingspeak, a cloud platform that maintains records and updates water quality reports, receives the data wirelessly. Additionally, the If This Then That (IFTTT) system immediately sends alert messages to users' phones upon detection of contamination. The conducted research yields compelling results, demonstrating the system's ability to continuously monitor water quality and initiate necessary actions. Furthermore, remote access, integration with other IoT applications, and enhanced robustness render this IoT-based solution useful for water quality management. The present research concludes that the use of Internet of Things-based water quality monitoring systems can ensure safe drinking water, and its implementation is both simple and feasible. In a way, it promotes the cause of public health as well as environmental conversation.

Keywords: Arduino IDE programming, Contaminated Water, Internet of things, Notification Alert (IFTTT), ThingSpeak, Water parameters, water quality monitoring, and control.

ACKNOWLEDGEMENTS

I wish to thank:

- I would like to sincerely thank Prof. Raji, my supervisor, for his constant support, guidance, and inspiration during my studies.

- Additionally, I would like to express my gratitude to the DEECE senior laboratory technicians, Mr. C. Wills and Mr. P. Msiza, for providing some of the components I needed during the process of developing my prototype and ensuring that the laboratory is available anytime I needed to use it.

TABLE OF CONTENTS

DECLARATION ii

ABSTRACT iii

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

CHAPTER ONE 1

INTRODUCTION 1
1.1 Research Background 1
1.2 The Motivation of the research problem 2
1.3 Problem statement 2
1.4 Research Aims and Objectives 4
1.5 Hypothesis 4
1.6 Delimitation of the research 5
1.7 Assumptions 5
1.8 Research Design and Methodology 6
1.8.1 Research Methodology Diagram 7
1.9 Thesis Structure 8
1.10 Conclusion ..... **Error! Bookmark not defined.**
CHAPTER TWO 9

LITERATURE REVIEW 9
2.1 Introduction 9
2.2 Low-Income water requirement 10
2.3 Water crisis 11
2.4 Demand of water quality 12
2.5 Water quality parameters 12
2.5.1 TDS 12
2.5.2 Electrical Conductivity 13
2.5.3 pH ..... **Error! Bookmark not defined.**
2.5.4 Turbidity 15
2.5.5 Water flow 16
2.5.6 Temperature 16
2.6 Comparison between traditional and current water quality monitoring Techniques 17
2.7 Various Water Monitoring Methods 25
2.7.1 Method to measure the suitability of water. 26
2.7.2 Water quality index value and mathematical of water quality parameters 26
2.8 Parameters Measured with Smart Quality Online Monitoring systems. 30
2.9 IoT water quality Management 31
2.10 IoT based water quality Monitoring 31
2.11 Feed water characteristics and water treatment unit processes 33
2.12 Smart system for Monitoring and detecting water quality 34
2.13 Application of water monitoring System 35
2.14 Technology scan on the application of microprocessors in water quality monitoring 37
2.15 Building blocks of Smart water quality online monitoring 37
2.16 Internet of Things and ThingSpeak integration 38
2.17 Conclusions ..... **Error! Bookmark not defined.**
CHAPTER THREE 40

RESEARCH METHODOLOGY 40
3.1 Overview 40
3.2 System design 40
3.2.1 Arduino Mega Microcontroller 40
3.2.2 Interfacing the turbidity sensor to Arduino 42
3.2.3 Interfacing the pH sensor to Arduino 43
3.2.4 Interface the Electrical Conductivity sensor to Arduino 44
3.2.5 Interfacing Total dissolve solids to Arduino 45
3.2.6 Flow sensor 45
3.2.7 Interfacing the Temperature Sensor to Arduino 46
3.2.8 Breadboard, Wire and Resistor 48

3.2.9 Relay 48
3.2.10 DC pump 49
3.2.11 Pipe 50
3.2.12 Lithium batteries and 18650 Holder with a Charger 50
3.3 Interface Ethernet shield to Arduino 53
3.4 Flow Chart of the system 53
3.5 Control system 55
3.6 System Architecture 55
3.7 Raw data collection 58
3.8 Data Pre-processing 58
3.9 Conclusion .... **Error! Bookmark not defined.**
CHAPTER FOUR 60
ENGINEERING SYSTEM DESIGN 60
4.1 Overview 60
4.2 Arduino IDE Program 60
4.3 Sensor integration 62
4.3.1 Water quality sensor setup 62
4.3.2 Calibration Process of TDS 62
4.3.3 Calibration Process of Analog electrical conductivity (EC) 63
4.3.4 The BMT Temperature Probe 65
4.3.5 The process of calibrating the turbidity sensor 65
4.3.6 The process of calibrating the pH sensor 67
4.3.7 Water flow 68
4.4 Cloud integration 69
4.5 IFTTT Integrated with ThingSpeak. 71
4.6 Data Terminal (PC) for Smart monitoring and control system 74
4.7 User friendly interface 76
4.8 Final Project 78
4.9 Conclusion .... **Error! Bookmark not defined.**
CHAPTER FIVE 82
VALIDATION AND DISCUSSIONS OF RESULTS 82
5.1 Introduction 82
5.2 EC Sensor: Analysis test 82
5.2.1 EC Sensor: Practical test 83
5.2.2 Different solutions 84
5.2.3 Temperature and EC indicators and Results on IDE monitor and ThingSpeak day one of testing 84
5.2.4 EC and Temperature Results on Arduino IDE monitor and ThingSpeak on the second day of testing 87
5.3 TDS sensor's analysis and testing functionality 88
5.3.1 TDS Solutions 89
5.3.2 TDS Level indicator captured on IDE monitor 90
5.4 pH sensor Analysis test 92
5.4.1 pH sensor Practical test 92
5.4.2 pH Level indicators and ThingSpeak graphs 94
5.5 Turbidity sensor Analysis and practical test 96
5.6 Water sensor: analysis 100
5.6.1 YF-S201 water flow sensor: Practical test 100
5.6.2 The Process of Testing Flow water sensor 101
5.7 The results analysis for overall system performance 102
5.8 Conclusion .... **Error! Bookmark not defined.**
CHAPTER SIX 107
DELIVERABLES, CONCLUSIONS AND RECOMMENDATIONS 107
6.1 Introduction 107
6.2 Aims and objectives 107
6.3 Thesis deliverables 108
6.3.1 Chapter 2: Literature Review 108
6.3.2 Chapter 3: Research Methodology 108
6.3.3 Chapter 4: Engineering System Design 109
6.3.4 CHAPTER FIVE: Validation and Results Analysis 109
6.4 Academic and Industrial Application 110

6.5 Future Work 110
6.6 Recommendations 110
6.7 Conclusion .... **Error! Bookmark not defined.**
REFERENCES 112
APPENDICES 117

LIST OF FIGURES

Figure 1.1: Schematic diagram demonstrating factors that lead to problems of the treatment plant (V Mema, 2010) 3
Figure 1.2: Research Methodology Structure 7
Figure 2.1: Literature Review Framework 10
Figure 2.2: Environmental effects, pH ranges, and examples of pH (Science, 2021). 14
Figure 2.3: Organic and inorganic particles in all sizes contribute to the suspended solids concentration (Heger, 2017) 15
Figure 2.4: General structure of the WQI model (Uddin et al., 2021).**Error! Bookmark not defined.**
Figure 2.5: The key parameter in smart water quality monitoring (Geetha & Gouthami, 2016). 30
Figure 2.6: Water treatment unit (water, 2022). 34
Figure 2.7: Concept of online water quality and control system (Wiranto Goib et al., 2015). 35
Figure 2.8: Application of smart water monitoring systems (Geetha & Gouthami, 2016). 36
Figure 2.9: process of monitoring and control system (Prapti et al., 2022). 36
Figure 2.10: The schematic diagram of the water monitoring system(Geetha & Gouthami, 2016). 38
Figure 3.1: Arduino Microcontroller Mega 2560 42
Figure 3.2: DF Robot gravity analog turbidity sensor electronic schematic (Trevathan et al., 2020b). 42
Figure 3.3: Interfacing the Turbidity Sensor to Arduino Microcontroller (Hakim et al., 2019). 43
Figure 3.4: pH range and Interfacing pH Sensor Module to the Arduino Microcontroller (Samsudin et al., 2018) 44
Figure 3.5: Interface EC meter to Arduino (Shamsi et al., 2020). 45
Figure 3.6: Total dissolved solids connected to Arduino. 45
Figure 3.7: Water Flow Pump 46
Figure 3.8: Waterproof Temperature connected to Arduino 47
Figure 3.9: Breadboard, resistors, and wires. 48
Figure 3.10: Relay 49
Figure 3.11: DC pump 50
Figure 3.12: Pipe 50
Figure 3.13: 18650 2 Battery Shields 52
Figure 3.14: Rechargeable lithium batteries 52
Figure 3.15: Ethernet shield connected to Arduino. 53
Figure 3.16: Flowchart of the System 54
Figure 3.17: Connection of DC Pump, Relay, and batteries 55
Figure 3.18: Architect of the system 57

Figure 4.1: Arduino IDE 61
Figure 4.2: TDS Calibration Flowchart 63
Figure 4.3: Flowchart of Calibration process of EC sensor 64
Figure 4.4: Calibration of Temperature Sensor 65
Figure 4.5: Flowchart of the calibration Turbidity sensor 66
Figure 4.6: Flowchart Calibration of pH sensor 67
Figure 4.7: Flowchart of calibration of the Water Flow 68
Figure 4.8: Channel 69
Figure 4.9: Channel ID and API key 70
Figure 4.10: Include libraries (Ethernet and ThingSpeak) 70
Figure 4.11: Credentials of ThingSpeak and Internet 71
Figure 4.12: Integration of IFTTT and ThingSpeak 72
Figure 4.13: Flowchart of IFTTT integrated with ThingSpeak 74
Figure 4.14: Data Terminal (Laptop) 76
Figure 4.15: ThingSpeak App integrated with information from IFTTT 77
Figure 4.16: Complete setup of IFTTT 77
Figure 4.17: Final prototype side view 79
Figure 4.18: Final prototype Top View 80
Figure 5.1: Water, Saline, and buffer solutions for EC sensor 84
Figure 5.2: Results of Tap water or bottled water on the IDE monitor 85
Figure 5.3: Results of the Saline water and Temperature 85
Figure 5.4: Graph Visualisation on ThingSpeak for tap water and Saline solution 86
Figure 5.5: Temperature readings captured on ThingSpeak 86
Figure 5.6: Results of the saline solution captured on Arduino IDE 87
Figure 5.7: Results of a tap water 87
Figure 5.8: Results of Saline and Tapwater on ThingSpeak 88
Figure 5.9: Results of the Temperature 88
Figure 5.10: Tap water and 0.3 grams of Table salt in distilled water 89
Figure 5.11: Results of TDS using Tap water 90
Figure 5.12: Results using 0.3 grams of Table salt in distilled water solution 90
Figure 5.13: Results of a Tap and Unacceptable Water 91
Figure 5.14: Results of Tap Water 91
Figure 5.15: Results of unacceptable water 92
Figure 5.16: Vinegar, Natural Spring water and Tapwater mix with baking soda solutions ... 93
Figure 5.17: Acidic results 94
Figure 5.18: Neutral and Alkaline 94
Figure 5.19: Results of drinkable water on the pH 95
Figure 5.20: Results of Acidic, Neutral, and Alkaline 95

Figure 5.21: Results of Neutral and alkaline.. 96
Figure 5.22: Solutions for testing Turbidity 97
Figure 5.23: Dirty water.. 97
Figure 5.24: Results of a clean turbidity value.. 97
Figure 5.25: Results of an NTU for clean water using Turbidity sensor 98
Figure 5.26: Results Voltage of a Dirty Water 98
Figure 5.27: Results of NTU Dirty water using Turbidity sensor 99
Figure 5.28: Results of a Voltage for Clean Water using a Turbidity Sensor 99
Figure 5.29: Results of a Voltage for Clean Water using a Turbidity Sensor 100
Figure 5.30: Water flow rate testing. 101
Figure 5.31: Results of Flow rate on the IDE monitor 102
Figure 5.32: Results of all Six and pump on Arduino IDE monitor 102
Figure 5.33: Results of TDS, Turbidity, Temperature, and Flowrate on ThingSpeak 103
Figure 5.34: Results of all sensors testing on ThingSpeak 104

LIST OF TABLES

Table 2.1: Review summary of Traditional and current water quality monitoring Methods.... 17
Table 2.2: Review summary of Traditional and current water quality monitoring Methods.... 23
Table 2.3: comparison of various water monitoring methods (B A Karomah, 2022) 25
Table 2.4: Water quality parameters and their acceptable ranges.. 26
Table 3.1: Specifications for ATmega2560 Arduino Microcontroller 41
Table 3.2: Specifications for 18650 holder with a charger 50
Table 3.3: Specifications for lithium batteries .... **Error! Bookmark not defined.**

TABLE OF APPENDICES

APPENDIX A: ThingSpeak and Ethernet shield(1 of 3).. 117
APPENDIX B: ThingSpeak and Ethernet shield (2 of 3)... 118
APPENDIX C: ThingSpeak and Ethernet shield(3 of 3)... 119
APPENDIX D: pH code (1 of 2).. 120
APPENDIX E: pH code (2 of 2).. 121
APPENDIX F: Flow rate measurement code.. 122
APPENDIX G: TDS sensor code (1 of 2). 123
APPENDIX H: TDS sensor code (2 of 2).. 124
APPENDIX I: EC and Temperature sensors code (1 of 4). 125
APPENDIX J: EC and Temperature sensors code (2 of 4).. 126
APPENDIX K: EC and Temperature sensors code (3 of 4). 127
APPENDIX L: EC and Temperature sensors code (4 of 4).. 128
APPENDIX M: Turbidity sensor code (1 of 3).. 129
APPENDIX N: Turbidity sensor code (2 of 3). 130
APPENDIX O: Turbidity sensor code (3 of 3). 131
APPENDIX P: Combination of sensors and DC pump and relay (1 of 7)........................... 132
APPENDIX Q: Combination of sensors and DC pump and relay (2 of 7). 133
APPENDIX R: Combination of sensors and DC pump and relay (3 of 7)........................... 134
APPENDIX S: Combination of sensors and DC pump and relay (4 of 7)........................... 135
APPENDIX T: Combination of sensors and DC pump and relay (5 of 7)........................... 136
APPENDIX U: Combination of sensors and DC pump and relay (6 of 7)........................... 137
APPENDIX V: Combination of sensors and DC pump and relay (7 of 7)........................... 138
APPENDIX W: 3D Box Design.. 138
APPENDIX X: Simulation of this Monitoring and control system using circuito................... 139
APPENDIX Y: ATmega2560 Arduino Specifications 140
APPENDIX Z: Specifications of water flow sensor from Communica 141

ABBREVIATIONS/ACRONYMS

AC	:	Alternating Current
AR	:	Autoregressive
BED	:	Binomial Event Discriminator
BOD	:	Biochemical Oxygen Demand
β-CD	:	Beta-Cyclodextrin
C	:	Celsius
CDOM	:	Chromophoric Dissolved Organic Matter
CDOM	:	Coloured Dissolved Organic Matter
COVID-19	:	Coronavirus Disease 2019
CPU	:	Central Processing Unit
CUSUM	:	Cumulative Sum
DC	:	Direct Current
DO	:	Dissolved Oxygen
EC	:	Electrical Conductivity
EGOFET	:	Electrolyte-Gated Organic Field-Effect Transistor
FUI	:	Fluorescence Unit Intensity
FDOM	:	Fluorescent Dissolved Organic Matter
GPS	:	Global Positioning System
GND	:	Ground
HTTP	:	Hypertext Transfer Protocol
IFTTT	:	If This Then That
IDE	:	Integrated Development Environment
IoT	:	Internet of Things
KH	:	Carbonate Hardness
LCD	:	Liquid Crystal Display

LED	:	Light Emitting Diode
MISO	:	Master In Slave Out
Mg/l	:	Milligrams per liter
mS/cm	:	Milli siemens per centimetre
µS/cm	:	Micro siemens per centimetre
NTU	:	Nephelometric Turbidity Units
NGS	:	Next Generation Sequencing
ORP	:	Oxidation-reduction potential
OBCPD	:	Online Bayesian Changepoint Detection
PCR	:	Polymerase Chain Reaction
PC	:	Personal Computer
PWM	:	Pulse Width Modulation
Ppm	:	Parts Per Million
ROC	:	Receiver Operating Characteristic
RS	:	Remote sensing
SDGs	:	Sustainable Development Goals
SCK	:	Serial Clock
SPI	:	Serial Peripheral Interface
SPRT	:	Sequential Probability Ratio Test
SS	:	Suspended Solids
TDS	:	Total Dissolved Solids
TSS	:	Total Suspended Solids
TN	:	Total Nitrogen
UN	:	United Nations
UNICEF	:	United Nations International Children's Emergency Fund
USEPA	:	United States Environmental Protection Agency
VCC	:	Voltage Common Collector

WQP	:	Water quality parameter
WHO	:	World Health Organization
WTP	:	Willingness to Pay
WSNs	:	Wireless Sensor Networks

GLOSSARY

Arduino: An accessible to everyone electronics platform with easily navigable hardware and software. Sensor data such as light may be received by Arduino boards, which could then use that data to produce outputs like turning on a motor.

Arduino IDE Program: The Arduino microcontroller's code is written, compiled, and uploaded using the integrated development environment (IDE). The Arduino programming language is supported by it.

Arduino Microcontroller: A small, programmable circuit board used for controlling electronic devices and systems. Arduino microcontrollers are widely used in DIY electronics projects, including those related to IoT.

Applications of Water Quality: The different practical uses of water purity data, such as in environmental monitoring, public health, water treatment, and regulatory compliance.

Control: The regulation or management of devices, systems, or processes to ensure they operate as intended.

Contaminated Water: Water that contains harmful substances like chemicals, pathogens, or pollutants, making it unsafe for human consumption or use.

DC Pump: A type of pump powered by direct current (DC), commonly used in applications where portability or battery operation is required.

Data Pre-Processing: The procedure for preparing raw data for analysis or additional processing by cleaning, converting, and organizing it into a more suitable format.

Detecting Water Quality: The procedure for evaluating and characterizing water's properties, such as its pH, turbidity, presence of pollutants, and other aspects, to establish whether it is safe to use.

Electrical Conductivity: A measurement of the electrical conductivity of water, which is correlated with the number of dissolved salts and minerals in the water.

Ethernet Shield: A module or add-on for microcontrollers like Arduino that allows them to connect to the internet or a local network via an Ethernet cable.

IFTTT (If This Then That): A web-based tool that helps users design basic automation rules or applets that link various gadgets, programs, and online services in response to triggers.

Internet of Things (IoT): A system of linked devices that can monitor, operate, and automate tasks remotely by exchanging data and communicating with one another via the internet.

IoT Water-Based Monitoring: The use of Internet of Things (IoT) technology to remotely monitor water quality, flow, and other parameters through connected sensors and devices.

Monitoring: The regular observation and recording of activities, processes, or systems to ensure they are functioning correctly or to gather data.

Modern Method: Contemporary techniques or approaches that utilize advanced technologies, often involving automation, digital tools, or new innovations.

Pollution: The presence of hazardous materials or pollution, either released into the environment or otherwise, which negatively affects ecosystems and living organisms.

pH: A scale from 0 to 14, where 7 is neutral, that indicates how acidic or alkaline a solution is. Alkalinity is indicated by greater values, and acidity is shown by lower ones.

Raw Data Collection: The process of gathering unprocessed, unfiltered data directly from sources like sensors without any manipulation or analysis.

Relay: A switch that is powered by electricity and that lets a low-power circuit manage a high-power circuit. Relays are frequently seen in control and automation systems.

Safe Drinking Water: Drinkable water that won't damage humans because it is free of dangerous pollutants and pathogens.

TDS (total dissolved solids): The total amount of dissolved materials, such as minerals, salts, and organic stuff, in water. Typically, parts per million (ppm) are used to measure it.

Temperature: The measure of the thermal energy of a substance, typically measured in degrees Celsius (°C) or Fahrenheit (°F).

ThingSpeak: An IoT analytics platform that enables users to collect, analyse, and visualize data from sensors and devices connected to the internet.

ThingSpeak and Notification Alert (IFTTT): The integration of ThingSpeak with IFTTT to create automation rules that trigger notifications or other actions based on data received from IoT devices.

Traditional Method: conventional approaches or techniques that have been used over time, often involving manual or non-digital processes.

Turbidity: The measurement of water's clarity or cloudiness, which is usually brought on by suspended particles. Elevated turbidity may be a sign of contamination or pollution.

User-Friendly Interface: A system or software interface designed to be easy for users to understand and interact with, often focusing on simplicity, clarity, and accessibility.

Water Crisis: A circumstance in which a region's supply of drinkable, clean water is insufficient to meet demand, creating a scarcity or making it more difficult to obtain clean water.

Water Demand: The quantity of water required for various uses, such as drinking, agriculture, industry, and sanitation, within a specific region or system.

Water Flow: The rate at which water moves through a system, often measured in liters per second or gallons per minute.

Water Management: The procedure for planning, supplying, assigning, and managing the best possible use of water resources in a fair, efficient, and sustainable way.

Water Parameters: parameters or measurements that are used to evaluate the quality of water, such as pH, turbidity, temperature, and TDS.

CHAPTER ONE
INTRODUCTION

1.1 Research Background

Safe water is becoming a rare resource due to increased population pollution and climate change. People's access to clean water has significantly declined because of the massive rise in industrial output worldwide, individuals moving from rural to urban places, excessive land use, and high fertilizer use in farms and marine resources. In the world, contaminated water is the cause of almost 40% of deaths (Ashbolt, 2004). Therefore, it's important to assure that everyone, in towns and rural communities, has access to water that is suitable for drinking water (Vanshika Ambati et al., 2023). Water must be safe and readily available for the general public's health because it is utilized for drinking, home chores, food production, and recreational activities. Improving the quantity and quality of water, along with improved resource management and sanitation, might spur economic expansion and significantly lower poverty (Vanshika Ambati et al., 2023).

The World Health Organization estimates that 60 billion people used properly managed drinking water services in 2022, utilizing improved on-site water supplies that were clean, readily available, and accessible when needed. the 2.2 billion individuals who, in 2022, will not have access to services that are securely run (Anon, 2024). Cholera and diarrhoea are two recognized illnesses that can spread because of tainted water and poor sanitation. According to UNICEF/WHO, 2023, 2.2 billion people have their health needlessly endangered by inadequate, poorly managed, or non-existent water quality services, while 3.5 million people lack access to clean water and sanitation.

An estimated 1 million people are thought to pass away from diarrheal sickness each year because of inadequate hand hygiene, sanitation, and drinking water quality, according to the figures. Nevertheless, 395 000 children die from diarrhoea each year, even though it is largely preventable (Who progress report 2021). in developing countries, where 2.2 billion people lack access to potable water. Only when the two factors, which are relationships between the different features and functions of water, are better understood will progress towards water security be possible. In addition to being a natural resource, water is also a good or entitlement for human use as well as an economic benefit. Three primary categories can be used to organize the issues around water insecurity: accessibility, availability, and utilization (Singh Sumer, 2022). The Internet of Things (IoT) is a popular term in the field of information technology, and it is also the wave of the future; it turns physical objects into remotes, which will allow us to monitor and control the objects in our environment (Mishra et al., 2021; Ashbolt, 2004).

1.2 The Motivation of the research problem

Water is essential for all life, and ensuring its purity is crucial. Monitoring water quality is our first line of defense against contamination, especially in water used for drinking and agriculture. Pollution poses risks to humans and animals alike. Treating waterborne diseases is expensive and contributes to a substantial portion of global death. Regular, effective monitoring of drinking water quality is therefore important to prevent diseases and protect public health. By integrating Internet of Things (IoT) technology into a water quality monitoring system, we can improve the reliability of data transmission. This reduces reliance on error-prone manual lab methods, leading to more accurate decision-making. The goal of developing smart monitoring and control systems for water quality is to optimize operations and reduce waste resulting from infectious pollution management practices.

1.3 Problem statement

It’s essential to handle water resources considerately since insufficient water safety can cause waterborne epidemics in places with ecological strain. To solve these problems, it is important to promote recycling of water and better plant management. Such measures will not only guarantee drinking water safety but also maintain river health, thus enhancing the community’s strength during climate change events (Rich et al., 2023). Eutrophication occurs because of high nutrient pollution originating from agricultural runoff and wastewater, leading to severe water quality deterioration, poisonous blue-green algae blooms (Cyanobacteria), and ultimately decreased biological diversity (Pranta et al., 2023). Everyone in South Africa has much easier access to piped drinking water now that apartheid is over. Most families in both rural and urban areas now have access to piped drinking water; the remaining destitute groups are in more remote rural areas and urban informal settlements. Even though drinking water in rural areas might not always be clean or readily available, there haven't been any more waterborne outbreaks comparable to the cholera epidemic of 2000–2001. Ongoing health problems in rural areas without access to water, however, are indicated by the case fatality rates of diarrhoea in children under five (Mbana & Sinthumule, 2024a) (Mbana & Sinthumule, 2024b).Considering this, South Africa faces severe issues with water quality, particularly in remote rural areas where most residents rely on untreated water from sources like rivers, boreholes, springs, and rainfall collecting. The high levels of fluoride contamination affecting different regions in India are severely affecting the health of the populace, especially that of children. Evidence shows that people exposed to a higher amount of fluoride over long periods have higher chances of getting oral and skeletal disorders. This makes it clear that a high number of ingested fluorides cause worse health outcomes. Additionally, as time goes by, this situation gets worse since there has been an obvious rise in frequency and severity rate of these sicknesses. Despite there being some data about it that is nationally representative, we need to comprehend how serious an issue it is for kids aged between seven and fourteen to find ways to come up with effective interventions aimed at reducing their negative impacts (Aggarwal Khushboo & Ahmad Nadeem, 2022). In This project will develop a low-cost smart monitoring and control system to improve the water quality in low-income residential areas. Finding a solution for the water issues is challenging due to the apparent inadequate management of water infrastructure by water utilities. Numerous factors contribute to this management deficiency, as illustrated in Figure 1.1 below, which highlights the persistent issue of inadequately treated wastewater and sewage discharges, along with the underlying causes (V Mema, 2010).

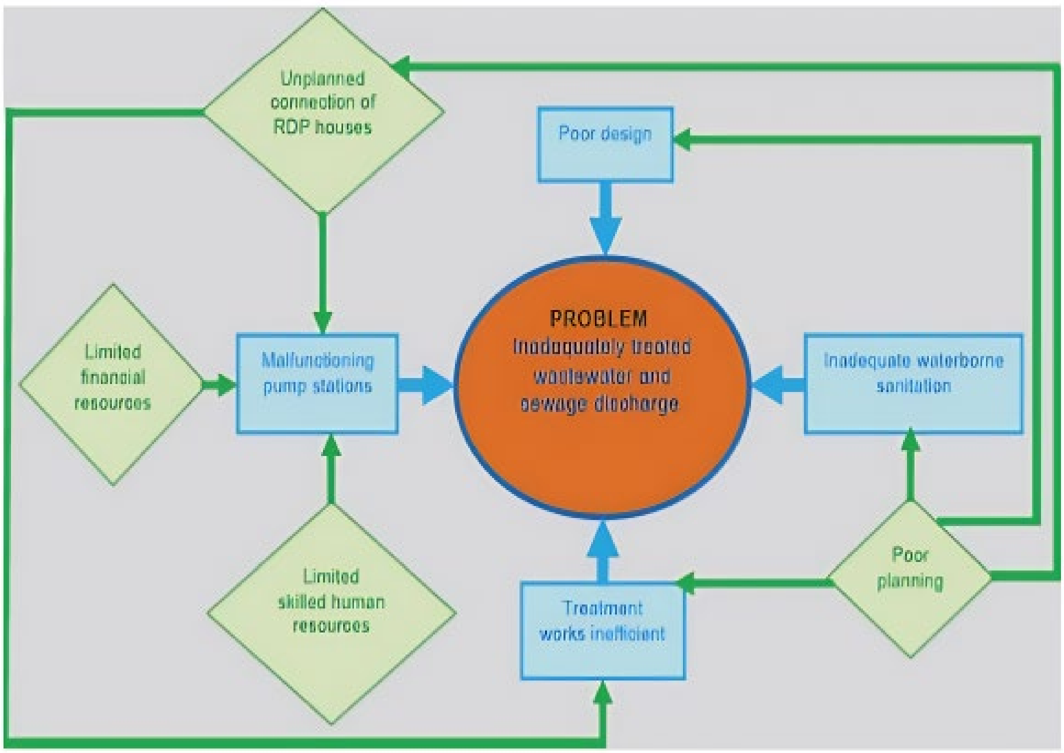


Figure 1.1: Schematic diagram demonstrating factors that lead to problem of the treatment plant (V Mema, 2010)

1.4 Research Aims and Objectives

This study builds on the understanding that ensuring clean and safe water is crucial and highlights the potential of smart monitoring systems in addressing gaps left by traditional methods.

The primary goal of this research is to develop a low-cost, smart monitoring and control system for water quality measurement. To explore the effectiveness and necessity of such a system, we set the following objectives:

- The goal of the literature review is to review 20 and more studies on Arduino water quality monitoring systems in eight weeks to find useful ideas.
- Water Quality Parameters: In eight weeks, students will learn about key water quality measurements (temperature, TDS, pH, turbidity, EC, and water flow) and their importance.
- The Monitoring and Control Model involves the development of a model that utilizes sensors, Arduino, Ethernet, Thingspeak, and FTTT to measure water quality within a timeframe of twenty-eight weeks.
- Automated Control System: Design a system to automatically treat water based on sensor readings, aiming for reliable accuracy in pump operation within eight weeks.

1.5 Hypothesis

Rural treatment plants across South Africa have successfully implemented cost-effective monitoring and control frameworks to improve wastewater quality, thereby enhancing operational effectiveness and efficiency. The Internet of Things (IoT) equips sensors with monitoring equipment to enable real-time monitoring of water quality parameters, avert pollution problems, and improve control measures.

This system's hypothesis is founded on:

- Cheap materials used in this type of monitoring reduce financial barriers for efficient water quality management.
- An Ethernet shield on an Arduino board allows for real-time water parameter monitoring.
- The process of controlling water quality relies on automated control, which includes relays and DC pumps. These components aid the automation process based on sensor readings, thereby limiting manual intervention.
- ThingSpeak reliably transmits data and monitors historical record analysis.
- It is user-friendly because it has a plain setup and simple IFTTT.

1.6 Delimitation of the research

In this study, we focus on six specific aspects of water quality monitoring: turbidity (the quantity of light that a sample of water can transmit) and total dissolved solids (the quantity of dissolved substance), temperature (which compensates with the EC sensor), pH (the acidity or alkalinity of the water), water flow (the rate at which water moves through a system), and electrical conductivity (which measures the electrical conductivity of water). An Arduino-based system with specialized water sensors measures these parameters. These parameters are measured using an Arduino-based system equipped with specialized water sensors. To send data, the system includes an Ethernet shield, controls the DC pump via relay, and uses lithium batteries for power. These components form the core of this investigation, enabling cost-effective methods for real-time monitoring and management of water quality.

The limitations of this study are necessary because they make sure that designs and functionalities of the system meet the practical needs of management settings concerning water quality. The aim is to restrict its application context to cheap, environmentally friendly implementations, including home water monitoring systems or small-scale plants focusing on purification processes. This limitation allows the design and functions of the system to be relevant in contexts related to managing water quality. By concentrating on a controlled environment, the study enables precise performance testing and evaluation, eliminating the need to navigate the complexities of wide-ranging geographic variances, all while conducting tests within clearly defined boundaries. However, while this system integrates ThingSpeak as a remote data logging and notification alerting framework, the advanced data analytic aspects only extend to basic monitoring and control operations. Finally, we are still working on conventional water chemistry analysers, so we are primarily focused on developing and testing this software.

1.7 Assumptions

The following assumptions are the results of the conducted research and its findings:

- The selected components (Arduino, Ethernet shield, water sensors, relay, DC pump, lithium batteries) are assumed to work well together without technical issues.
- The system is assumed to operate effectively in a typical indoor or sheltered outdoor environment suitable for water quality monitoring.
- The Ethernet shield and ThingSpeak are assumed to reliably transmit and store data, providing continuous monitoring of water parameters.

- The use of lithium batteries is assumed to provide sufficient and stable power for extended periods, supporting autonomous operation of the system without frequent recharging.
- The water sensors are assumed to accurately measure critical parameters such as turbidity, TDS, EC, etc., ensuring reliable water quality assessment.
- The integration notification alerts through ThingSpeak are assumed to effectively inform users about deviations from preset water quality contamination, enabling quick response and intervention.
- The project assumes that the chosen components (Arduino-based system, Ethernet shield, etc.) offer a low-cost alternative compared to traditional monitoring systems.
- It is assumed to be user-friendly interface for easy configuration, remote monitoring via ThingSpeak, and straightforward access alerts.

1.8 Research Design and Methodology

This research project aims to develop a low-cost, smart monitoring and control system for water quality. It uses components like Arduino, Ethernet shield, water sensors, relay, DC pump, and lithium batteries and connects to the ThingSpeak cloud platform for data management. Programming tasks are handled using the Arduino IDE, which helps the system to make decisions based on the collected data from the sensors.

To start, the project begins with a thorough review of existing technologies in IoT-based water quality monitoring. This review guides the selection of components and design principles for the system. Next, a prototype is built that includes Arduino to gather data from water sensors. A relay-controlled DC pump is also integrated for automatic water management.

Connecting to ThingSpeak through the Ethernet shield allows real-time monitoring and storage of data in the cloud. Simple decision-making algorithms programmed on Arduino interpret sensor readings. The system will undergo testing to ensure it works reliably across different conditions.

This methodology Aims to create a practical and low-cost solution for monitoring and managing water quality. By using accessible components and IoT technology, the project aims to improve how water quality is monitored and controlled in real-world applications.

1.8.1 Research Methodology Diagram

The goal of this project is to develop and implement a low-budget smart water pollutant control system that effectively manages water pollution in real time. This study explores the integration of various Internet of Things technologies and sensors to establish a comprehensive, 24/7 monitoring and automated control system. For the system to effectively handle various water quality parameters such as turbidity, TDS, pH, and electrical conductivity, it must require minimal human intervention. This study employs a methodology consisting of three phases to achieve its objectives. The first phase entails identifying the most pertinent technological principles from various literature to guide our understanding of the IoT and Arduino IDE statements. Secondly, we will select the components, design the system, use programming to control Arduino, perform real-time processing via cloud computing platforms, and then develop an easy-to-use user interface. Experimentation and results are the last part associated with this stage, where arranging the setup, collecting data from it, and testing performance as well as analysing output to verify its effectiveness. This structured approach combines theoretical research, practical system development, and thorough testing to create a reliable and efficient solution for water quality maintenance. The diagram in Figure 1.2 illustrates the framework of this research methodology.

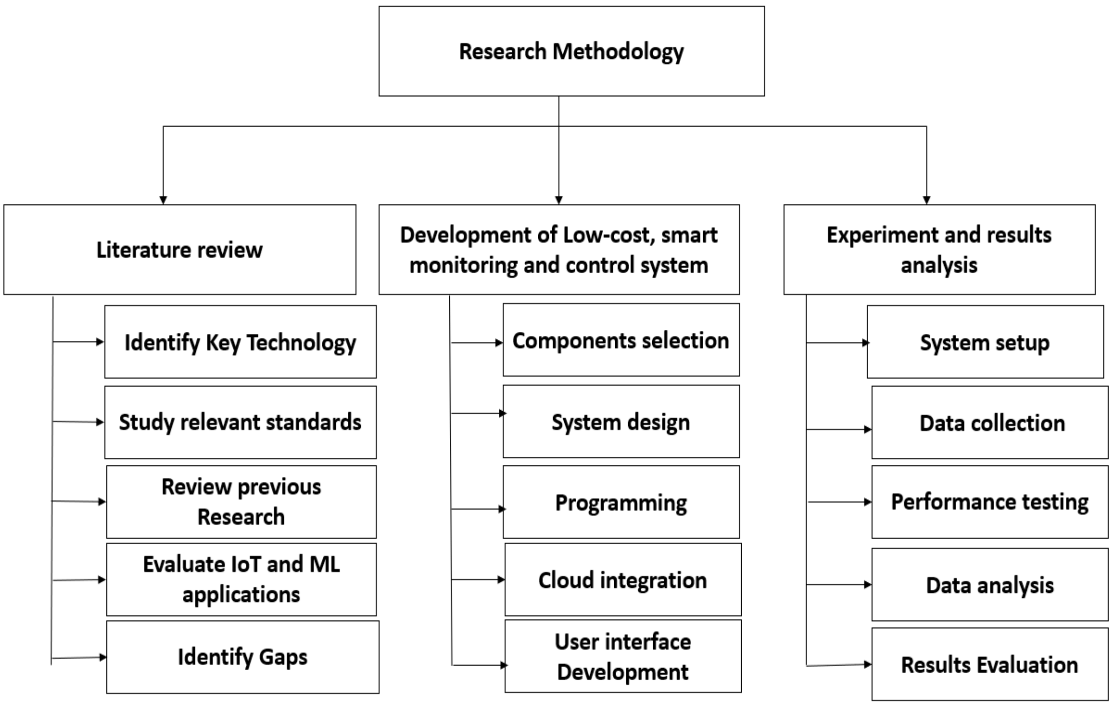


Figure 1.2: Research Methodology structure

1.9 Thesis Structure

The following bullets list and describe the six chapters and appendices that make up the thesis document.

- **Chapter 1:** The introduction of the study, a description of the problem, and a hypothesis.
- **Chapter 2:** This chapter summarizes the existing studies on water quality monitoring and the use of IoT in this field. It highlights the gaps in current research that this project seeks to fill, particularly focusing on affordable and effective monitoring solutions.
- **Chapter 3:** This chapter details the design of the system, describing the components, such as Arduino, and Sensors, and the steps taken to integrate them into functional setup. It explains how the system is structured to monitor and control water quality efficiently.
- **Chapter 4:** The process of developing the software using Arduino IDE is explained. The chapter also covers the setup of ThingSpeak for cloud data monitoring and the creation of a user-friendly interface for viewing data and receiving alerts.
- **Chapter 5:** This chapter describes the experimental setup, and the methods used for collecting data from the water sensor. It discusses the various conditions under which the system was tested and evaluates its performance based on the collected data.
- **Chapter 6:** The final chapter displays the outcome of the experiments, examines the information, and discusses the findings in relation to the project’s objectives. It also identifies any limitations encountered and suggests recommendations for future improvements and further research. Finally, it summarizes the overall contributions of the project.
- **Appendices:** This section includes additional figures, diagrams, detailed parameters, and extra results that support the research but were not included in the main chapters.

1.10 Summary

In conclusion, everyone's health and wellbeing depend on having access to pure and safe water. Regrettably, the current methods of monitoring water quality are costly and challenging to apply, particularly in remote locations. The creation of an affordable intelligent monitoring and control system for the evaluations of water quality is the aim of this project. The intention is to use Arduino, Ethernet shield, and water sensors to increase the accessibility and accuracy of water quality monitoring. In the upcoming Chapter 2, we will examine previous studies and comprehend the various approaches taken.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The literature review will explore both past and current research on online water quality monitoring and control systems utilizing IoT hardware and sensors. This study aims to explore unresolved issues surrounding the implementation of a sensor interface with Arduino for efficient data transmission in such a system. The development of these technologies is essential in mitigating potential harm to our world's ecosystem. The review encompasses a wide range of sources, including books, articles, textbook chapters, updates from water management authorities, and government databases. These sources discuss complex interpretive models and highly accurate predictive models used to anticipate future water quality.

Figure 2.1 below illustrates the structure of the chapter on the literature review. It begins with Water Issues, delving into the difficulties associated with low-income communities' access to water, the continuous water crises, changing water needs, and the essential characteristics that determine water quality. Next, the study will examine several approaches to water quality testing, specifically the Traditional and Current Water Quality sections. These sections will examine both conventional methods and current intelligent systems for problem-solving and monitoring. This section will also go over the fundamentals of online water quality, a review of recent advancements in application controller technology, and the real-world uses of these technologies. Finally, this section will delve into how Internet of Things (IoT) technology revolutionizing water quality management through IoT-based solutions is. It will also explore the characteristics of IoT systems, their integration with water treatment, and their role in supporting real-time monitoring of water quality. This section will provide a comprehensive picture of the impact that IoT is having on water resource management.

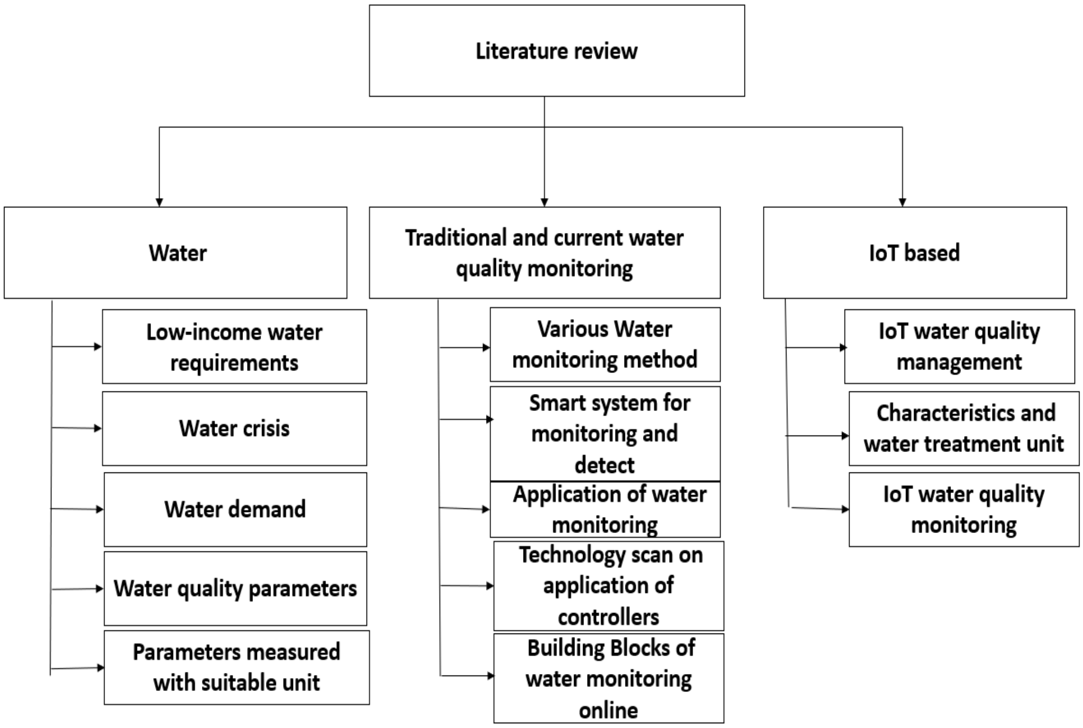


Figure 2.1: Literature review framework

2.2 Low-Income water requirement

The Cape Water Crisis that affected Cape Town between 2015 and 2018 had a significant impact on access to water, with stark disparities evidenced in low-income areas such as Khayelitsha. The informal settlements in these areas are often characterized by inadequate infrastructure and management problems leading to decreased provision of water, thus increasing the suffering of their inhabitants. Although most paying customers would profit from the implementation of a cost recovery model for water services, many impoverished people would not regularly have access to clean and safe drinking water in South Africa, where water is a fundamental human right. There are also inequities in supply that don't only exist spatially but also indicate how socio-economic components can perpetuate drought-prone areas. This situation in Khayelitsha shows a strong need for fairer policies regarding water management aimed at serving all members of society without regard to their economic background (Mokoena, 2023) (Egieya et al., 2024). The apartheid era (1948–1994) saw the peak of the physical and financial division of people that began when Whites first landed in the Cape. Hundreds of thousands of Capetonians who identified as "coloured" or "black" were forced to relocate to poor housing in low-lying neighbourhoods that frequently leaked and limited access to essential utilities like water and sanitary facilities. Fair access to water for all locals has been a goal of post-1994 policy, but municipalities have had difficulty putting those rules into practice, particularly in the fast-expanding informal settlements. The City of Cape Town intensified its water demand control program during the recent drought, including pressure reduction, leak repairs, and public awareness-raising activities.

The fear of "Day Zero" during the first half of 2018 caused a near-panic situation.

But poor communication and a lack of trust also played a part. Cape Town is now looking at additional water sources and creating a new water strategy after being saved by the winter rains. When considered collectively, the city's experiences show that effective water governance must consider the interconnected dangers of drought and flooding, as well as the variety of effects these threats and the city's actions have on a population that is still characterized by high inequality (Mokoena, 2023) and (Enqvist & Ziervogel, 2019).

2.3 Water crisis

Water conservation is a global challenge that has become especially prominent in South Africa, where water shortages are a big issue. Research on how well the workers at one of the South African universities are aware of the water challenge showed that there exist moderate levels of harvesting awareness, even though many people do not really know about campaigns aimed at creating awareness on the matter. However, it appears as if they are willing to acquire more knowledge and thus indulge in responsible use of this precious resource. Thus, according to this research, the actual campaigns being carried out so far leave much to be desired, but improvements can be made. In the future, these programs need to be made more far-reaching and efficient so that they would reach out to many people and encourage them to use water responsibly (Van Der Vyver, 2015).Water-related problems and difficulties at the global, South Asian, and Indian levels were highlighted. The global water crisis and its effects on women in underdeveloped nations are severe problems. Health is negatively impacted by the effects of water on health outcomes and by waste contaminating drinking water sources because of a lack of supply. Millions of lives and livelihoods are at risk because of India's biggest water crisis in history. Currently, 600 million Indians experience moderate to severe water stress, and a lot of people each year die because of insufficient access to clean water. Top Indian cities experienced groundwater shortages, affecting 100 million people. As a result of the situation, village wells and pumps have run dry, forcing thousands of people to flee their homes in search of water (Suresh Lal, 2018).

2.4 Demand of water quality

Between 490mm annual rainfall and severe water scarcity, South Africa was rated among the world’s driest nations. It is a country that has an average requirement of 280 liters per person per day, which compared to the world’s average of 175 liters is too high, and there are many unneeded uses like garden watering. In 1904, South Africa experienced its worst drought, causing it to face a water shortage and consequently bad impacts on agriculture. By 2025, it was anticipated that demand for water would be greater than supply, with a gap of 17% being experienced between supply and demand by 2030. Some national limitations were placed on water use, with most of the provinces except Gauteng having been declared disaster areas due to draught (Grewar, 2019). A 40% decrease was predicted in water supply in 2030 by the UN worldwide; this might impact about 1.8 billion people in total across the globe. Due to these circumstances, improved treatment of wastewater with an emphasis on reusing and secure methods for handling water are essential strategies for realizing SDGs, as stated by the United Nations. That should not be ignored at all (Grewar, 2019). In early 2020, with the challenges of COVID-19 such as unemployment and economic uncertainty, outdoor recreation saw increased participation as one of the few permitted activities. According to (Parsons et al., 2022) , a survey was conducted during this time in Delaware aimed to understand how these experiences affected public willingness to pay (WTP) for improving water quality statewide. The average household’s WTP for water quality improvement only decreased by 7% by May 2020.

2.5 Water quality parameters

This section reviewed the important parameters to be measured for adequate water quality monitoring. the parameters reviewed are as following: TDS, EC, pH, Turbidity, Flow and Temperature.

2.5.1 Total dissolved solids

TDS, or total dissolved solids, is a crucial indicator of the quality of water. The term TDS refers to the number of dissolved materials, such as salts, minerals, and organic matter, in water. In addition to man-made sources like agriculture and industry, these dissolved solids can also originate from natural sources like rocks and soil. Because excessive levels can impact water's taste, safety for drinking, and suitability for various uses, measuring TDS is essential. Comparatively speaking, measuring TDS levels is less expensive and more difficult than measuring electrical conductivity (EC). This is since measuring TDS takes longer and requires more specialist equipment. To simplify matters, researchers have investigated the reliable estimation of TDS from less complex measurements such as EC. By comprehending the connection between TDS EC, we can estimate TDS levels without needing to measure them directly, according to (Parsons et al., 2022) and (Adjovu et al., 2023a) .

2.5.2 Electrical Conductivity

Electrical Conductivity is widely used in a diversity of industries, such as food and environmental monitoring. It is measured by the ability of the solution to conduct electrical current. Electrical conductivity has a direct impact on the number of ions in a solution. When it is used in an environment, it monitors a range of different qualities, such as farms, industrial operations, lakes, and rivers (Thirstrup & Deleebeeck, 2021).

When the electrically conductive sensor is placed in a solution, it detects resistances between two electrodes. Whenever there's a greater conductive from high ion concentration in a solution, then the sensor will detect low resistance, but when the solution has a low conductivity, then it detects high resistance. This electrical conductivity sensor is used in a water quality monitoring system to determine the total dissolved solids and salinity of water bodies. When the conductivity is high, it means that there's more salt, pollutant in the solution, or dissolved minerals, which can have an impact health in ecosystems and aquatic life. Regularly identifying source of contaminants and detecting water quality changes and taking mitigation measures to protect humans and aquatic ecosystem's health but continuing monitoring the conductivity (Banna et al., 2014).

2.5.3 Potential Hydrogen

pH is a measurement of the water's basic alkalinity or acidity. Its definition is the concentration of hydrogen ions divided by the negative log. The pH ranges between 0 and 14 and is a logarithm. Acidic solutions have high concentrations of hydrogen ions, which results in low pH levels, while alkaline solutions have low concentrations of hydrogen ions and high pH values (Tantray et al., 2023). The researchers elucidated that reference electrode drift, which is caused by ion interchange at the solution/reference electrode interface, was a main problem with pH measurement. This drift necessitated frequent recalibrations of ion-selective electrodes, especially pH sensors, leading to up to 70% of the total lifetime cost of a pH probe. Hence, to address this issue, they came up with a new methodology for in situ recalibration of the sensor. It involved tracking the reference potential and thereby permitting automatic recalibration without end-user intervention. The aim of their approach was to lower maintenance costs and improve the precision as well as reliability of pH measurements over time (Sisodia et al., 2022). Researchers constructed a powerful electrolyte-gated organic field-effect transistor (EGOFET) that is sensitive to pH levels from 1 up to 10. Such was made possible through the employment of an imidazole solution on β-cyclodextrin (β-CD) and investigating its pH-dependent supramolecular host-guest complexation. The sensing process involved adding β-CD-modified magnetic nanoparticles in imidazole solutions with different pH values, then trapping these complexes on a magnetic carbon electrode, which served as the gate contact in the EGOFET. The study determined that at higher pH levels, the formation of complexes was preferred, and this caused a negative shift in the transfer characteristics of the device. More importantly, it should be noted here that the carbon gate electrode could be polished and reused while also exhibiting high stability and reproducibility performance. This invention has great potential for pH sensors that are highly reliable with reusable parts (Tamayo et al., 2023).The Figure 2.2 illustrates the different pH levels, ranging from acidic to neutral to basic/alkaline, environmental effects, and examples of the pH. As shown in the diagram, the pH scale shows the level of acidity in a substance ranging from 0 (very acidity) to 14 (very basic). Pure water has a neutral pH equal to 7. Clean rainfall is mildly acidic because its pH ranges from 5.0 to 5.5. However, when it rains and combines with emissions from vehicles and power generation facilities, it tends to be more acidic, ranging from 4.0 on the pH scale. Acidity rises ten times if you go from pH 5.0 to 4.0 (Science, 2021)

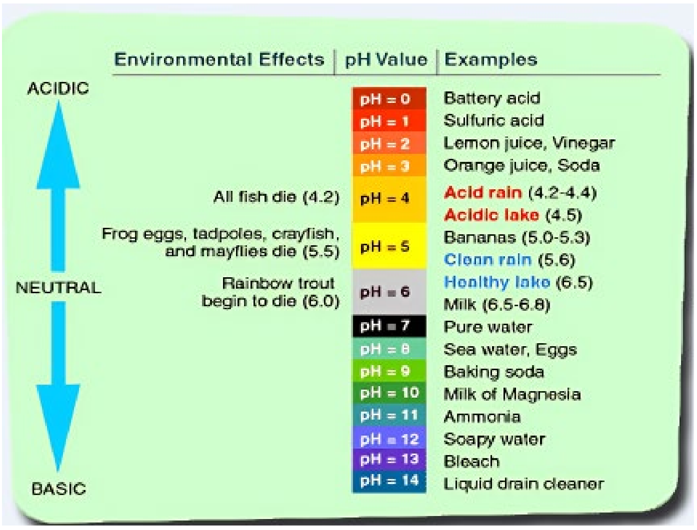


Figure 2.2: **Environmental effects, pH ranges and examples of pH (Science, 2021).**

Water with a pH that is too high or too low is fatal for animals. The toxicity and solubility of heavy metals and other chemicals in water can also be impacted by pH. While some aquatic species can thrive in water with a pH outside of this range, most prefer a pH range of 6.5 to 9.0. pH= -loh10 H+ is the formula to show on how to achieve computing the negative algorithm of the H+ concentration (Cong et al., 2023).

2.5.4 Turbidity

Turbidity sensors are important and mostly used in industries, and the application is to measure the clarity or cloudiness of a liquid. Its indicator is frequently used to determine the quantity of particles of contaminants suspended in water (Postolache' et al., 2002). When the light passes through a liquid, suspended particles in the liquid scatter the light in various directions. Which in turn corresponds with turbidity. The suspended particles can carry contaminants and diseases that are dangerous to people and animals. The detector that is placed at the angle to the light source measures the intensity of the scattered suspended particles (Wang et al., 2018).

The concentration of suspended solids is influenced by both inorganic and organic particles of various sizes. In both the organic and inorganic wastewater effluent compartments, TSS stands tall as the most frequently employed measurement criteria of solids. The TSS concentration in raw sewage typically ranges from 155 to 330 mg/L, with a median value of approximately 250 mg/L as observed in figure 2.3, which demonstrates various degrees of purity in water (Heger, 2017).

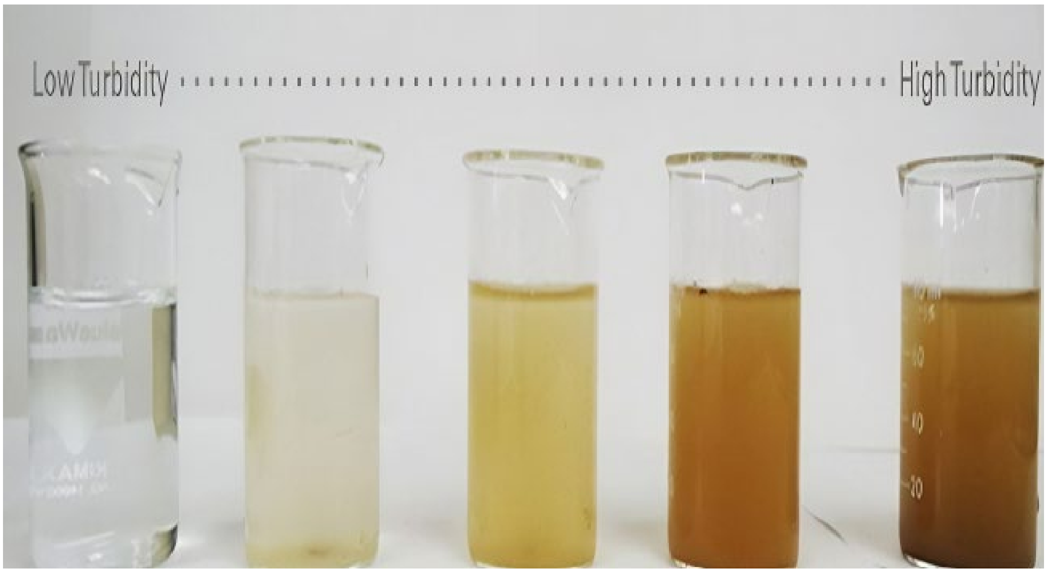


Figure 2.3: Organic and inorganic particles in all sizes contribute to the suspended solids concentration (Heger, 2017)

2.5.5 Water flow

A water flow sensor is designed to measure the rate of water flow within a system. It typically includes a rotor with blades that rotate as water passes through. A sensor, either magnetic or optical, detects these rotations and translates them into flow rate measurements, usually displayed in liters per minute (L/min) or gallons per minute (GPM). This technology is essential for monitoring and controlling water usage in various applications such as irrigation systems, water dispensers, and industrial processes, ensuring efficient and accurate management of water resources (P Singh & S Saikia, 2016).

2.5.6 Temperature

Temperature is a key environmental factor that impacts the physical and chemical properties of the water, such as its electrical conductivity, total dissolved solids (TDS), etc. Many studies have shown that temperature consistently affects the conductivity and TDS of water from various source (Rusydi, 2018). Explain that water temperature has a big impact on how well ions move in water. In water, ions move faster, which boosts conductivity. (P Singh & S Saikia, 2016) adds that the link between EC and TDS isn't always direct and can be heavily affected by sanity. A study found that the water temperature was between 27 and 28.1 degrees Celsius. pH ranged from 6.10-6.97, conductivity 8,25-14.46 µS/cm, and TDS ranged from 4.13-7.22 mg/L, so clearly the temperature has an impact on water parameters.

2.6 Comparison between traditional and current water quality monitoring Techniques

The Table 2.1 below provides the summary of review findings on traditional and current water quality monitoring methods.

Table 2.1: Review summary of Traditional and current water quality monitoring Methods

	Paper	Aim	Methods	Limitations	Achievement
1.	(Barros et al., 2019) Study of the requirements of an autonomous system for surface water quality monitoring.	The paper aimed to define the technical requirements and technologies needed for continuous water quality monitoring.	The researchers proposed a new approach to data collection, addressing the shortcomings of traditional methods.	Traditional approaches were irregular, inconsistent, and manual, indicating a pressing requirement for better systems of monitoring.	The research was a success for constructing a monitoring system that works in real time and makes better use of water resources through renewable methods with the use of wireless energy transfer.
2.	(Bonadonna et al., 2019).	The researchers set out to explore and describe new	They focused on advances in molecular biology,	The researchers pointed out that traditional culture	They highlighted how NGS and other rapid detection methods

	Innovative analytical methods for monitoring microbiological and virological water quality.	techniques for quickly detecting microorganisms in water.	like PCR, biosensors, immuno-magnetic separation, flow cytometry, and next-generation sequencing (NGS), as effective ways to identify and measure microorganisms.	methods are slow and often miss the full extent of contamination, causing delays in detection.	can significantly improve the speed and accuracy of identifying microbes in water, offering better insights into microbial diversity.
3.	(Pichel et al., 2019) The problem of drinking water access: A review of disinfection technologies with an emphasis on solar treatment methods	The researchers aimed to review both traditional and new water purification technologies, with a special focus on solar disinfection.	They looked at existing water treatment methods used on a medium to large scale and explored new technologies that are currently being developed.	They noted that traditional water treatment methods, though effective, often require a lot of energy and chemicals, which limits their use in the most vulnerable areas.	The study highlighted solar disinfection as a highly suitable method for producing safe drinking water in developing countries and introduced a new technology in this field.
4.	(Adjovu et al., 2023b)	The purpose of the study was to present	The researchers used conventional	The study highlighted that using traditional	By improving spatiotemporal

	Measurement of Total Dissolved Solids and Total Suspended Solids in Water Systems: A Review of the Issues, Conventional, and Remote Sensing Techniques	a thorough analysis of the methodologies employed in water quality measurement (WQPs), with an emphasis on total suspended solids (TSS) and total dissolved solids (TDS).	and gravimetric approaches by remoting sensor (RS) applications using satellite and UAV sensors.	techniques for measuring TDS and TSS is costly and labour-intensive. And using a remote sensing technique has atmospheric interference and resolution issues.	regulation for detecting changes in water quality parameters, they further stated that by using atmospheric mechanisms and utilizing ML and deep learning models to achieve an effective monitoring system. .
5.	(Sani et al., 2023) Drawbacks of Traditional Environmental Monitoring Systems	This study intends to identify the challenges and shortcomings that traditional methods of measuring water quality face and point out the necessity for other strategies that	The methods used for this study are traditional environment monitoring techniques and their drawbacks of evaluating water quality.	The limitations are because of expensive equipment and skilled personnel required for monitoring, which can lead to human error, data loss, and	Drawbacks of traditional environmental monitoring systems are highlighted in this study, and challenges in evaluating water quality using

		can conquer these problems in comparison with them.		an inability to predict patterns.	outdated monitoring techniques are discussed.
6.	(Zainurin et al., 2022) Advancements in Monitoring Water Quality Based on Various Sensing Methods: A Systematic Review	The study's primary goal was to evaluate various traditional and modern approaches to water quality monitoring and to pinpoint the advantages and disadvantages of each approach.	The study examined the parameters, complexity, and dependability of the conventional and current approaches under investigation. Cyber-physical systems (CPS), virtual sensing, the Internet of Things (IoT), and optical approaches were among the techniques used.	The investigation made clear that both the sensors used to detect contaminants and the earlier monitoring techniques had limits when it came to obtaining accurate real-time measurements and yielding useful data on the quality of the water.	According to the study's findings, CPS is a good tool for monitoring the quality of water, but more sophisticated, real-time detecting techniques are more costly.
7.	(Satyanarayana Murthy & Ahamed, 2023)	This paper's main goal was to solve the	Methods used were wireless sensor	The limitations of this study were traditional	The achievement highlighted a

	Water quality monitoring and measuring physicochemical parameters using wireless sensor networks	important challenges by presenting a novel method for monitoring water quality utilizing wireless sensor networks.	networks and physicochemical parameter measurements.	methods, which involved delayed outcomes and a lack of real-time data due to cumbersome grouping of points.	wireless sensor network for real-time monitoring, and physicochemical parameters measured include pH, temperature, conductivity, and flow rate.
8.	(Jin, 2022) Application of remote sensing technology in water quality monitoring	The aim of the study was to introduce remote sensing technology and discuss data sources and indicators for the methods in remote monitoring.	The methods for this paper were empirical, semi-empirical, analytical, automation, and machine learning methods.	The limitations included labor-intensive, expensive, and time-consuming traditional water quality monitoring methods. The development problems in water quality remote sensing monitoring neighborhoods.	The achievements for this study: data source for water quality remote sensing included satellites and UAVs. And the sensing indicators that were used included FUI, suspended solids, CDOM, Chla, and TN.

9.	(Manjakkal Libu et al., 2023) Toward Real-Time Water Quality Monitoring Using Wireless Sensor Networks	The use of wireless networks for water quality monitoring and the switch from manual to real-time monitoring were the main topics of this study.	This study primarily used real-time monitoring using sensor networks (WSNs) using industry 4.0 technologies like IoT and big data analytics.	Limitations of the study manual approaches lack real-time monitoring and skilled work. The traditional methods require special apparatuses and laboratory analysis.	Using wireless sensor networks and industry 4.0 technology integrations, the researchers were able to monitor water quality in real-time.

Table 2.2 outlines the key research gaps in traditional water quality monitoring techniques and demonstrates how this thesis addresses these gaps using modern, low-cost, smart monitoring systems.

Table 2.2: Review summary of Traditional and current water quality monitoring Methods

Research Gaps	Definitions	How the gaps are addressed in this project
Real-Time Data Collection	None of the conventional ways can allow for real-time monitoring; hence, problems are detected with a lot of delay.	This study will provide an approach that employs up-to-date sensors for the ongoing online monitoring of factors influencing water quality, as shown in Chapter 5, to all different subsections of the results that were captured from the Arduino IDE and the visualization graph from ThingSpeak.
Cost-Effectiveness	Old systems used in monitoring might be too expensive because they depend on laboratory facilities and specialized devices required for analyses.	This work describes an inexpensive monitoring solution employing cheap sensors as well as components while remaining effective at monitoring water quality, as shown in Chapter 3, which is a methodology, and the descriptions of the low-cost components that are used for this project are moved from

		Figure 3.2.1 to Figure 3.2.18.
Proactive Management and Early Detection	Traditional ways may not give timely alerts, leading to delayed responses to municipalities for monitoring purposes, hence hindering any chance of acting when there are changes in the water parameters under disharmony in concentrations due to environmental factors like very high temperatures or low pH values, etc.	The inclusion of the IFTTT alert system makes real-time alerting possible in this thesis so that users can know instantly when there are any alterations that occur within water quality, as illustrated in Chapter 4 Figures 4.15 and 4.16.
Remote Accessibility and Data Visualization	Old methodologies do not allow remote access or even visualizations for better observation, hence limiting one's chance to observe water quality from faraway places within which individuals are found living far away from sources.	The system can integrate with cloud-based applications such as ThingSpeak so that it enables remote access as well as real-time data display, as shown in Chapter 4 Figure 4.8.
Automation and Efficiency	Manual sampling and testing procedures make use of old techniques that require many workers; hence, they consume much time too.	As the final system is shown in Figures 4.17 and 4.8 of Chapter 5, the suggested system reduces manual work and increases efficiency by automating data collection and processing.

2.7 Various Water Monitoring Methods

When it comes to the assessment and preservation of the state of water in natural and unnatural environments, Table 2.2 elaborates on the diverse water monitoring techniques. Monitoring is important in tracing pollutants, understanding the chemistry of water, and following biological processes. These greatly contribute to ensuring the health of both water and the environment. Each method focuses on parameters of water quality, providing a comprehensive average view of aquatic habitats and the overall safety of water for diverse uses (B A Karomah, 2022).

Table 2.3: comparison of various water monitoring methods (B A Karomah, 2022) **.**

1. CDOM/FDOM MONITORING	Naturally occurring dissolved organic matter (CDOM), also known as chromophore matter, is found in streams. This organic substance breaks down and releases tannin, an organic contaminant that discolours water, when it absorbs UV light. In addition, tannin reduces oxygen content and water's pH (acidity).
2. CHLOROPHYLL-FLUORESCENCE ANALYSIS	Using algal toximeters, the ratio of wet chemicals to active chlorophyll in the water sample is determined. This method helps monitor water quality and identify excessive algal growth.
3. CONDUCTIVITY, SALINIT, TDS MONITORING	The conductivity of a lake or river serves as a trustworthy gauge of its quality. The salinity and total dissolved solids (TDS) content of water are influenced by conductivity, and these factors in turn affect the oxygen content of the water.
4. pH and KH MONITORING	Water testing sets are a useful tool for

	measuring pH because they come in a variety of color-coded pH values. The most accurate way to determine the pH range of the water is with these kits. However, measured data from automated pH sensors is provided with a precision of up to two decimal places. Another factor to consider is KH, or carbonate hardness, which regulates the pH of water by measuring the concentrations of carbonate and bicarbonate.

2.7.1 Method to measure the suitability of water.

Table 2.3 lists the parameters used to assess the suitability of water quality for different purposes. These include acceptable ranges of pH, turbidity, temperature, dissolved oxygen, and electrical conductivity major indicators of water quality. These parameters determine the suitability of water for drinking, industrial use, and the sustenance of aquatic creatures. Therefore, monitoring these parameters within their limits ensures that the water is safe for use or living, considering health and environmental concerns.

Table 2.4: Water quality parameters and their acceptable ranges

Parameters	Unit	Range
pH	pH	6.5-8.5
Turbidity	NTU	5-10
Temperature	C	10-25
Dissolved Oxygen	Mg/l	4-6
Electrical conductivity	mS/cm	0.05-0.8

2.7.2 Water quality index value and mathematical of water quality parameters

- Sensor calculations techniques

pH is a function of the activity of the hydrogen ion in a solution. The unit that is used to measure the acidity of a substance is called pH. The term "H" is a negative log of the hydrogen ion concentration (Y Chen & R Compton, 2022).

$$pH = -\log aH^{+} \left(\frac{yH^{+}}{m^{\theta}}\right) \tag{2.1}$$

Where aH^{+} = a single ion activity (measured on the molality scale kg^{-1}

aH^{+} = activity coefficient of the hydrogen ion H^{+} at a molality of $m_{H^{+}}$

m^{θ} = standard molality

- Pollutants Detection algorithm

An algorithm was developed in an event of early pollution detection. This fuse multiple sensors in for early water contamination detection. The algorithm is based on normalized sensor output given by equation 2.

$$\boldsymbol{N_i} = \frac{|\boldsymbol{S_i} - \boldsymbol{\mu_i}}{\boldsymbol{\tau_i \sigma_i}} \tag{2.2}$$

- Pressure sensor technique.

Piezoresistive sensors were used. The basic principle behind the sensor lies when a material has been mechanically deformed which alters the change of electrical resistance of the material based. Equation 3 below stimulates the phenomenon (Almassri et al., 2015).

$$R = \frac{\rho x l}{t x w} \tag{2.3}$$

Where:

$\rho = \mathrm{Resistivity}$

$R = Resistance$

$l = length$

$t = thickness$

$w = width\ of\ the\ contact$

- Water Quality index Calculation

Several calculations for water quality index were performed for certain parameters namely: pH, turbidity, TDS, BOD, alkalinity, chlorides, and electrical conductivity. By using equation 4, these calculations were made/obtained following (G, 2020) .

- Pollutant Detection algorithm

An algorithm was developed in the event of early pollution detection. This fuse multiple sensors for early water contamination detection. The algorithm is based on normalized sensor output given by equation 2. Water quality index calculation

Several calculations for water quality index were performed for certain parameters namely: pH, turbidity, TDS, BOD, alkalinity, chlorides, and electrical conductivity. By using equation 4, these calculations were made/obtained following (G, 2020) .

$$Q_i = \frac{Mi-Li}{Si-Li} x\ 100 \qquad (2.4)$$

$$W_i = \frac{K}{S_i} \qquad (2.5)$$

$$WQI =>^n \frac{W_i Q_i}{\sum_{i17}^{n} Wi} \qquad (2.6)$$

Where:

$Q_i = \text{sub index of the ith parameter}$

$W_i = unit\ weightage\ of\ ith\ parameter$

$M_i = monitored\ value\ of\ the\ parameter$

$L_i = ideal\ value$

$n = Number\ of\ parameters$

$S_i = Standard\ value\ of\ the\ ith\ parameter$

- Turbidity removal efficiency

The percentage turbidity reduction is achieved by using the following equation (J.K Mwabi et al., 2013):

$$\%Tubrbidity\ removal\ efficiency = \frac{(Turbidity_{unfiltered} - Turbidity_{\ filtered})}{Turbidity_{unfiltered} \times 100\%} \qquad (2.8)$$

- Bacteria removal efficiency

To calculate the log of a bacteria reduction by the in equation below and convert it to the percentage to kill.

$$\log red = log_{10} Bacteria\ count_{\ before\ fil} - log_{10} Bacteria\ count_{after\ fil} \qquad (2.9)$$

$$The\ kill\% = 100_{count} - \frac{survivor}{initial\ count} \times 100 \qquad (2.10)$$

Figure 2.4 shows a process for calculating the WQ index value. Collecting various water quality parameters such as pH, turbidity, temperature, dissolved oxygen, and electrical conductivity starts from Step 1. Afterward, these parameters become sub-indices that indicate their respective contribution to the overall water quality in Step 2. Each parameter is accorded its weight according to its relevance in measuring water quality in Step 3. An aggregation function is then employed in Step 4, which considers all weighty sub-indices to generate an overall WQ value, or more simply put, one monolithic measurement of water quality (Uddin et al., 2021).

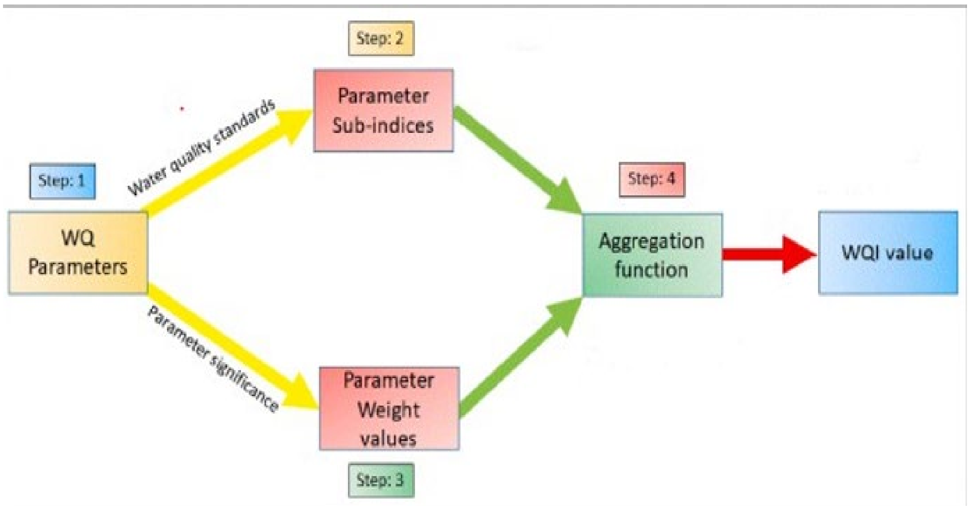


Figure 2.4: General structure of the WQI model (Uddin et al., 2021).

2.8 Parameters Measured with Smart Quality Online Monitoring systems.

After conducting extensive examinations, the U.S. Environmental Protection Agency (USEPA) has determined that chemical and biological contaminants have a considerable impact on electrical conductivity (EC), pH, turbidity, and oxidation-reduction potential (ORP) (Lambrou et al., 2014a). This research highlights the necessity of real-time monitoring for the early detection of water characteristic changes. Likewise, (Geetha & Gouthami, 2016) Conducted a comprehensive review to identify important parameters used in smart water quality monitoring with their summary as well as corresponding reference points as illustrated in Figure 2.5.

Parameters monitored	References
pH	Vijayakumar and Ramya (2015), Mitar et al. (2016), Tomoaki et al. (2016), Vinod and Sushama (2016), Niel et al. (2016), Goib et al. (2015), Theofanis et al. (2014), Peng et al. (2009), Jayti and Jignesh (2016), Poonam et al., 2016, Xin Wang (2011), Gerson et al. (2012), Pandian and Mala (2015), Liang (2014), Xiuna et al. (2010), Christie et al. (2014), Azedine et al. (2000), Offiong et al. (2014), Anthony et al. (2014), Sathish et al. (2016)
Dissolved Oxygen	Vijayakumar and Ramya (2015), Goib et al. (2015), Jayti and Jignesh (2016), Gerson et al. (2012), Liang (2014), Xiuna et al. (2010), Christie et al. (2014), Offiong et al. (2014), Anthony et al. (2014)
Oxidation reduction potential	Niel et al. (2016), Theofanis et al. (2014)
Temperature	Vijayakumar and Ramya (2015), Mitar et al. (2016), Niel et al. (2016), Theofanis et al. (2014), Peng et al. (2009), Jayti and Jignesh (2016), Poonam et al., 2016, Gerson et al. (2012), Pandian and Mala (2015), Liang (2014), Xiuna et al. (2010), Francesco et al. (2015), Christie et al. (2014), Azedine et al. (2000), Anthony et al. (2014)
Turbidity	Vijayakumar and Ramya (2015), Tomoaki et al. (2016), Vinod and Sushama (2016), Theofanis et al. (2014), Jayti and Jignesh (2016), Poonam et al., 2016, Gerson et al. (2012), Pandian and Mala (2015), Francesco et al. (2015), Offiong et al. (2014), Sathish et al. (2016)
Conductivity	Vijayakumar and Ramya (2015), Niel et al. (2016), Theofanis et al. (2014), Jayti and Jignesh (2016), Gerson et al. (2012), Francesco et al. (2015), Christie et al. (2014), Azedine et al. (2000), Anthony et al. (2014), Sathish et al. (2016)
Water level sensing	Thinagaran et al. (2015)
Flow sensing	Niel et al. (2016)
Air temperature	Mitar et al. (2016)
Relative Humidity	Mitar et al. (2016)
Presence of organic compounds	Mitar et al. (2016)
Chlorine concentration	Eliades et al. (2014), Francesco et al. (2015)
Chlorophyll	Francesco et al. (2015)

Figure 2.5: The key parameter in smart water quality monitoring (Geetha & Gouthami, 2016).

2.9 IoT water quality Management

According to (Jang M. Y. B. et al., 2020) Studying in urban area settings, a water supply that incorporates distribution, treatment, and source management is essential. Traditional water quality studies tended to concentrate on factors rather than covering all in depth, which reduced accuracy and prediction. The current study focused on the development of a smart data analysis by following the method of indicators of water quality to analyze and predict water quality. The raw data was collected from the water supply. With the focus of creating two methods, which are random forest and adaptive learning rate BP neutral network, the project was proven successful for future risk control and decision-making in large cities water supply (Wu et al., 2020). (Kirankumar et al., 2021) Conducted research on smart monitoring and water quality management by implementing emerging technology and proposed a method using Arduino and machine learning. The Arduino gathered the information data from water parameters, and the parameters were able to predict data in advanced using machine learning techniques. When the parameters exceed the standard range, it will send an alert to the farmer in advance to keep aquafarming water safe. According to (Ba & McKenna, 2015), a water treatment facility had to close its doors in 2016 due to pollution in the plant's water flow. Since sensors can determine the purity of water and provide a safety solution for users, it was decided to utilize them to identify the quality of the water after the supply water failed to identify the origin of the contamination that was polluting. The sensor data was subsequently uploaded to the GPS data connection.

2.10 IoT based water quality monitoring

According to (K & S, 2017), water is a basic human need, and systems must be put in place to continuously and rigorously verify the quality of drinking water. The suggested approach was an inexpensive Internet of Things-based real-time water quality monitoring and control system. The device was made up of physio-chemical sensors that measured the temperature, turbidity, conductivity, pH, and flow rate of water, among other physical and chemical characteristics. These sensors identify contaminated water. A Raspberry Pi processed the sensor data before sending it to the cloud. Finally, cloud computing was used to display the collected data in the cloud and control the water flow in the pipeline with IoT. (Astya, 2017) stated that the data from the Human Rights Commission indicates that 20 million individuals in our nation continue to consume contaminated water. To decrease illnesses caused by contaminated water and prevent water pollution, we must constantly monitor certain water quality indicators. In early water monitoring techniques, data are manually gathered from a variety of sources. This will take a lot of time and careful physical work. We must use fresh approaches to water monitoring to solve these issues. The primary goal used was to develop an IoT-based system to continuously monitor the parameters affecting water quality. The suggested model makes use of a variety of sensors to measure the necessary parameters. A core controller processes the parameter values that come from the sensors. Using a Wi-Fi module, measured data from sensors is shown on the cloud platform. According to (Faruq et al., 2017), people who reside in isolated locations with limited availability of pure drinking water should take note of the need for installing an inexpensive water quality monitoring system. The very accurate microcontroller-based water quality monitoring device can measure temperature, turbidity, and hydrogen potential (pH), among other water parameters. These water quality parameters must be detected to live a healthy life because various water sources are becoming contaminated owing to overpopulation. There are many different analytical methods for determining the quality of water, some of which are time-consuming and some of which are employed for industrial purposes and are not suitable for basic water quality monitoring systems. Therefore, it is unnecessary to create a straightforward device that continuously tracks different water-related metrics. The researchers used a microcontroller that serves as the central processing unit (CPU), while numerous sensors detected different parameters and communicated the data to the microcontroller, and then the results were displayed on an LCD.

(Lakshmikantha et al., 2021) Highlighted that, given the increasing contamination and pollution of drinking water, the greatest threat facing humanity today is water pollution. The diseases that animals and humans may contract from the contaminated water influence the life cycle of the environment. When contamination of water is discovered early, hazardous situations can be avoided, and the proper course of action can be followed. To guarantee the provision of clean water, real-time water quality monitoring is required. As sensors, connectivity, and Internet of Things (IoT) technology progress, intelligent approaches to monitoring water contamination are becoming increasingly crucial. The developed model was assessed using three water samples, and the parameters were forwarded to the cloud server for additional processing. These researchers only utilized fewer samples, but they could have used more, but the methodology they chose is more reasonable and appears to be more accurate for the use of water quality, depending on the number of sensors they planned to employ. According to (Alabrsh, 2023) water is an essential resource for life and nature. That is why water springs must be constantly monitored to detect contaminants that can threaten water quality.

(Pasika & Gandla, 2020) discussed significant improvements in wireless communication that are opening new sensor possibilities. The latest advances in sensor network technology are critical for environmental applications. Connections between different devices are made possible by the Internet of Things (IoT), allowing for the collection and sharing of data. Water is one of the necessities for human existence; hence, it is imperative to incorporate a system for routinely assessing the water's quality. Nearly 40% of deaths worldwide are caused by contaminated water. There are steps that need to be made to ensure that people have access to potable water in both urban and rural areas. The Water Quality Monitoring (WQM) system uses Internet of Things (IoT) technology to efficiently and economically monitor the quality of drinking water. The proposed system in this study makes use of multiple sensors to measure different parameters such as the surrounding air humidity, temperature, tank level, water turbidity, pH value, and so on. In addition, the personal computer (PC) processed extra data when the microcontroller unit (MCU) was attached to these sensors. The ThinkSpeak IoT-based application transfers the gathered data to the cloud to monitor the water quality. This researcher's methodology seems more affordable and accurate because of the cheap components to use and the free cloud.

2.11 Feed water characteristics and water treatment unit processes

(Verlicchi & Grillini, 2020) conducted a study to illustrate examples of water quality problems in vulnerable communities of the Sub-Saharan region, which is mostly home to the nations with the least access to clean drinking water, as per a recent evaluation conducted by the World Health Organization (WHO). The Republic of South Africa's (RSA) and Mozambique's (MZ) rural and peri-urban areas' amounts of common contaminants, inorganic substances, microbes, and micropollutants are summarized in this paper. Their findings were contrasted with the two nations' drinking water regulations. Large amounts of microorganisms were found to be present in surface water; other important components are boron (MZ) and nickel (RSA). The two most significant elements in connection to groundwater are lead and arsenic (RSA) and boron, sodium, and chloride (MZ).

The process by which water treatment facilities ensure that the public has access to drinkable water is shown in Figure 2.6. Public water systems use a variety of treatment techniques, including filtration, flocculation, sedimentation, coagulation, and disinfection. These actions are crucial because they aid in the removal of impurities, making the water fit for consumption under specific circumstances (water, 2022).

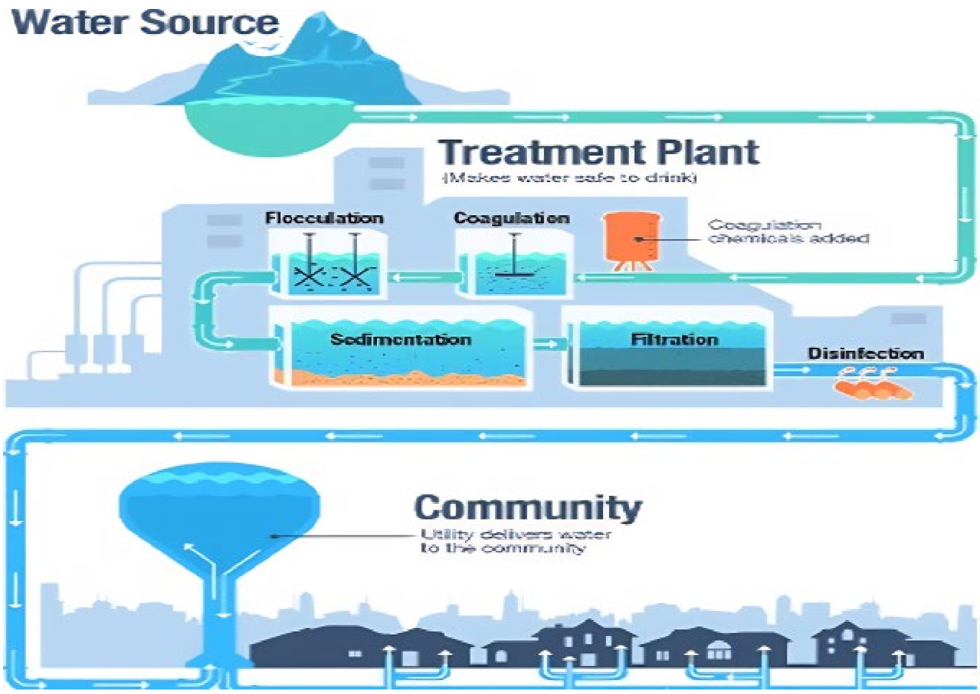


Figure 2.6: **Water treatment unit (water, 2022).**

2.12 Smart system for Monitoring and detecting water quality

The method for detecting contamination events and tracking time series data related to water quality was created by (Rajalashmi et al., 2021). The technique used affine projection techniques in conjunction with an autoregressive (AR) model to predict water quality time series. Then, to determine if contamination events are occurring or not, apply online methods for identifying change points in the projected residuals. In particular, the efficiency of four change-point detection methods was evaluated using residuals from four water quality parameters: online Bayesian changepoint detection (OBCPD), binomial event discriminator (BED), cumulative sum (CUSUM), and sequential probability ratio test (SPRT). The key performance metric for assessing the four change-point detection systems is provided by the receiver operating characteristic (ROC) curve, which is defined as the true positive rate as a function of the false positive rate. The selection of change-point detection was also discussed, as were algorithms to consider when creating effective systems for detecting contamination. According to (Lambrou et al., 2014b), the study employed two distinct methods to detect alterations in water quality. The first method involved laboratory and testing loop evaluations of sensors, which were then linked to event detection algorithms. These analyses made it possible to measure the chemical changes brought on by contamination directly in the background of the water's quality.

An automated sampling system powers an example of online water quality monitoring in Figure 2.7. Sensors in this system continuously measure various aspects of water quality at predetermined intervals. Commonly monitored parameters in rivers and several other applications usually include dissolved oxygen (DO), pH, conductivity, and temperature. Additionally, it is common to analyse ammonium, phosphate, and heavy metal ions, thereby providing an overall evaluation of water quality (Wiranto Goib et al., 2015).

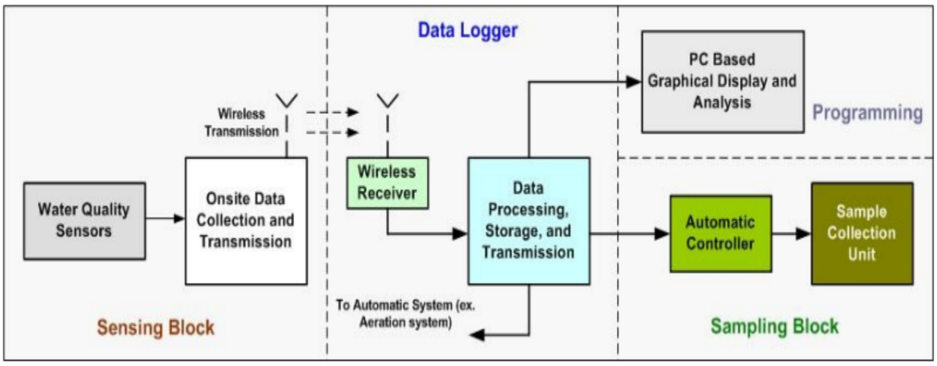


Figure 2.7: Concept of online water quality and control system (Wiranto Goib et al., 2015)**.**

2.13 Application of water monitoring System

Figure 2.9 emphasizes the varied uses of online smart water quality monitoring systems in domestic, industrial, and urban domains. (Geetha & Gouthami, 2016) did an all-inclusive syllabus on the applications of those systems, thus collating certain statements that showed how these technologies ensure quality of water across different situations.

Application	References
Domestic running water	Vijayakumar and Ramya (2015), Niel et al. (2016), Theofanis et al. (2014), Jayti and Jignesh (2016), Poonam et al., 2016, Xin et al. (2011), Xiuli et al. (2011), Offiong et al. (2014)
Domestic Stored water	Thinagaran et al. (2015), Vinod and Sushama (2016), Pandian and Mala (2015), Azedine et al. (2000), Sathish et al. (2016)
Lake, River, Sea water, Environmental monitoring	Tomoaki et al. (2016), Vinod and Sushama (2016), Peng et al. (2009), Francesco et al. (2015), Christie et al. (2014), Haroon and Anthony (2016), Anthony et al. (2014), Li et al. (2013)
Aquaculture centers	Goib et al. (2015), Xiuna et al. (2010), Gerson et al. (2012)
Drinking water distribution systems	Eliades et al. (2014), Ruan and Tang (2011)
Water and Air quality	Mitar et al. (2016)
Not limited to specific application	Liang (2014), Wei et al. (2012)

Figure 2.8: Application of smart water monitoring systems (Geetha & Gouthami, 2016).

The water quality monitoring plays an essential role in safeguarding our water bodies and the wellbeing of the communities that depend on water quality. The system plays a crucial role in monitoring various factors, including the chemical, physical, and biological properties of water. They have a wide range of applications in different fields, including rivers, lakes, industrial, and agricultural operations. The system enables early detection of water quality contamination by continuously monitoring parameters such as pH, turbidity, oxygen dissolved, and the presence of contamination. Early detection is crucial for the control of long-term water supply, the preservation of underwater ecosystems, and the prevention of waterborne disease. Figure 2.10 below illustrates the process of the monitoring and control system, while the diagram of the internet of things-based agriculture system includes sensors, cloud computing, data processing through database stations, communication protocols, and an alarm system that activates the actuators in the event of water contamination (Prapti et al., 2022).

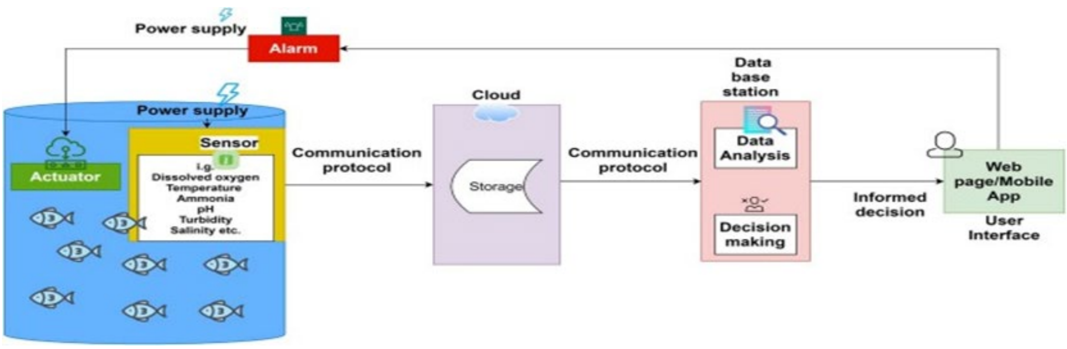


Figure 2.9: Process of monitoring and control system (Prapti et al., 2022).

2.14 Technology scan on the application of microprocessors in water quality monitoring

This research assessed optical sensors for estimating suspended solids (CSS) in the Coxim River, comparing Nephelometric methods with the A400-600 absorbance approach using UV/VIS'S spectrophotometers. The study was carried out in highly sedimented waters and evaluated the correlation of these techniques with sediment characteristics, and strong correlations were established (R = 0.942, R = 0.959, respectively) for the nephelometric and A400-600 methods. Through granulometric analysis, it was established that sediment consisted largely of silt at a percentage of 52.6%, while other proportions were clay, fine sand, and medium sand, respectively. The A400-600 technique recorded lower detection limits during dry seasons (26.47 mg/L) than during wet seasons (Dr. Puneeth GJ et al., 2024).

Ongoing advancements in microprocessor speed, affordability of sensors, wireless communication, cloud storage, and the application of machine learning and artificial intelligence are driving significant innovation in smart water quality monitoring. These technologies are increasingly facilitating the adoption of the Internet of Things (IoT), promising more effective and efficient solutions for monitoring and managing water quality in real-time.

2.15 Building blocks of Smart water quality online monitoring

Figure 2.10 shows an example of smart water quality that is tracked online. There were numerous sensors available to track various aspects of water quality. The water that was being tested has these sensors in it. Either rushing or stored water may be present. Using a wireless communication device, the sensor transforms the physical parameter into an equivalent, measurable electrical quantity that is sent as input to the controller. Reading data from sensors, processing it, and sending it to the application via the proper communication protocol is the controller's main job. The needs of the application determine which communication technology and characteristics need to be monitored. The application has features for data analysis, data administration, and a warning system depending on factors that are tracked. (Geetha & Gouthami, 2016).

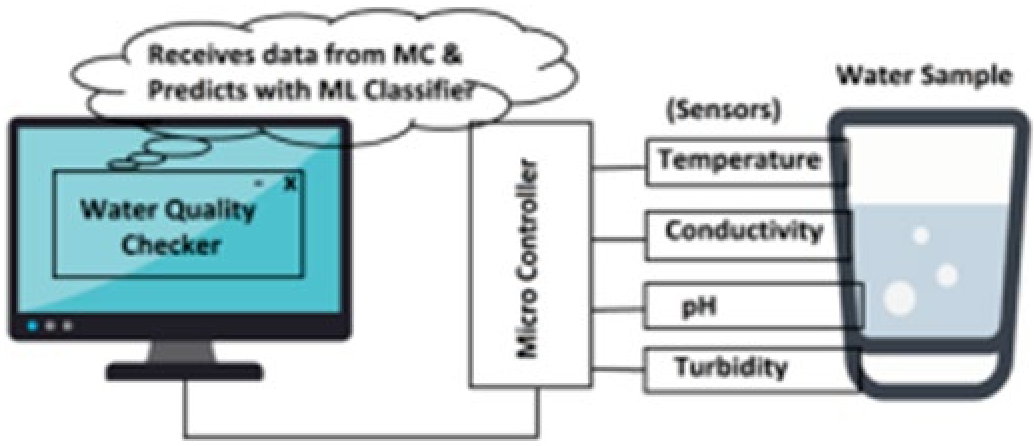


Figure 2.10: The schematic diagram of water monitoring system(Geetha & Gouthami, 2016)**.**

The literature reviewed underscores the need for affordable smart systems designed to monitor water quality across diverse locations. These systems integrate technologies like water quality sensors, microcontrollers, Wi-Fi, and cloud platforms to deal with the difficulties of restricted resources in water quality monitoring and control. Particularly crucial for rural communities reliant on local water sources, these solutions enable real-time data collection, analysis, and automated responses to ensure water quality meets established standards. Further research and practical implementation of these cost-effective, smart solutions are essential to enhancing the health and development of rural areas.

2.16 Internet of Things and ThingSpeak integration

With the help of microcontrollers, digital communication transmitters, and receivers, common things will soon be able to communicate with others and other gadgets to form a kind of communication network known as the Internet of Things (IoT). As a result, the Internet of Things will become a vital part of it. The goal of the Internet of Things (IoT) concept is to gain more notoriety online. Furthermore, by enabling straightforward communication with a variety of gadgets, such as home appliances, security sensors, engines, monitors, etc. Several industries, including home automation, industrial automation, healthcare, mobile healthcare, assistance for the elderly, smart power management, intelligent networks, and cars, can benefit from the IoT concept. These networks are made up of intelligent, independent, stand-alone devices that may observe or engage with their environment. Additionally, the term "smart" refers to an object's ability to gather data from its environment, process it, share it with other objects, and interact independently with it. This interaction relies on data that takes advantage of embedded electronics, software sensors, actuators, and communication capabilities. In other words, by being dispersed across the environment and incorporated into many contexts, these smart objects have the potential to significantly enhance how people interact with their surroundings, resulting in the development of services and applications in every area. The growing popularity of the Internet of Things has led to the creation of numerous platforms that link diverse things through the Internet. It is difficult to provide IoT solutions for different smart products from several manufacturers because there are no standards for things like communication protocols and connections. As a result, a novel development strategy based on the cloud user interface has been proposed (Miry & Aramice, 2020). A source such as ThingSpeak enables users to exchange their data from various sources and manage data in various formats. An open-source data platform built on the cloud underlies it. It allows for online data gathering, MATLAB analysis, and activation via an open API. Monitoring, MATLAB visualization, data storage, and integration of user data with multiple third-party platforms, including well-known IoT platforms like Arduino, Thing HTTP, and MATLAB, are all made possible with the aid of apps and extensions. Several applications gather sensor data in each channel, which has eight fields that can hold different types of data, one status field, and three site fields. These applications include Time Control (which uses the ThingSpeak application to perform actions automatically at pre-set times), Tweet Control (real-time interaction), React (which reacts when channel data is compatible with certain conditions), and Talkback Reaction measures (Ray, 2016).

2.17 Summary

This chapter addressed issues related to water quality, specifically focusing on the challenges faced by the rural population in accessing clean water. It provided information related to the prevailing water crises in the world today, which includes clean water as a resource cutoff, and the parameters of water quality.

Additionally, traditional means of assessing water for its quality were discussed in the transition as well to better methods of assessing water quality. The chapter illustrated the capabilities of IoT in modern-day water management techniques by showcasing its use in water treatment processes.

In the following chapter, attention will turn to the research strategy on the development of low-cost smart water quality monitoring and control systems. It will provide a comprehensive plan for connecting essential components like water sensors, Arduino, and Ethernet shields, designing the system, establishing the process flow, and utilizing IFTTT for sending alert messages.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Overview

Chapter 3 covers the research methodology for developing a low-cost, smart water quality monitoring and control system. This chapter concentrates on the design and planning phases, detailing the integration of key components such as water sensors and Ethernet shields with the Arduino Mega microcontroller. It explains the system's overall architecture and the reasons behind these choices to ensure effective water quality management. Additionally, we include a flowchart to visually represent the design process and system workflow, giving a clear picture of how everything fits together.

3.2 System design

This section describes the components that go into creating water quality monitoring. It consists of the Arduino Mega microcontroller, which interfaces each water parameter sensor with the WI-FI connection module.

3.2.1 Arduino Mega Microcontroller

The Arduino microcontroller is simple to understand and program. Arduino is programmed using Arduino IDE. Arduino IDE is a tool used to write programs for Arduino boards. Arduino IDE is free software that is available to download and install on a computer. It has many ready-to-use libraries available in the Arduino IDE. In 2005, the Arduino was released. Arduino was first designed to let professionals and students create technology that could use sensors to interact with their surroundings (Trevathan et al., 2020a). The Arduino microcontroller AT mega pins is shown in Figure 3.1. The water sensor modules connect to Arduino to be able to gather data that is programmed on Arduino and to be controlled using actuators that connect to Arduino for better water quality. The features for the Arduino Microcontroller 2560 are shown in Table 3.4. The operating voltage for Arduino is 5 volts. The supply voltage is between 5 and 12 volts. It is advised to use 7-12 volts. Each of the 16 input pins on the Arduino Microcontroller 2560 offers 10 bits of resolution (Butuner & Uzun, n.d.).

Table 3.1 outlines the specifications of the ATmega2560 microcontroller that is used in Arduino boards. In addition to its versatility, this microcontroller is preferred for complex projects that require higher input-output capacity. All specifications are obtained from Arduino.cc and are shown in Appendix Y.

Table 3.1: Specifications for ATmega2560 Arduino Microcontroller

Microprocessor Specifications	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O pins	54(of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O pin	40mA
DC Current for 3.3V pin	50mA
Flash Memory	256 KB of which 8KB used by bootloader
SRAM	8KB
EEPROM	4KB
Clock Speed	16MHz

The ATmega2560 microcontroller is illustrated in Figure 3.1, which identifies all of its pins and their individual functions. The diagram specifically elucidates the different kinds of pins given as such: digital I/O pins, analog inputs, PWM outputs, and communication ports that are essential for microcontroller connectivity and potentials. This depiction aids in understanding the usage of every pin under various settings or applications.

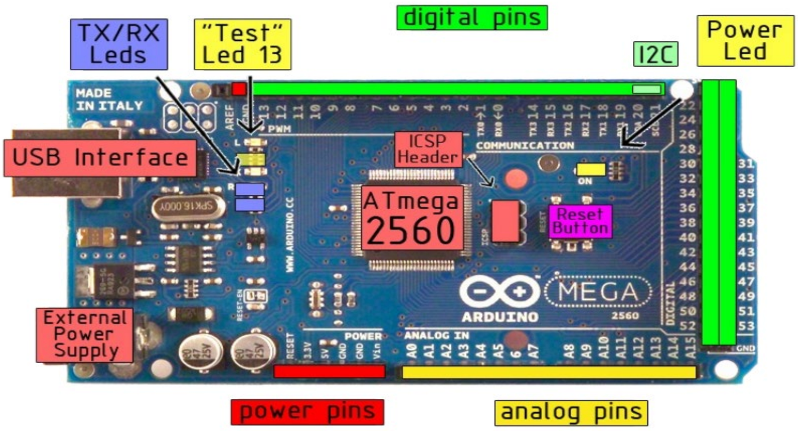


Figure 3.1: **Arduino Mega 2560 Microcontroller Board**

3.2.2 Interfacing the turbidity sensor to Arduino

As illustrated in Figure 3.2, it shows the gravity sensor schematic from DF Robot. This sensor can work in two modes: analog and digital. It utilizes an infrared (IR) LED to emit light across a solution, while an IR phototransistor measures how much of that light passes through it. The scattering effect is caused by the particles present in the solution, which causes a reduction in voltage as it is detected by the phototransistor; hence, turbidity is measured. In digital mode, turbidity changes are indicated in either 'yes' or 'no' but without giving an actual concentration level that informs us about turbidity levels in specific fluids. We need to recognize its analog behavior to understand more precisely those variables we are measuring using such sensors.

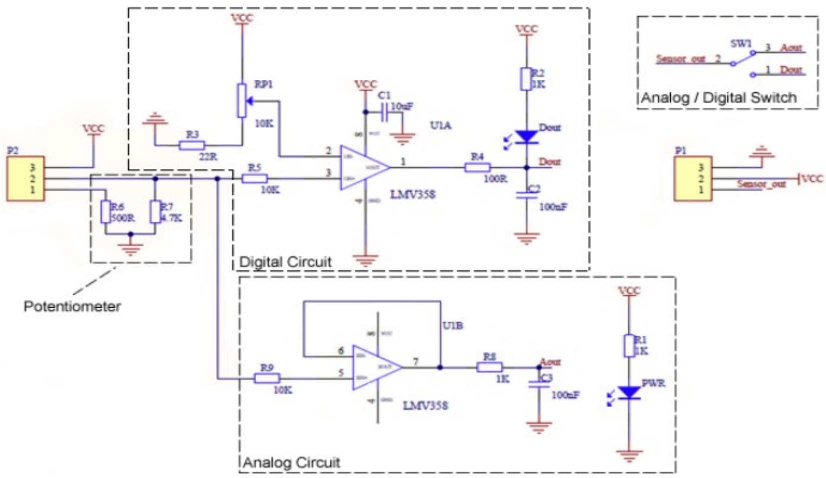


Figure 3.2: **DF Robot gravity analog turbidity sensor electronic schematic** (Trevathan et al., 2020).

Interfacing the turbidity sensor to Arduino involves several key connections. The experiment used the sensor 0189DfRobot gravity analogue, which detects water quality turbidity levels or invisibility. By monitoring light transparency and scattering rate, which alter the amount of total suspended solids (TSS), it uses light to identify suspended particles in water. The liquid turbidity level rises as TSS rises. The sensor has a sender and a receiver, which are, respectively, an LED and a phototransistor. The connection shown in Figure 3.3 is when interfacing the turbidity sensor to Arduino. It begins by connecting the sensor's ground (GND) pin on the ground of the Arduino pin and its power supply (VCC) pin 5V to the output on the Arduino. The next connection of the signal pin of an analog output sensor is connected to A0 on the Arduino, which is an input analog pin. Connection of the signal pin of a digital output sensor is connected to pin 2 on the Arduino. And the resistance is included between the signal pin and Arduino input pin (Trevathan et al., 2020) and (Anshori et al., n.d.).

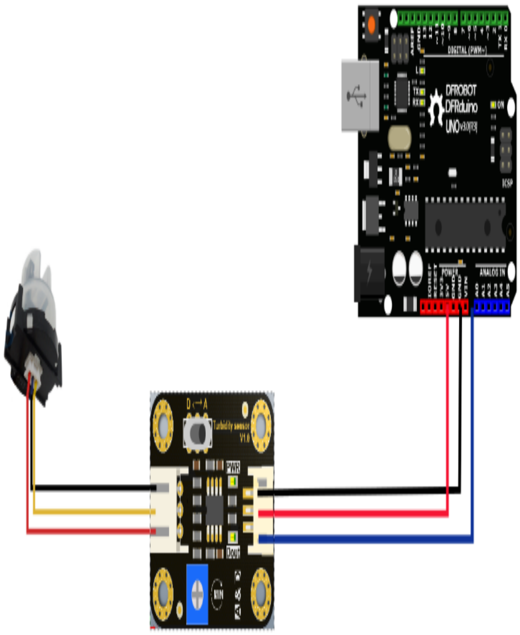


Figure 3.3: Interfacing the Turbidity Sensor to Arduino Microcontroller (Hakim et al., 2019)**.**

3.2.3 Interfacing the pH sensor to Arduino

pH has a variable supply voltage of 3.3≈5 volts, supported by the inbuilt voltage regular chip, which is compatible with both the 3.3- and 5-volt main control boards. The control board (Arduino microcontroller) and the software library pH meter can be implemented right away without the need of soldering, welding, or any other modifications. The pH connected to the Arduino microcontroller is shown in Figure 3.5. Under the process of calibration, the software library adopts the two calibration methods of a known number or range of pH, which automatically can identify two standard buffers of those known values (Shamsi et al., 2020).

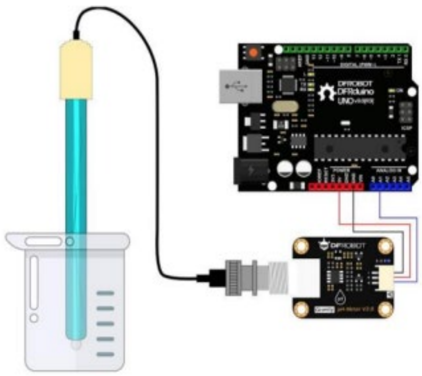


Figure 3.4: pH range and Interfacing pH Sensor Module to the Arduino Microcontroller (Samsudin et al., 2018)

3.2.4 Interface the Electrical Conductivity sensor to Arduino

The analog gravity electrical conductivity sensor meter V2 is used especially to measure the electrical conductivity of water or solution. It supports 3≈5 Volts input voltage. And it is compatible with 3.3V and 5V on a main control board (Arduino Microcontroller). The circuitry filters the output signal, resulting in less jitter. The excitation source uses an AC signal, which successfully lessens the polarization effect, increases accuracy, and extends the life of the probe. The electrical conductivity interface to Arduino, the VCC AND GND of the EC sensor output were connected to the VCC and GND of the Arduino input. The EC output terminal was connected to Arduino input analog 5, as shown in figure 3.6 below. During calibration for this EC sensor, two solutions are used, which are 12.88 mS/cm and 14.13mS/cm(Mukta Monira et al., 2019).

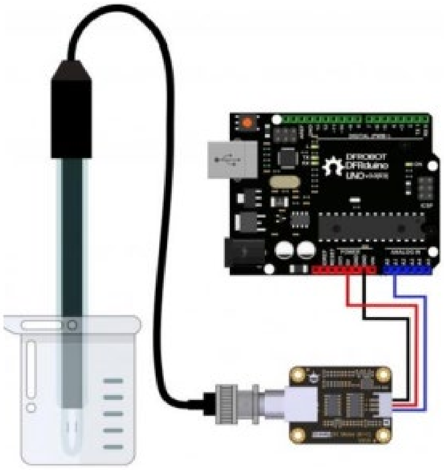


Figure 3.5: **Interface EC meter to Arduino** (Shamsi et al., 2020)**.**

3.2.5 Interfacing Total dissolve solids to Arduino

Total Dissolved Solids (TDS) are dissolved solids that may be detected in water, and this TDS sensor kit is made to measure TDS in water. It can be used for several things, including hydroponics and home water testing. It can be easily integrated with Arduino controllers as it has an analogy output (0 to 2.3V) and a wide voltage input range (3.3V to 5.5V). Probe life is extended, and signal output is stable when AC signal stimulation is used to prevent probe polarization. It's crucial to maintain the connector and signal transmitter board dry even though the waterproof probe can be submerged in water indefinitely (Handandi et al., 2024).as illustrated on Figure 3.7.

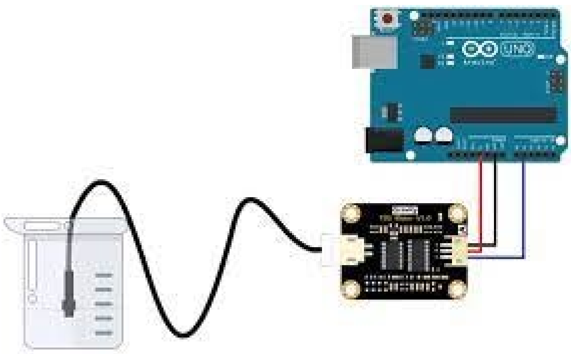


Figure 3.6: Total dissolve solids connected to Arduino.

3.2.6 Flow sensor

A water flows is a rotating impeller; an external housing, a magnetic core, and a hall-effect sensor make up a water flow sensor. When water flows through the rotor, it activates the magnetic core and rotates. By identifying the corresponding pulse signals that the hall-effect sensor produces, users may ascertain the flow speed. It works well for flow detection (Sapkal et al., 2019).

The following Specification for water flow sensor, were captured from the manufacturer of this sensor as shown on the Appendix Z.

Flow Sensor Specifications:

• Flow range:1-60L/min
• Interior diameter: 20mm (0.81")
• Maximum current: 15 mA (DC 5V)
• Working voltage range: DC 4.5-18 V
• Load capacity: 10 mA (DC 5V)
• Operating Temp: 80°C
• Operating humidity:35%-90%RH
• Water pressure < 1.20 Mpa
• Flow = 4.8 * units of flow (L / min) * time (seconds)
• Connector Type: Female 3-Pin JST-SMP

The connection of this water flow shown on Figure 3.8 to an Arduino, GND to GND, VCC to VCC and the signal pin of the water sensor to Arduino Digital pin.

Figure 3.7: Water Flow pump

3.2.7 Interfacing the Temperature Sensor to Arduino

The temperature sensor that is used has a high level of stability and accuracy. The DS18B20 sensor has an accurate and good submerged application. It provides accurate temperature readings that are essential, offering resolution and accuracy levels that satisfy the standard for water quality monitoring. It provides long-term dependability and is waterproof, which means it can be used in moisture environmental conditions (Mukta Monira et al., 2019).

The following specifications were captured from the manufacturer for this sensor.

Specifications

- Usable with 3.0V to 5.5V power/data
- ±0.5℃ Accuracy from -10℃ to +85℃
- Usable temperature range: -55 to 125℃ (-67°F to +257°F)
- 9-to-12-bit selectable resolution
- Uses 1-Wire interface- requires only one digital pin for communication.
- Unique 64-bit ID burned into chip.
- Multiple sensors can share one pin.
- Temperature-limit alarm system
- Query time is less than 750ms.
- 3 wires interface:
  - Red wire - VCC
  - Black wire - GND
  - Yellow wire - DATA
- Stainless steel tube 6mm diameter by 35mm long
- Cable diameter: 4mm
- Length: 90cm

This temperature sensor is connected to Arduino through a single data pin, and it sends digital data to an Arduino, and the data pin was pulled up by the high resistor, which is 4.7kΩ, and some jump wires, and this will allow an Arduino to be able to pull it low when it wants raw data from it. This temperature sensor has three wires, which are VCC, data, and GND; it connects to the Arduino's VCC, digital pin 2, and GND (Hermawan et al., 2023). The diagram is shown in Figure 3.9 below.

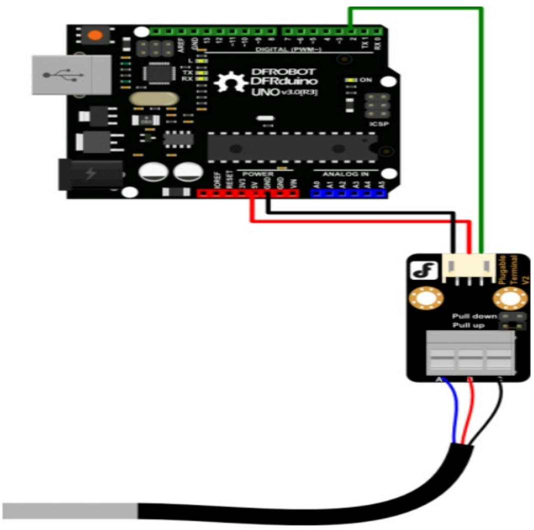


Figure 3.8: **Waterproof Temperature connected to Arduino.**

3.2.8 Breadboard, Wire and Resistor

In developing this project of smart monitoring and control system, other important components are breadboard, wire, and 4.7k ohm resistors. The breadboard is to easily test and prototype the circuit connections before finalizing the setup, making it easier to adjust. Wires are used to connect the Arduino, sensors, and other modules, enabling the system to transmit data and signals effectively. The 4.7K ohms resistor helps stabilize sensor reading and set the correct voltage for accurate measurement. These components work together to ensure the system functions correctly and reliably. As illustrated in Figure 3.10.

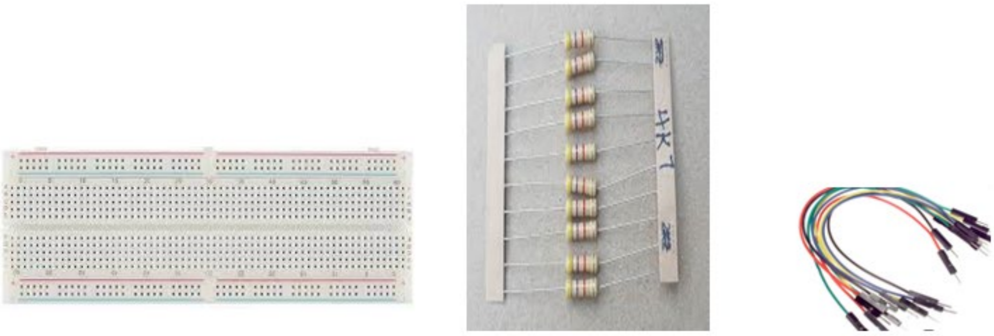


Figure 3.9: Breadboard, resistors, and wires.

3.2.9 Relay

The relay is necessary for this project's automated water quality control. Based on sensor readings, it functions as a switch that the Arduino controls to turn the DC pump on and off. The Arduino instructs the relay to turn on the pump when the sensors identify that metrics related to the quality of the water, like pH or turbidity, are outside of the intended range. This enables the system to change the water quality by automatically adding chemicals or carrying out other required tasks. By using the relay, the system can react to changes in water conditions instantly and without the need for human interaction (Rajput Singh Krishnapal et al., 2022).Figure 3.10 relay.

Specifications of a 5V relay are as follows: Normal voltage is 5V DC, Normal current is 70mA, Maximum AC load current is 10A at 250VAC or 125VAC, Maximum DC load current is 10A at 30V DC or 28V DC, contains 5 pins and is made of plastic. Operating time is 10 milliseconds, release time is 5 milliseconds, and maximum switching is 300 operations per minute.

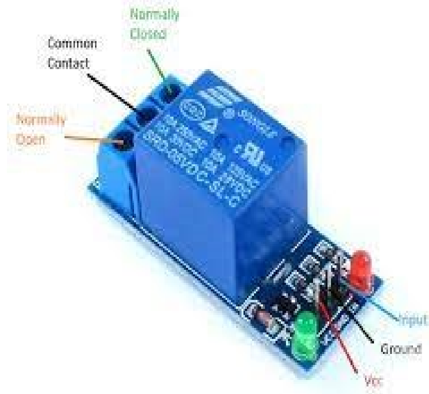


Figure 3.10: Relay

3.2.10 DC pump

The DC pump is important to keep the water quality in this project. The Arduino triggers the relay, which in turn powers the DC pump, when the sensors identify that parameters such as pH or turbidity are not within the intended range. The required chemicals or solutions are added by the pump to improve the quality of the water. For instance, it might change the pH levels or add chlorine to sanitize the water. The system is more dependable and efficient because of this automated procedure, which guarantees that the water quality is continuously maintained without requiring human involvement (Rajput Singh Krishnapal et al., 2022).

Technical Specification

Input Voltage: DC 3V – 5V

Flow Rate: 1.2 – 1.6 L/min

Operating Current: 0.1 – 0.2A

Maximum Suction Distance: 0.8m

Outside Diameter of Water Outlet: 7.5mm

Inside Diameter of Water outlet: 5.0mm

Diameter of Water Inlet: 5.0mm

Wire Length: 200mm

Operating temperature: less than 80˚C

Figure 3.11 illustrate the Dc pump that will be used on this project.

Figure 3.11: DC pump

3.2.11 Pipe

Pipes are employed in this project to move chemicals and water around the system. The Arduino sets off the relay to activate the DC pump when sensors identify those factors related to water quality, such as turbidity or pH, are not within the intended range. After that, the pump forces the required chemicals into the water to improve its quality via the pipes. By enabling the system to react swiftly to any detected changes, the pipes guarantee that these chemicals are delivered precisely and effectively, contributing to the maintenance of constant water quality, and the pipes, which are 1 meter of clear plastic tubing (8mm will be used, as shown in Figure 3.12.

Figure 3.12: Pipe

3.2.12 Lithium batteries and 18650 Holder with a Charger

The specification for the 18650 holders with a charger is shown on table 3.2.
The key specifications for this shield are as following:

Table 3.2: Specifications for 18650 holder with a charger

Specifications for the charger	
Battery holder	Accommodates two lithium-ion batteries

Input Voltage	5V on the USB or C type port
Output Voltage	5V
Protection feature	Over-charging, Overcurrent, over discharged and short circuit protection.
Charging current	1A and more
Boost converter Efficiency	90%
LED indicator	Display for charging and fully charged
Output port	USB ports and Pin headers

A battery 18650 shield with two cells is a practical, user-friendly module designed to hold and utilize two lithium-ion batteries. To ensure that the batteries run securely, these shields are outfitted with important safety features like protection circuits that stop overcharge, overdischarge, overcurrent, and short circuits. They frequently come with a charging module that has a Type-C or micro-USB connector, which makes it simple to recharge the batteries inside the shield. Furthermore, a boost converter that produces a steady 5V output perfect for powering microcontrollers and other electronic components is usually included in these shields. The shield's LED indications provide a quick visual reference by showing the battery levels and charging status. Connecting the shield to other projects and devices is made simple by its output connections, which include USB connectors and pin headers. Because of these characteristics, 18650 battery shields are ideal for applications where dependable and effective power control is important, and to use a battery 18650 shield for this project of water quality monitoring systems will help a lot, as illustrated in Figure 3.14.

Table 3.3: **Specifications for lithium batteries**

Specifications for the Li-ion battery	
Voltage of cell	3.7 volts
Nominal capacity	3000 mAh
Electrochemical material	Lithium ion
Size	L=49 x D =14 mm
Rechargeable	True

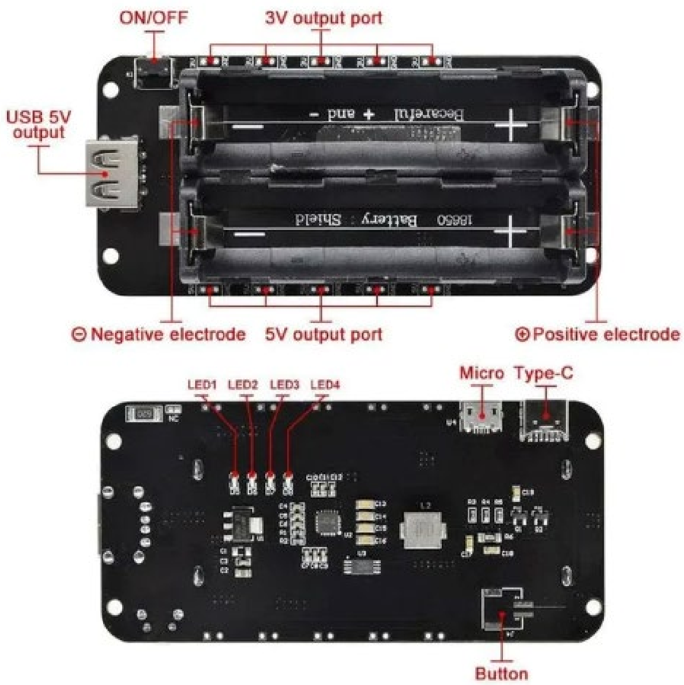


Figure 3.13: **18650 2 Battery Shields**

For figure 3.15, two lithium rechargeable batteries in 18650 sizes with a 3000mAh nominal capacity, reusable design that saves money and reduces waste for repeated use. High energy density for extended battery life in electrical gadgets like e-cigarettes and lamp vaporizers with built-in safeguards against short circuits, overcharging, and over discharge, this device is dependable and safe. This 18650 3000mAh rechargeable battery provides an efficient and sustainable way to power your gadgets. Its strong energy density and dependable safety features allow you to have uninterrupted power for all your preferred devices.

Figure 3.14: **Rechargeable lithium batteries**

3.3 Interface Ethernet shield to Arduino

The Ethernet shield is commonly used in Internet of Things projects to connect an Arduino to the internet since this Arduino does not have a built-in network capability. It allows Arduino Mega to send data to a cloud server via Ethernet cable. The shield has an Ethernet port and communicates with the Arduino using SPI (serial peripheral interface).

The connection on Figure 3.16 is the Ethernet shield is connected to Arduino; to set it up, connect the shield's VCC to Arduino’s 5V and GND to GND. Then link the SPI pins: MISO to pin 12, MOSI to pin 11, and SCK to pin 13 on the Arduino. Also, the SS pin on the shield to pin 10 on the Arduino. These connections enable the Arduino and Ethernet shield to communicate. Using the Ethernet library in the Arduino IDE. The code for this connection of an Ethernet shield to Arduino is in Appendix A. With this set up, the Arduino can perform various tasks like serving web pages, sending sensor data to cloud services, and receiving remote commands. For example, in this water monitoring project. The Arduino will use an Ethernet shield to upload sensor readings to a platform called ThingSpeak. This setup will make it easier to monitor and control in real time, enhancing the capabilities of the IoT project.

Figure 3.15: Ethernet shield connected to Arduino.

3.4 Flow Chart of the system

Figure 3.17 shows how the water quality monitoring system works. The first step is to switch on the control system as well as fix a network between the Arduino microcontroller and the internet. The next step is to initialize the water quality sensors to ensure accurate measurements. The following step involves reading real-time data from these sensors using Arduino and displaying it on the computer screen via the Arduino IDE’s serial monitor. Additionally, a cloud site such as ThingSpeak receives these results concurrently with the reading of other data, enabling remote control. Compare the sensors' readings with preset limits to determine what constitutes clean water. If they fall within acceptable limits, normal monitoring continues without any interruptions. Conversely, if there are no acceptable limits, WhatsApp sends an alarm message displaying disturbing statistics. After this notification has been communicated, the user receives more details about that issue through reporting and analysis, which can either go on continuously if the system has been designed accordingly or stop at any given point.

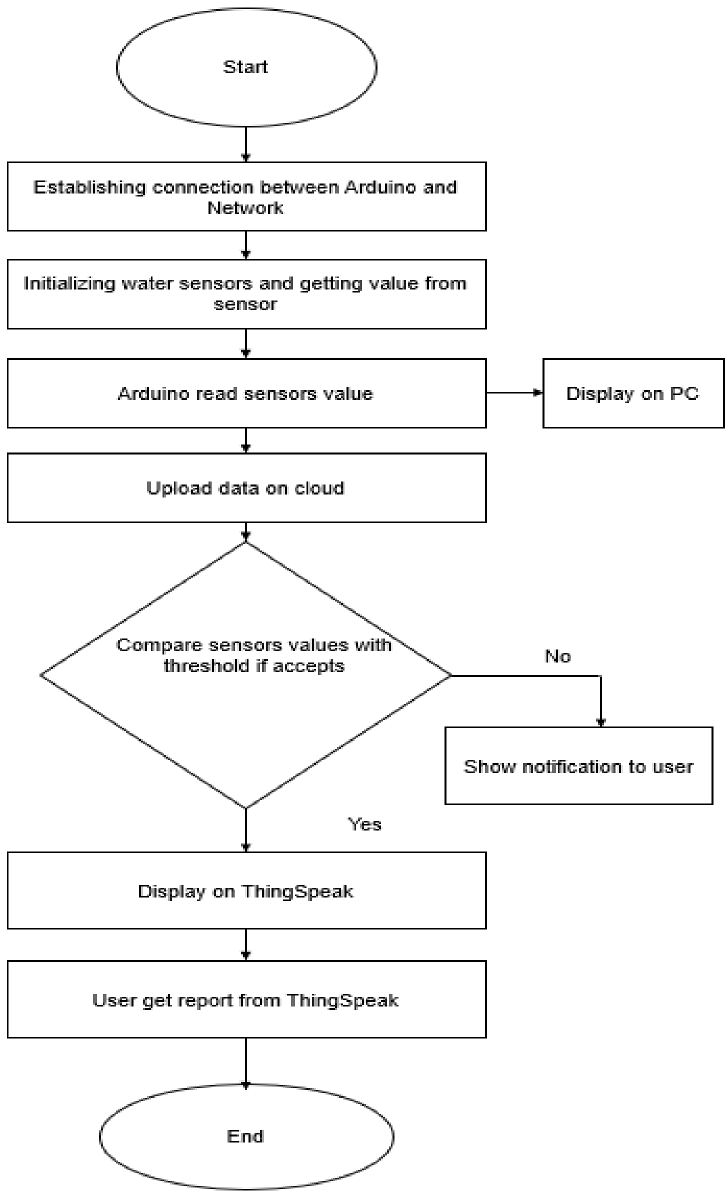


Figure 3.16: **Flowchart of the System**

3.5 Control system

The smart water quality monitoring system includes a control feature to manage different water conditions. It adjusts the pH levels by adding solutions such as white vinegar to lower the pH and baking powder to raise it. When the pH readings fall outside the desired range, an Arduino microcontroller manages the DC pump to add these solutions.

Because the Arduino can't directly power the DC pump, we use a relay circuit to control it. This setup allows the Arduino to operate the pump safely and effectively. The system automatically dispenses the chemicals based on real-time sensor data, ensuring that the water remains close to neutral. Figure 3.18 illustrates the integration of the DC pump, relay, and batteries.

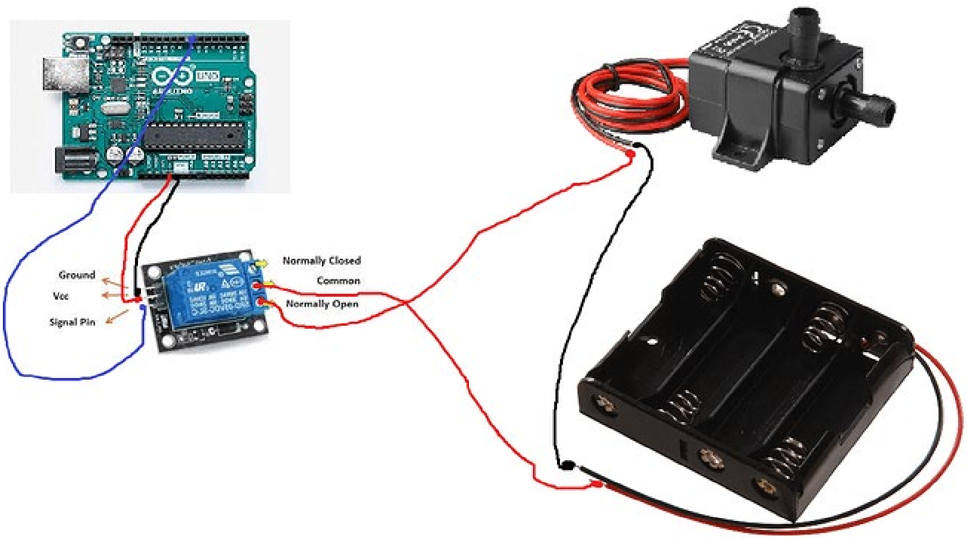


Figure 3.17: **Connection of DC Pump, Relay and batteries**

3.6 System Architecture

Figure 3.19 illustrates how the architect uses Internet of Things (IoT) models to make a water quality monitoring system low-cost and smart, enabling real-time measurement, analysis, and control of water quality. This system contains almost all the necessary hardware and software components, enabling effective monitoring, management, and control of water quality. Within this hardware setup, several types of sensors are essential to the measurement of different facets of water quality. A turbidity sensor evaluates the clarity of water by measuring the detectability of suspended particles; a TDS (Total Dissolved Solids) sensor establishes the concentrations of dissolved substances; a pH sensor tracks the acidity or alkalinity in aqueous solutions; an electrical conductivity (EC) sensor assesses ion concentration; a flow sensor tracks the water's circulating rate; and a temperature sensor, compensated with an EC, records the ambient temperature of the water.

An Arduino microcontroller serves as the main processing unit for all these devices. This device gathers data from other devices and processes it based on predetermined logic, thereby enabling communication with other system components. The system makes use of an Ethernet shield to connect Arduino boards to the cloud, thereby transmitting instantaneous analyses and visualizations of sensor parameters on the ThingSpeak cloud. Moreover, its integration with IFTTT, which sends notifications directly to your phone, triggers alarms when water quality deviations from specified values occur. In addition to its monitoring functions, this device also operates a relay-operated DC pump, which automatically adds treatment chemicals such as chlorine or soda ash as needed, thereby maintaining water quality without the need for manual intervention. It can even operate in faraway locations because it works using lithium-ion batteries. This arrangement ensures effective management of water quality thanks to frequent automated monitoring and alerts, which are accessible from any place around planet Earth.

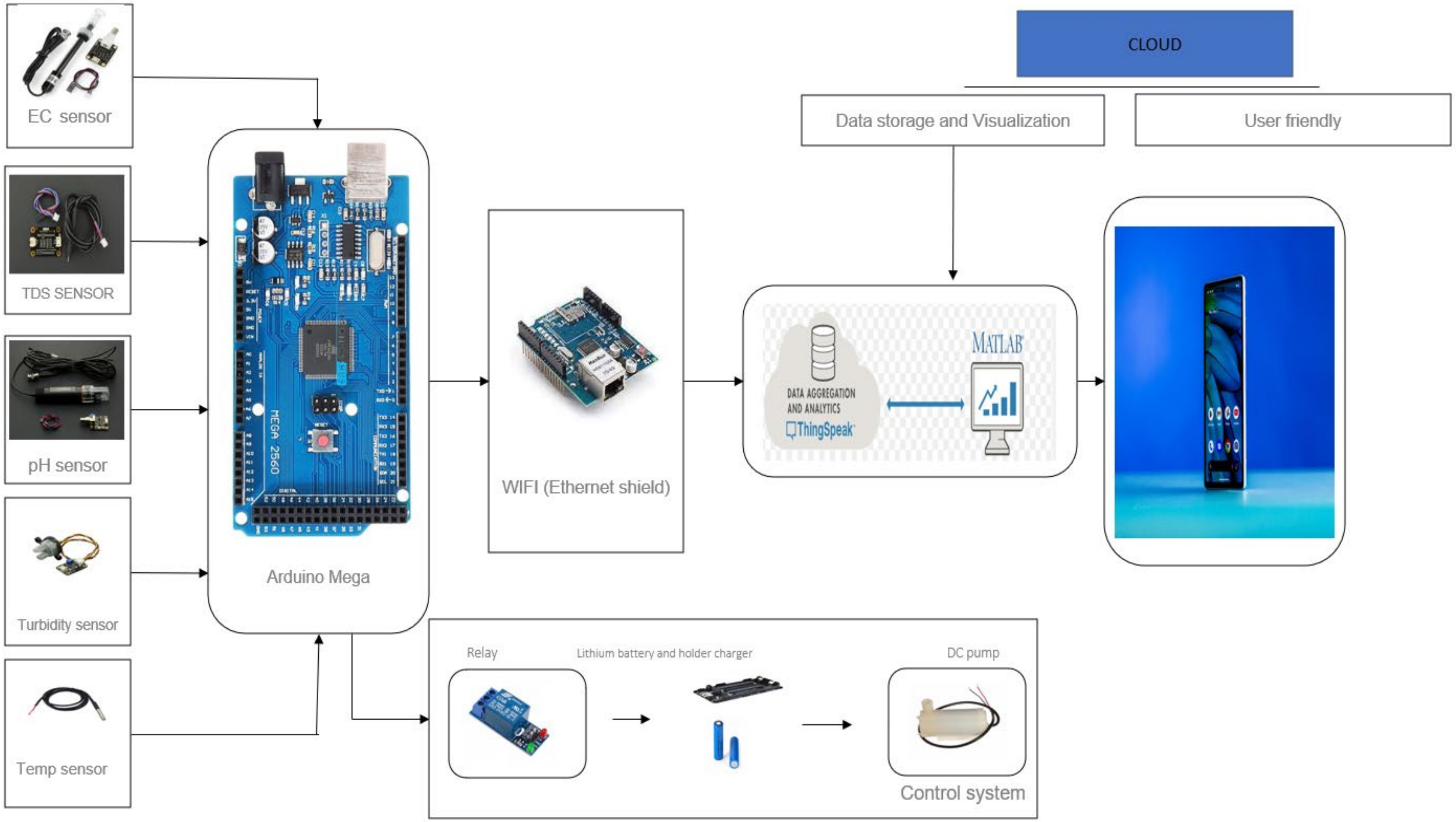


Figure 3.18: Architecture of the system.

3.7 Raw data collection

In the current era of advanced technology, raw data is vital for understanding environmental conditions. For our low-cost smart water quality monitoring system, the Arduino microcontroller plays a key role in managing this data. The Arduino gathers information from sensors that measure electrical conductivity (EC), total dissolved solids (TDS), turbidity, pH, temperature, and water flow. This data is then sent to ThingSpeak through an Ethernet shield, which acts as the API server in our system. ThingSpeak, a cloud-based platform, securely stores and processes the data in real time. HTTP requests transmit the data, and each field in a ThingSpeak channel corresponds to a specific sensor reading. This guarantees the preservation and accessibility of the data. Therefore, the system integrates IFTTT (If This Then That) to provide automated notifications based on the sensor readings. This feature sends alerts when readings move outside of predefined thresholds, enabling timely intervention and effective water quality management.

3.8 Data Pre-processing

In this water quality monitoring system, accurate data handling is crucial. The first step is checking sensor readings for things like electrical conductivity, total dissolved solids, turbidity, pH, temperature, and water flow. If we find missing data, we fill in the gaps by averaging nearby readings to keep the data reliable. The research must correct any unusual values, such as negative numbers, that could potentially be errors. The researcher must follow the manufacturer's guidelines to ensure our sensor readings are accurate. This means using the right formulas and calibration methods provided by the manufacturer to turn raw data into useful measurements. This can be achieved by cleaning the data, adhering to these guidelines, and ensuring that the information sent to ThingSpeak is precise. This accuracy helps the system make good decisions and manage water quality effectively.

3.9 Summary

To sum up, Chapter 3 has detailed the research methodology to develop the inexpensive, smart water quality monitoring and control system. Specifically, through careful design and planning, this chapter has illustrated the integration of the chief components, which are water sensors and an Ethernet shield, with the Arduino microcontroller, which are at the heart of its architecture. The chapter has provided a comprehensive overview of this system’s flowchart, control mechanisms, and methods for raw data collection and preprocessing. The researcher has made these decisions to maximize the system's efficiency and ensure reliable water quality management. Chapter three provides a thorough explanation of each choice, enabling readers to comprehend how their combination contributes to overall functionality. This understanding paves the way for the practical implementation that follows in the subsequent chapters.

CHAPTER FOUR
ENGINEERING SYSTEM DESIGN

4.1 Overview

This section focuses on the set-up, configuration, and testing of each functionality component that makes up the entire automated, smart monitoring and control system for water quality. The Appendices contain the system codes, while Chapter 3 details the connections between each sensor. The combination of these different electronic components and software applications collectively forms a prototype of a smart monitoring and control system. This study divided the section into five components and explained each one individually in detail. When appropriate, the study conducted a test demonstration to illustrate how we achieved the objectives of designing the automated water quality monitoring and control system. The components of the water monitoring and control system are listed below:

Component 1: Sensor Integration

Component 2: Cloud Integration

Component 3: Notification Alert

Component 4: user-friendly interface

Component 5: Final Design

4.2 Arduino IDE Program

This study utilized the Arduino IDE for creating, compiling, and uploading data to the Arduino board. Beginners can easily use the Arduino IDE to develop projects with Arduino microcontrollers that interact with a variety of sensors and components, and it also offers advanced features for skilled users. In this cost-effective intelligent monitoring and control system for analysing water quality measurement, the Arduino IDE played an integral part. To set up the system, first download the official website of Arduino and install it onto your computer. Once the installation was complete, the IDE displayed the toolbar at the top, as shown in figure 4.1. Key functions carried out by this toolbar include verifying code errors to detect possible mistakes, uploading code to the Arduino Board, saving projects, and accessing a serial monitor to view board data.

The keystone of the IDE is the sketch editor, or code editor as it is popularly known, where one can write a program on either C or C++. In this project, the sketch editor interfaced various water quality sensors, such as pH, turbidity, electrical conductivity (EC), total dissolved solids (TDS), water flow, and temperature, enabling program writing. Because coding may be tedious and challenging to read in block letters all the time, the editor indents automatically and highlights syntax, aiding comprehensibility. This meant that the researcher had to set up all sensors using variables that could receive data and then make decisions using conditional statements based on these data. For instance, messages on errors show up at the bottom of the IDE during compilation and upload processes about which you know nothing, including what should have gone wrong; hence, they make debugging much easier.

In general, the Arduino IDE made programming Arduino boards easy, which in turn simplified the design and testing of this water quality control and monitoring system. This system employed an approach that involved writing programs to manage sensor data, control DC pump operation, and link with visualization platforms like ThingSpeak or IFTTT to receive alerts. This allowed for automatic water quality regulation and real-time data processing and treatment from the sensors' outputs.

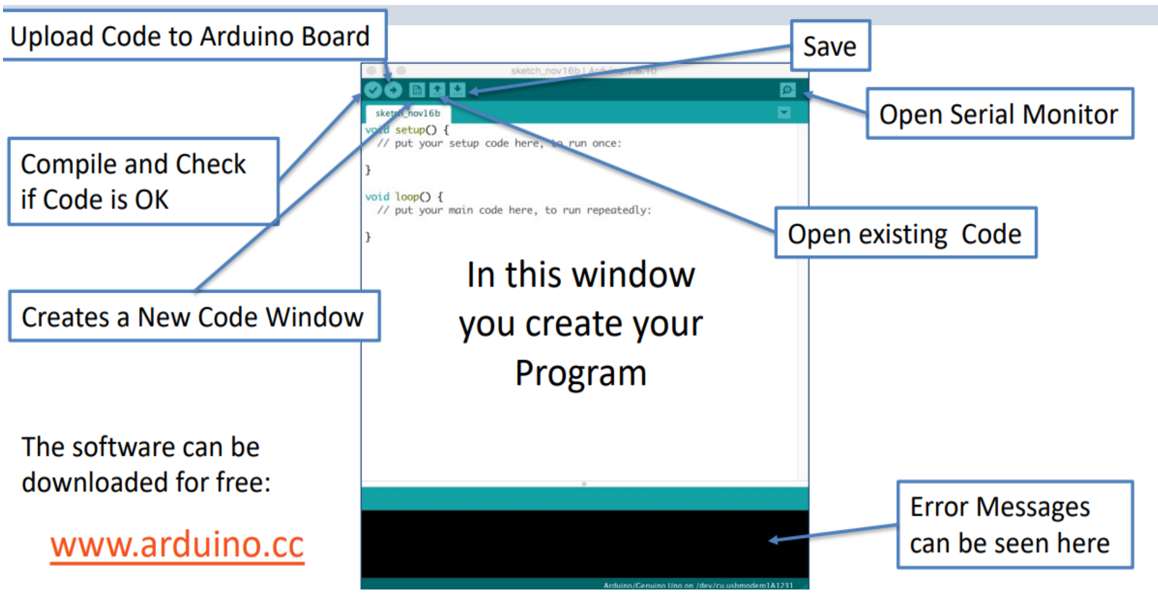


Figure 4.1: Arduino IDE

4.3 Sensor integration

4.3.1 Water quality sensor setup

This section focuses on the calibration, which is described using a flowchart, testing, and validation of the water quality sensor used for this study, and the water sensors integrated with Arduino connections are covered in detail in Chapter 3, Section 3.2.

4.3.2 Calibration Process of TDS

This project used the gravity analog TDS sensor and the Arduino Mega microcontroller to measure the TDS levels in the water. This sensor is helpful for the analysis of drinking water and other fluids. Chapter 3, Figure 3.7, explains the integration of the TDS sensor with Arduino and its functions. Figure 4.2 illustrates the flowchart of the calibration process. Calibrating a TDS sensor involves forming a connection between known TDS values and the sensor’s analog voltage output. The calibration begins with tap water known to have a certain TDS. Initially, the researcher read the sensor's analog voltage output and converted it to a corresponding voltage measurement. First voltage readings must be trusted. To ensure accuracy, the researcher spaced it out and divided the known TDS value by the measured voltage. So, this method gives you a way to change sensor voltages into TDS values that are given in parts per million (ppm), which makes sure that the data you get from water quality monitoring systems is correct.

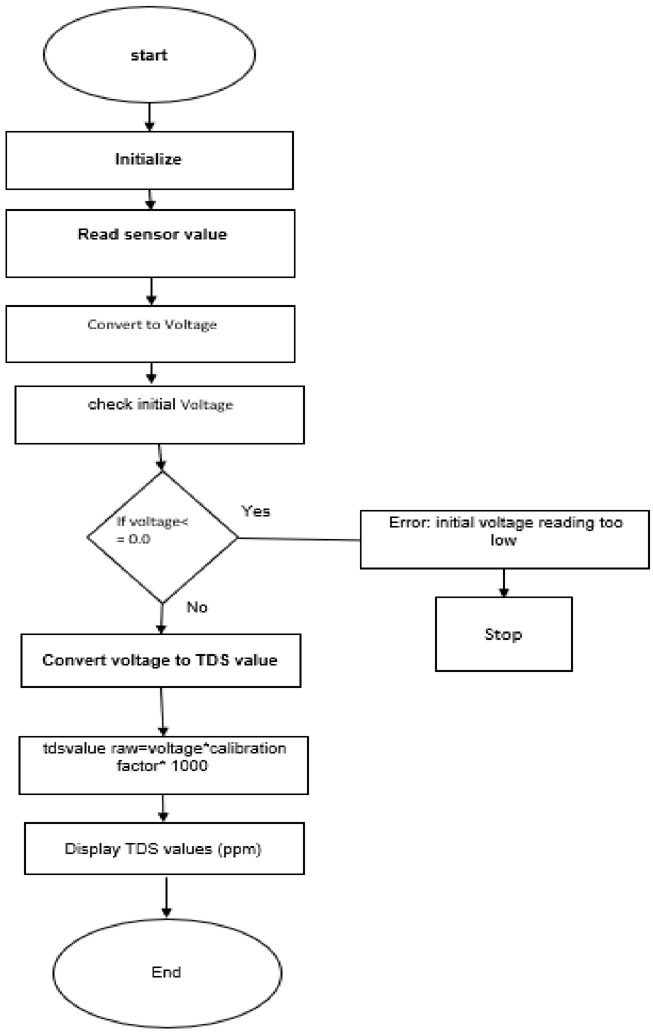


Figure 4.2: **TDS Calibration Flowchart**

4.3.3 Calibration Process of Analog electrical conductivity (EC)

In Chapter 3, Figure 3.6 provides a detailed account of how to calibrate the analog electrical conductivity (EC) sensor for this project. Important components included in this calibration process are an EC sensor, Arduino board, and standard calibration solutions (1413 μS/cm and 12.88 mS/cm). This sensor is temperature compensated; hence, when calibration was done, it also included temperature, as therefore, when we wanted calibration results for both sensors, they were printed at the same time. The researcher connected the EC sensor to the analog pin (A0) on Arduino and the temperature to the digital pin (D2). Initially, the researcher took some temperature readings from the temperature sensor to initiate the calibration, as DFRobot's algorithm requires proper temperature compensation for accurate EC measurements. Therefore, DFRobot's precautions render the elimination process unnecessary, as they enable more accurate EC readings that accurately reflect the characteristics of each solution, regardless of temperature fluctuations.

The process of calibration is illustrated in Fig. 4.3, where first the sensor pins are initialized and known calibration values (1413μS/cm and 12.88 mS/cm) are set. Next, the researcher dips the sensor into the buffer solution and records the initial EC values. Next, the researcher calculates the voltage output from the sensor; if it is less than or equal to zero, an error message stating "Voltage reading is too low for calibration" will appear, causing the program to halt. However, if the voltage output exceeds zero, we proceed to calculate a calibration factor using the formula factor = known EC value / voltage. Finally, it displays the calculated EC value and completes the procedure.

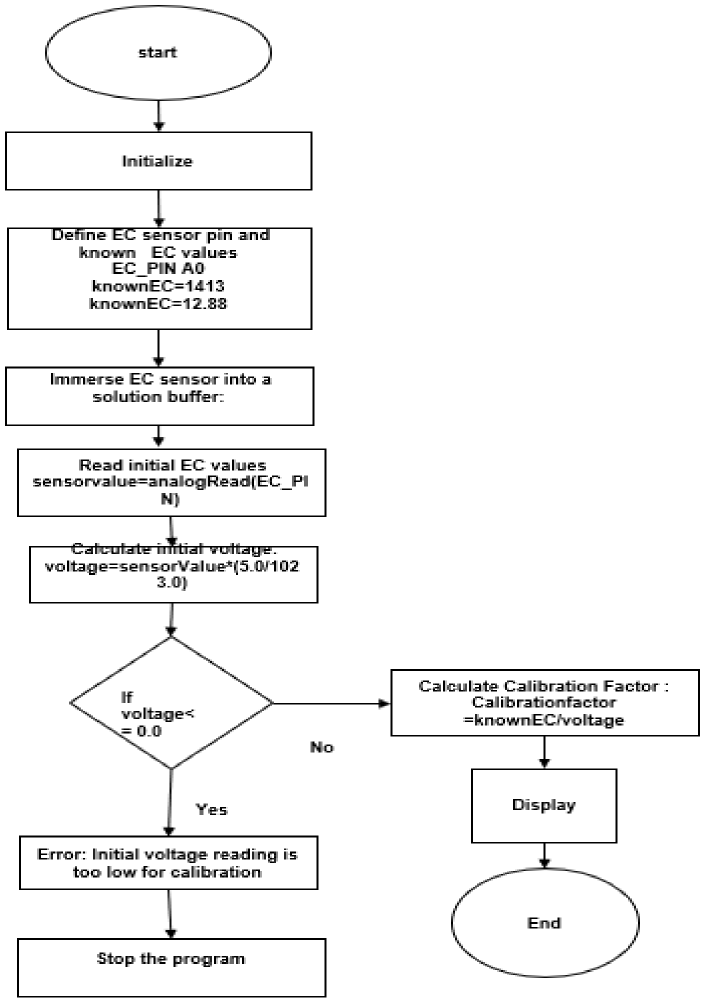


Figure 4.3: **Flowchart of Calibration process of EC sensor**

4.3.4 The BMT Temperature Probe

The BMT Temperature Probe DS18B20 was connected to digital pin 2 for calibration of temperature sensors. Figure 4.4 illustrates a flowchart of temperature. The DS18B20 temperature probe was initialized using the Dallas Temperature library, and this gadget was calibrated using an Arduino device. To display temperature readings, start serial transmission at 9600 baud rates in the setup function. The loop function will call and retrieve the Celsius temperature reading by using method {getTempCByIndex (0)} before sending it out through serial monitor. The position or code was altered to compare these digits against one from an accurate and renowned thermometer to confirm correctness since there were some differences when presenting accurate figures on an Arduino monitor.

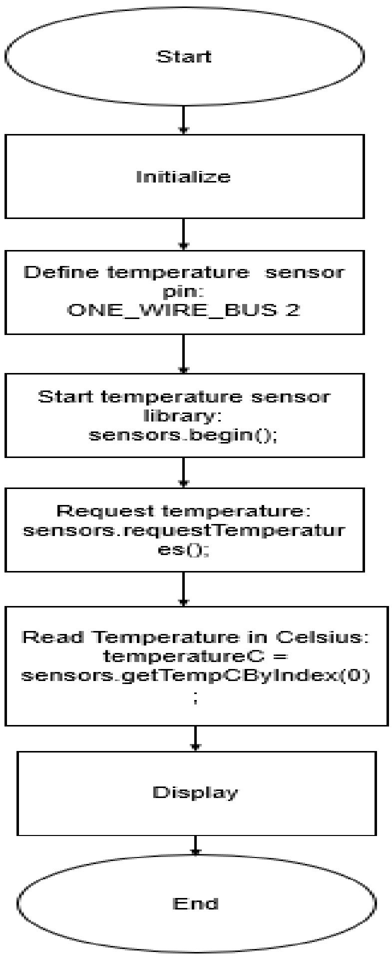


Figure 4.4: **Calibration of Temperature sensor.**

4.3.5 The process of calibrating the turbidity sensor

In the beginning of this project, the researcher calibrated the SENO189-DFRobot turbidity sensor by connecting the turbidity analog A1 to the Arduino, as shown in Figure

3.3 of Chapter 3 and Figure 4.5. The next step involved communicating through the serial port at a baud rate of 9600. The code repeatedly calls the'read_tur ()' function in the main loop, handling readings and calculating the turbidity. The sensor's voltage readings take up to 810 analog readings, convert each to a voltage using a reference value, and then average these readings. Then average voltage determined the turbidity. If the voltage falls below 2.5 V, set a high turbidity level, such as 3000 NTU, using a formula that calculates turbidity in NTU. The researcher then printed the voltage and turbidity values on the serial monitor. The researcher then compared the results with the manufacturer's known values to ensure accuracy. This is supposed to regularly calibrate the turbidity sensor to ensure reliable measurement over time; the code is available in Appendices M, N, and O.

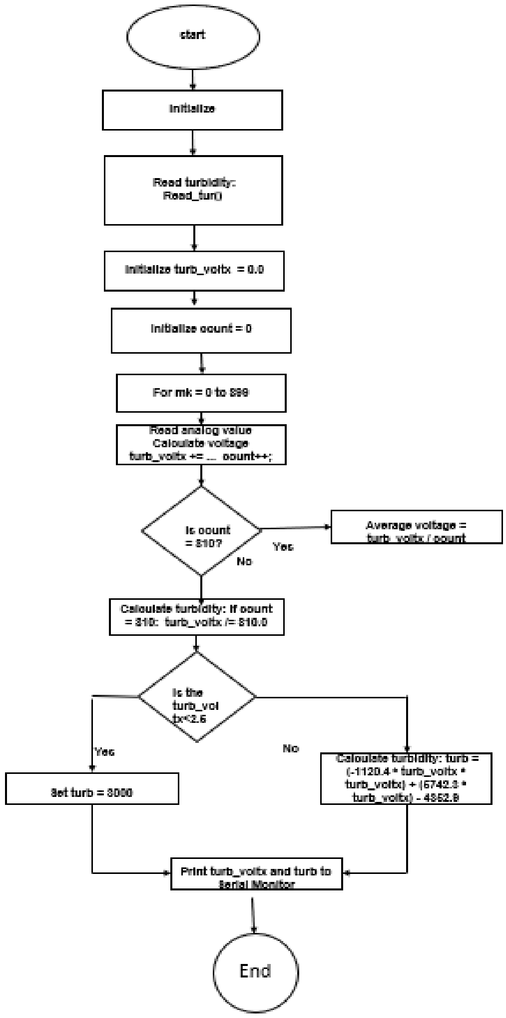


Figure 4.5: **Flowchart of the calibration Turbidity sensor**

4.3.6 The process of calibrating the pH sensor

Figure 4.6 presents the flowchart for calibrating the pH sensor. The initial step involves setting up the sensor and its components. The sensor subsequently reads the voltage from the pH probe and transforms it into pH values. The system determines if the voltage is excessively high or extremely low. If the voltage falls outside the required range, the researcher calculates the pH based on that specific reading; if the voltage falls within the expected range, the researcher applies standard practice for determining pH values based on voltage readings. Meanwhile, when a voltage does not lie within acceptable limits, the researcher equated the measurement to zero (0). Subsequently, the calculated pH value displays this information. Ultimately, ensure the accuracy of the measuring device's readings by making any necessary adjustments to the calibration parameter before ending the process.

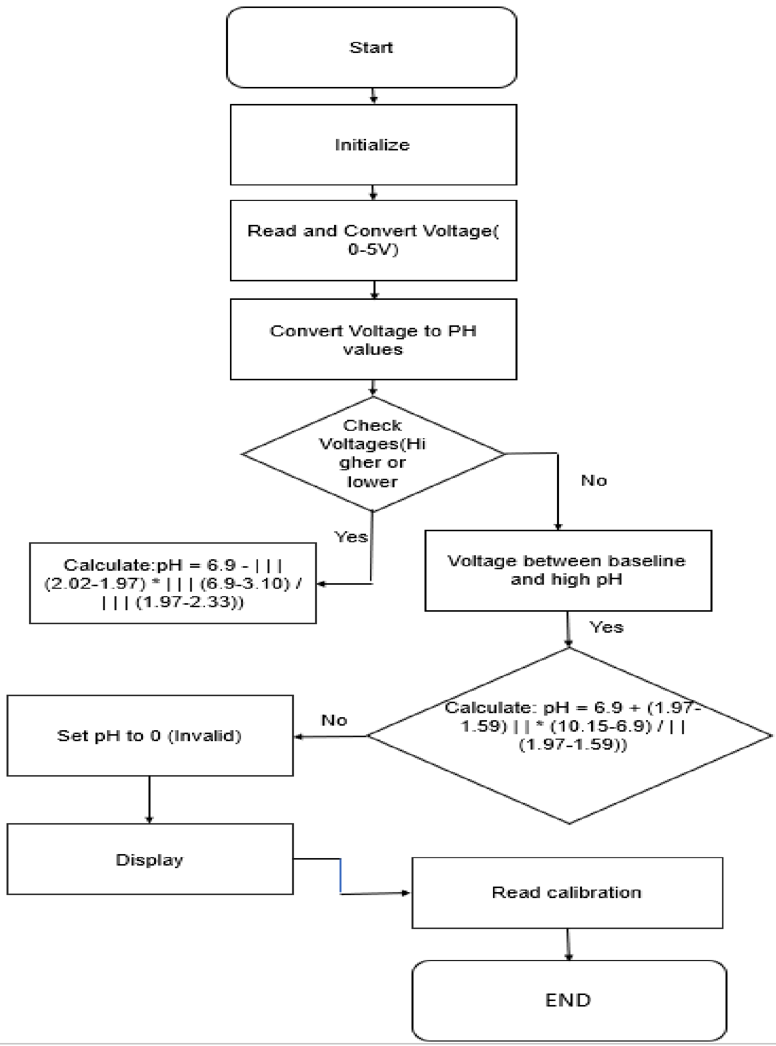


Figure 4.6: Flowchart Calibration of pH sensor

4.3.7 Water flow

Figure 4.7 illustrates the calibration of the flow water sensor. Initially, the researcher set up the calibration procedure by preparing a 1-liter container to capture the water flow and ensuring the area was suitable for measurement. Additionally, the researcher placed the flow water sensor accurately and connected it to the Arduino, following the instructions in the appendix code. The researcher then uploaded the calibration code to the Arduino, which utilized a volume formula to convert pulse counts into liters per minute flow rates. The next step is to let the water pass through this sensor and flow into the container. The Arduino's serial monitor recorded the sensor's readings. The researcher filled the container and timed the duration until it reached one liter to determine the actual flow rate. The researcher for this study compared the measurement with the calculated flow rate of the sensor and the calibration factor. The researcher disconnected the sensor once the readings were accurate.

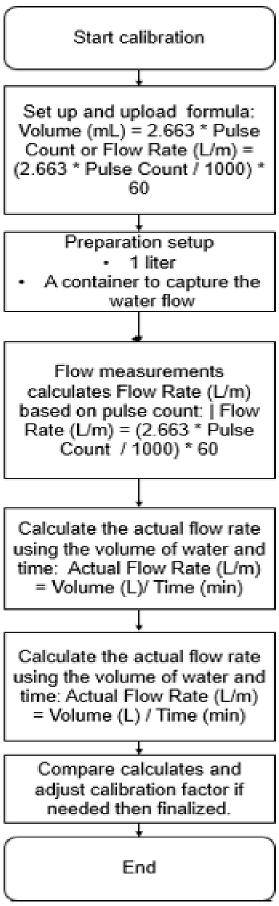


Figure 4.7: **Flowchart of calibration the Water Flow**

4.4 Cloud integration

This study integrated ThingSpeak with Arduino for an automated monitoring and control system. A few structured steps were followed. First, the study began by creating the ThingSpeak account on the website. After signing up, the researcher navigated to "Channels," chose My Channel, and entered all the required information, including the name of the channel and the parameters for the water sensors used in this study, like pH, temperature, and turbidity. The final step involved saving the channel, as illustrated in Figure 4.8, to complete its creation.

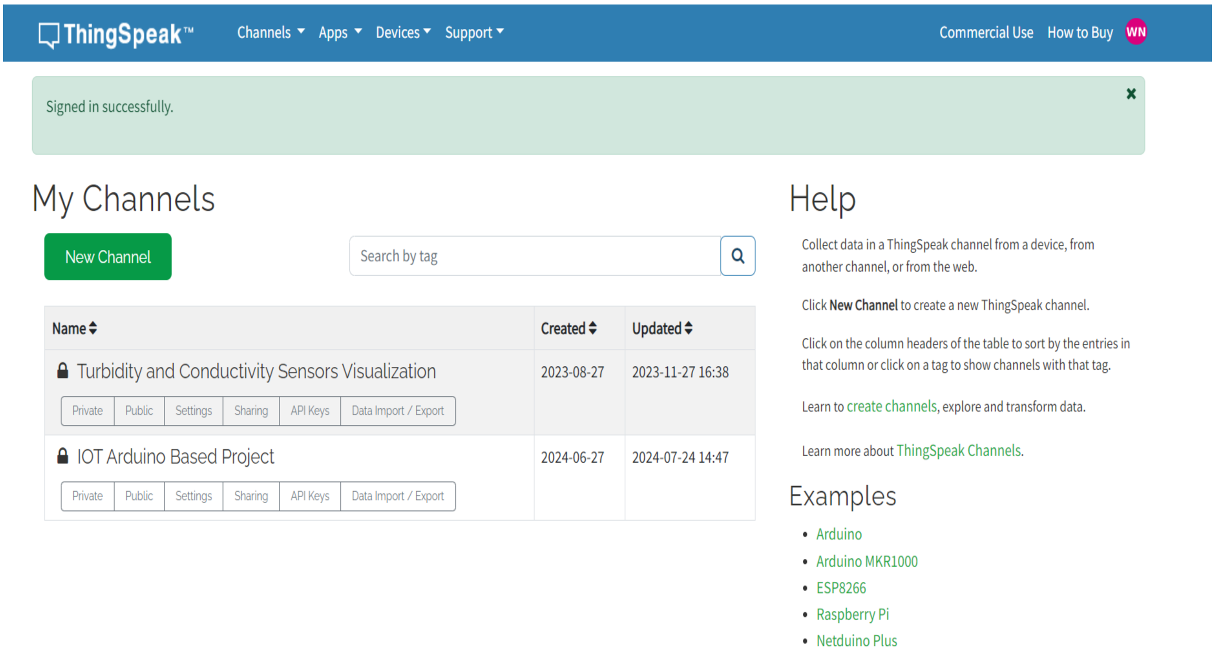


Figure 4.8: Channel

The researcher for this study selected and pasted the Channel ID and API key write key from the data communication flow between ThingSpeak and Arduino onto the Arduino IDE code, enabling data transmission to ThingSpeak, as illustrated in Figure 4.9.

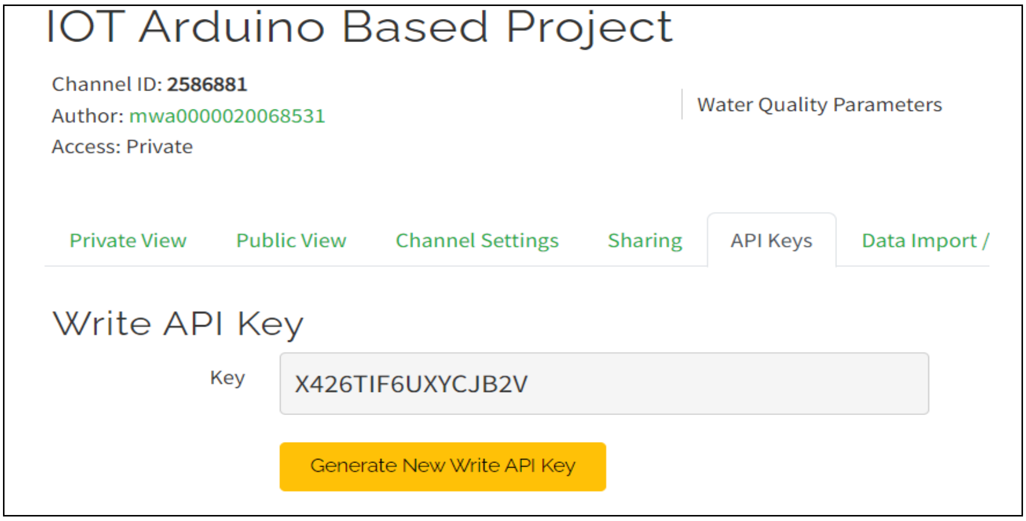


Figure 4.9: **Channel ID and API key**

As illustrated in Figure 4.10. After preparing the Arduino environment, the researcher for this study set up the ThingSpeak channel. Launch the Arduino IDE and add the necessary libraries. Launch "Sketch" > "Include Library" > "Manage Libraries," then install the Ethernet library for the Ethernet shield and the "ThingSpeak" library, as this project requires an Ethernet connection between the Arduino and ThingSpeak. Chapter 3, Figure 3.16, illustrates the connection between the Arduino and Ethernet shield.

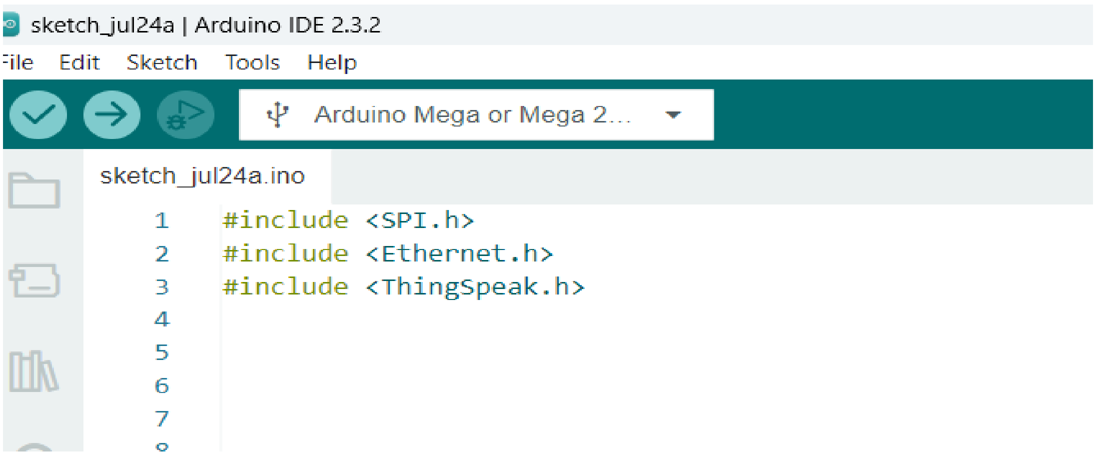


Figure 4.10: **Include libraries (Ethernet and ThingSpeak)**

The figure 4.11 illustrates on to set up the ThingSpeak and Ethernet credentials.

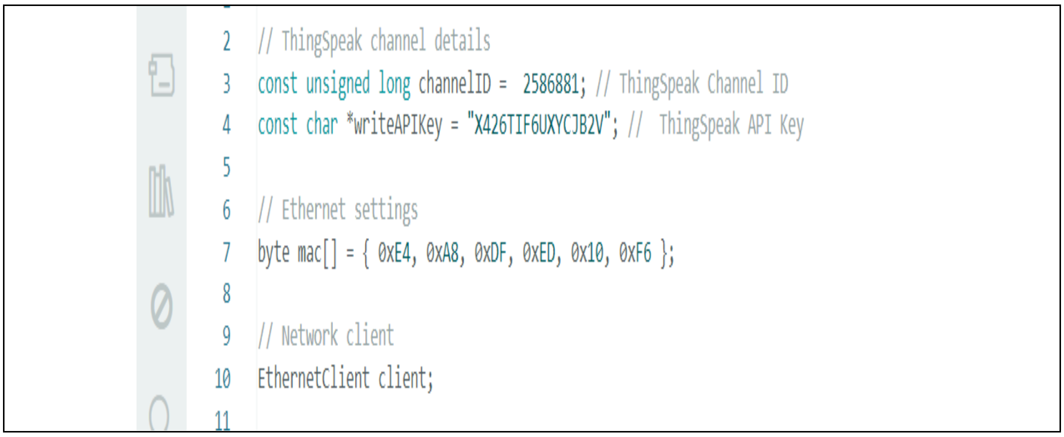


Figure 4.11: **Credentials of ThingSpeak and Internet**

Ultimately, the code was created to read sensor data and send it to ThingSpeak within the loop function. For example, one can read the results from turbidity and pH sensors by using analogRead (). These readings were assigned to the appropriate fields from the ThingSpeak channel using the ThingSpeak.setField() function. ThingSpeak.writeFields() was called to transmit the data to the channel once all the fields had been set. A delay to regulate the frequency of data updates was one of the features. By following all the important steps to integrate the cloud (ThingSpeak), we will enable real-time efficient monitoring and control. And be able to visualize data from sensors; the code is in Appendices A, B, and C.

4.5 IFTTT Integrated with ThingSpeak.

The making of an IFTTT applet using ThingSpeak is about selecting a triggering event then defining the resulting action between our input and output “If this’’ then “that’’ respectively. The front page on the web is illustrated in Figure 4.12. This project uses water contamination detection as its trigger, which is written on the webhook as input and is defined as the event name of the event. The sensors detect contamination; therefore, it triggers an IFTTT applet that sends a notification to the user’s smartphone through ThingSpeak, a platform that collects data from the sensors. This arrangement enabled easy monitoring of water quality by users with alerts for any issues that arise. Setting up ThingSpeak for collecting data from water quality sensors is the first step in creating this system. These sensors measure temperature, turbidity, or pH, among other things. They send their records to ThingSpeak, where it holds and analyzes them. Subsequently, you will design an IFTTT applet based on the information within ThingSpeak data. You chose ThingSpeak in this portion of the IFTTT platform.

One can specify the trigger, for example, a particular degree of contamination picked up by the sensors. You select the action, such as sending a notification to your smartphone, for the "then that" section. In this manner, anytime the sensors pick up any pollution. The IFTTT applet is triggered by ThingSpeak and provides you an alert. The configuration of the ThingHTTP request and the IFTTT applet implemented in this smart, inexpensive water quality monitoring and control system are shown in Figure 4.12. The image illustrates how the parts work together to make sure users are informed about the state of the water quality in an accurate way. This system is a great way to monitor and maintain the quality of the water since it is inexpensive and efficient.

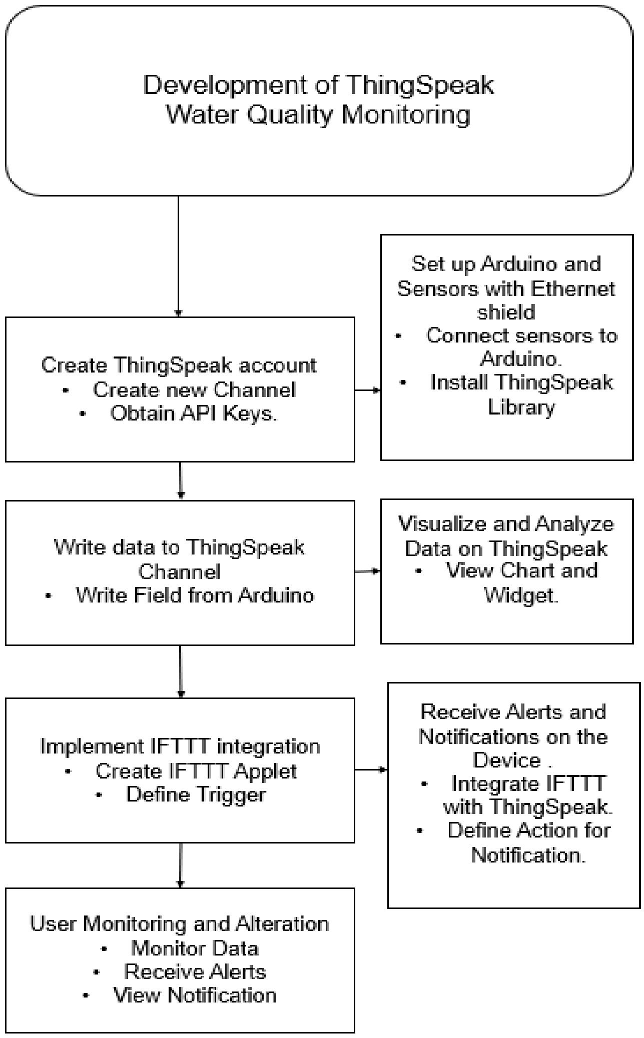


Figure 4.12: **Integration of IFTTT and ThingSpeak**

The integration of ThingSpeak and IFTTT is depicted in Figure 4.13. First, one must make sure that their system can be connected online by establishing a network connection. The next step involves configuring ThingSpeak to receive sensor data and store it as uploaded information. Afterward, one must configure ThingSpeak's webhook so that based on certain triggers that are pre-specified, this information will be sent to IFTTT from where it can be acted upon through a recipe setup. On the IFTTT side, then, you will also need a recipe that receives this webhook and takes an action, such as sending out notifications. If there is any attention needed, the user would receive a message sent directly to their Android phone from here onwards. The flow chart also displays how mobile applications interface with these systems, making it possible for users' access and view directly what was recorded within the ThingSpeak app on their cell phone devices. At this point, everything would seem fine since notifications have been dispatched successfully and everything runs smoothly.

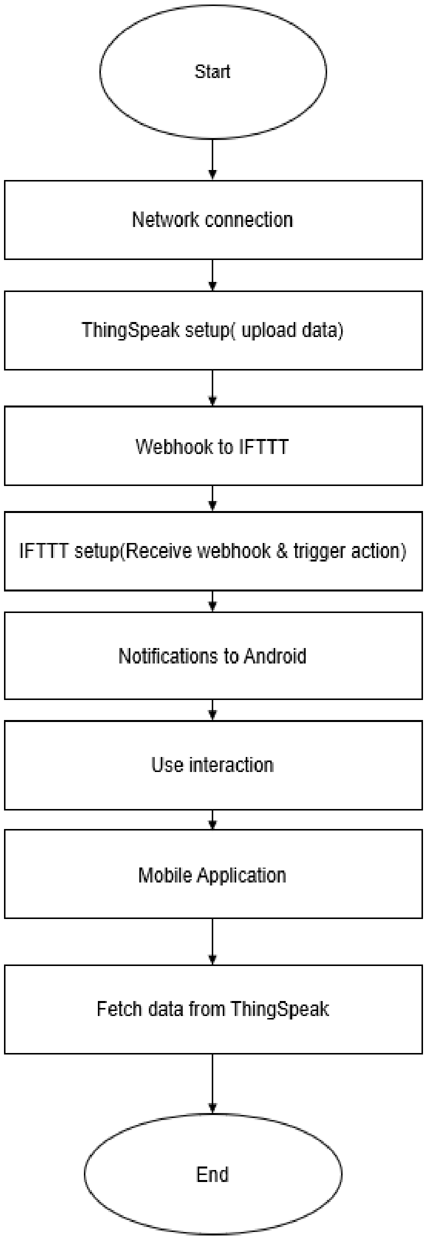


Figure 4.13: **Flowchart of IFTTT integrated with ThingSpeak**

4.6 Data Terminal (PC) for Smart monitoring and control system

To create a cheap smart system for monitoring and controlling the quality of water, we will use a laptop computer that is illustrated in Figure 4.14 as the main monitoring and control unit for water quality information. Throughout the study, this data terminal is referred to as the PC or laptop. It serves as a medium for accessing and managing software tools necessary to configure and operate sensor nodes within the system. The terminal facilitates interaction with several software programs:

- Microcontroller Unit: Programming of the MCU within sensor nodes.

- Graphical User Interface: The data terminal runs GUI, proving users can interact with the system and monitor in real-time.
- API server: The data terminal hosts the API server, enabling data communication between the sensor nodes and the cloud-based ThingSpeak.
- App: The data terminal supports the application software, which integrates with ThingSpeak and IFTTT to send notifications to the user's Android phone.

The data terminal's main purposes are to run the GUI for putting the system's decision-making algorithms into practice and to troubleshoot the microcontroller in the sensor nodes. The GUI is crucial for controlling system settings, keeping an eye on sensor data, and making sure the water quality monitoring system is operating correctly. Conditional statement methods are used in the IDE's programming to process data and make decisions based on sensor readings. The technical specifications for this laptop are:

Product: ThinkBook 15 G2 ITL
Product type: Notebook
Form factor: Clamshell
Processor family: 11th Generation Intel Core i7
Processor model: i7-1165G7
Processor Frequency: 2.80 / 4.70 GHz (12 MB Cache, 4 Cores)
Display diagonal: 39.6 cm (15.6")
HD type: FHD
Display resolution: 1920 x 1080 pixels
Display: IPS, 250nits, Anti-glare, 45% NTSC
Internal memory: 16 GB 3200 Mhz (8 GB Soldered + 8 GB SO-DIMM)
Internal memory type: DDR4-SDRAM
Total storage capacity: 512 GB M.2 2242 PCIe 3.0x4 NVMe
Storage media: SSD
On-board graphics card model: Intel Iris Xe Graphics
Operating system installed: Windows 11 Pro
Product colour: Mineral Grey
Weight: 1.70 kg

Figure 4.14: Data Terminal (Laptop)

4.7 User friendly interface

In this project, monitoring software for water quality can be utilized on both Android and iOS smartphones. The data from Arduino sensors that measure pH, turbidity, TDS, EC, flow rate, and temperature are displayed in real-time through a connection with ThingSpeak. Users who log in have personalized data such as current metrics and status (very good, contaminated, or average). Navigation is easy with a menu that offers extensive graphs and statistics for every parameter involved. Notifications sent through IFTTT enable users to respond quickly to drastic changes in water quality. The automation of tasks like turning on the water treatment pump when there are sensor problems using Arduino IDE is one of its uses. This setup is our goal towards developing an affordable framework for effective control of water quality. It highlights our research on low-cost water quality monitoring. ThingSpeak manages the information here via specific sensors, Arduino, and Ethernet shields. Timely alerts are critical when it comes to maintaining the standards set for water, and hence this integration with IFTTT. This study employed an IoT-based strategy to offer sustainable solutions for managing water. Figure 4.15 shows the set-up of integration of the app on ThingSpeak with information that was taken from IFTTT to ensure the app is integrated on both sides of ThingSpeak and IFTTT.

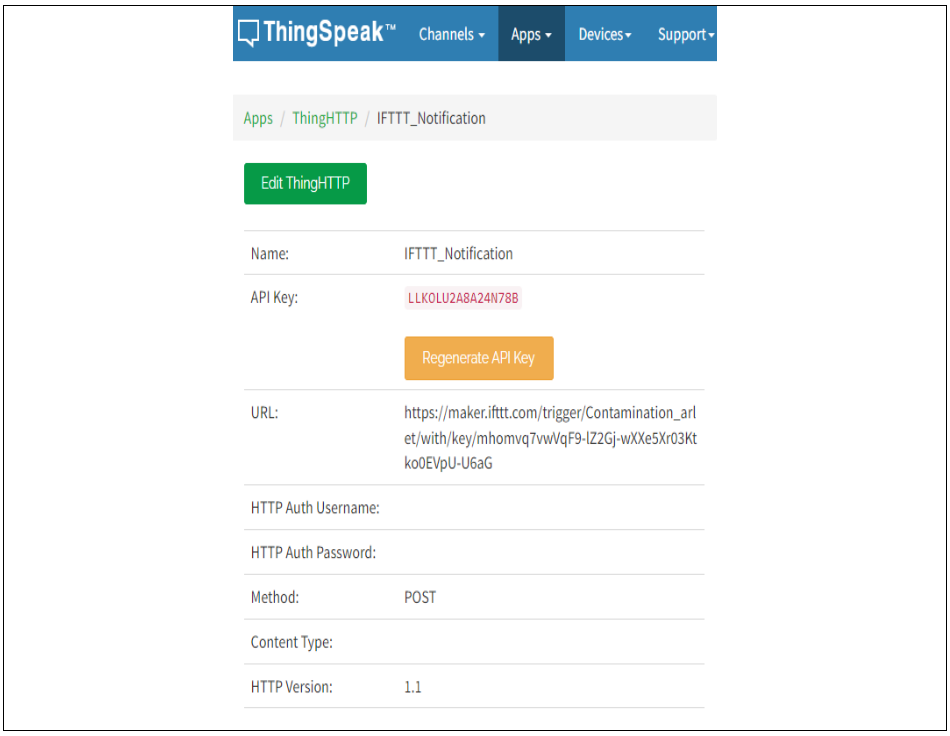


Figure 4.15: ThingSpeak App integrated with information from IFTTT

Figure 4.16 illustrates a completed app setup on the IFTTT, and it is connected so wherever there's contamination in water, the alert will be set on the mobile.

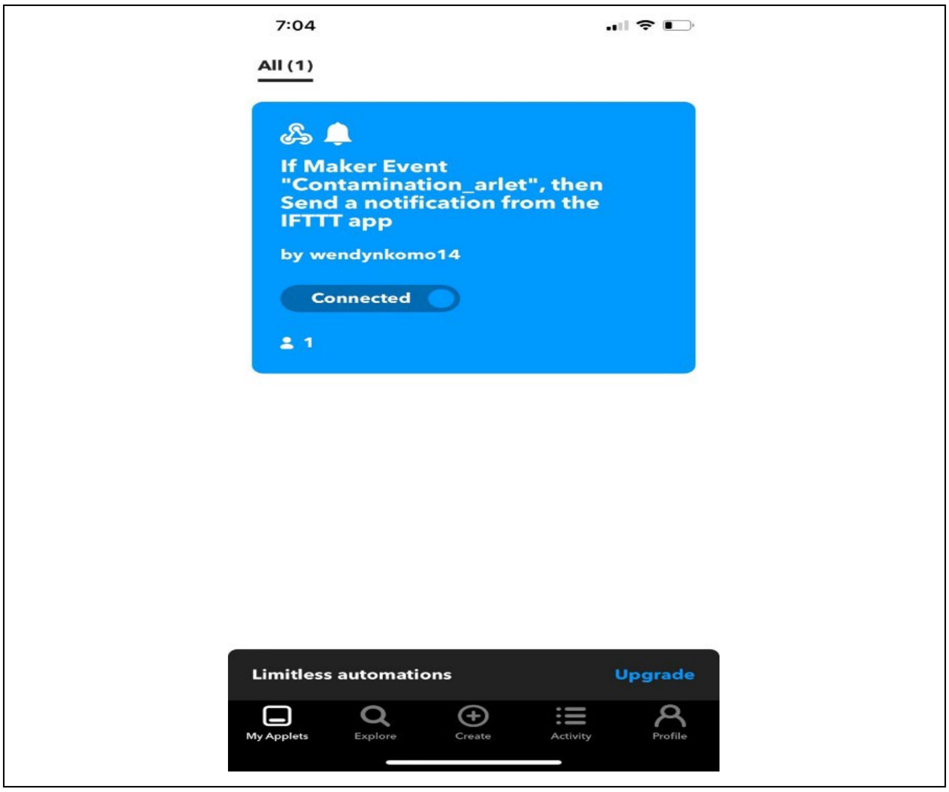


Figure 4.16: Complete set up of IFTTT

4.8 Final Project

The final part of this project used the AutoCAD diagram from Appendix W for design purposes. The researcher carefully designed a waterproof box to protect all electronic components from water damage or destruction. By using precise measurements, the researcher for this study ensures that the box can only accommodate the Arduino, Ethernet shield, batteries, relays, and sensor boards. The box dimensions are as follows: Length 108.3mm, width 166.62mm, height 22.05mm.

Base area calculations

The area of the base of the box are as follows:

$$Area = Length * width \tag{4.1}$$

$$Area = 108.3mm * 166.62mm$$

$$Area = 18{,}043.65mm^2$$

Volume calculations:

The total is of the box is

$$Volume = Length * Width * Height \tag{4.2}$$

$$Volume = 108.3mm * 166.62mm * 22.05mm = 398{,}027.4\ mm^3$$

Probes of water sensors are fixed on top of the enclosure, enabling them to directly measure the level of water. The probes are made of materials meant for underwater use; hence, they are not prone to any influence from water, while other electronics inside the box stay safe from it. Careful sealing of the enclosure was intended, especially at points where non-waterproof parts meet with those that do; thus, having waterproof grommets and seals in place to maintain the integrity of these places. When designing finally got done, soft foam padding was utilized around box boundaries. Such padding acts as a reinforcement against water getting into it and keeps its content free from potential damage caused by flowing water to maintain system functionality concerning monitoring parameters, and all were enclosed with plexiglasses as shown on Figure 4.17, which is a final prototype side view, and Figure 4.18 shows a top view of the prototype.

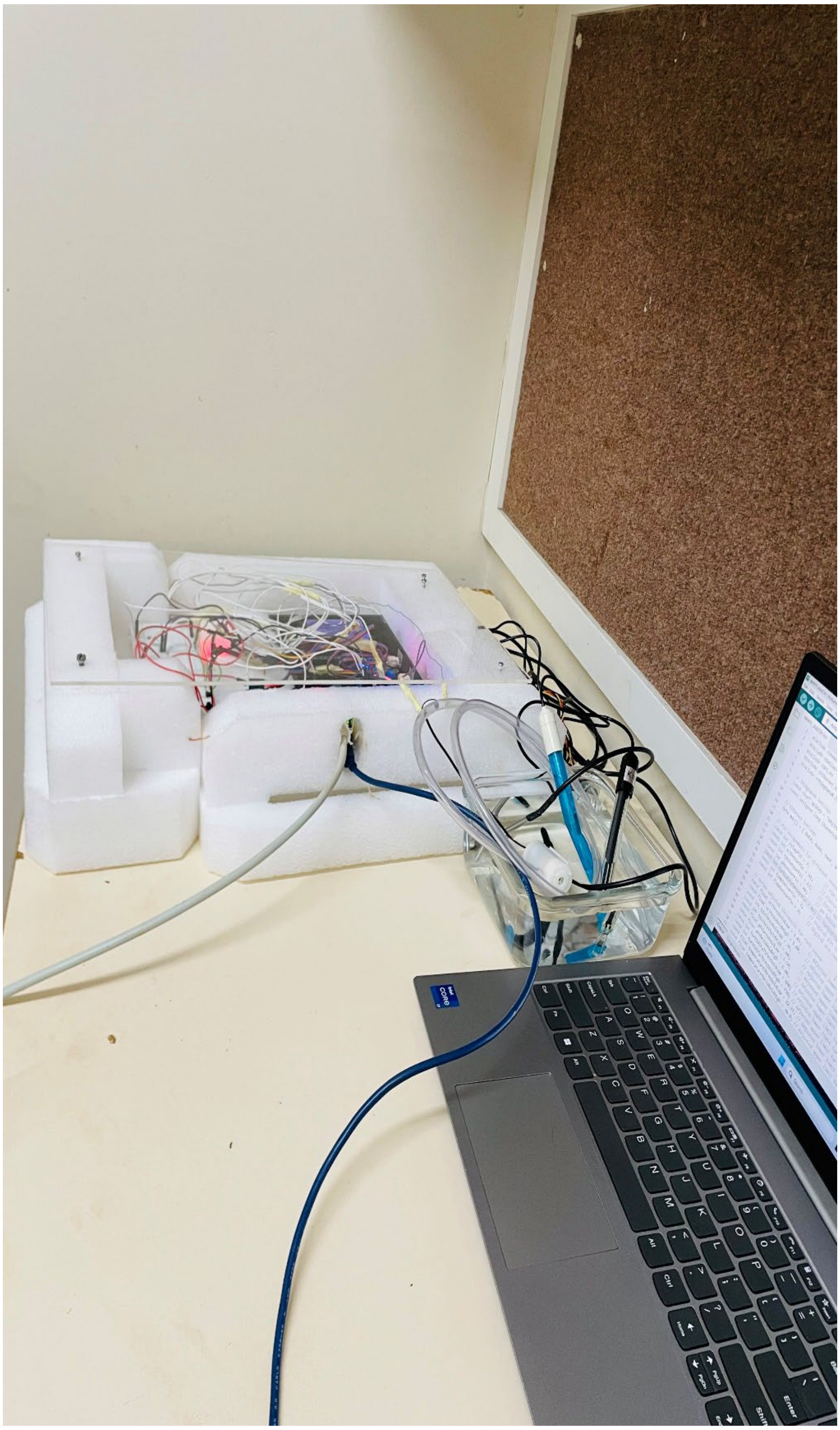


Figure 4.17: **Final prototype side view**

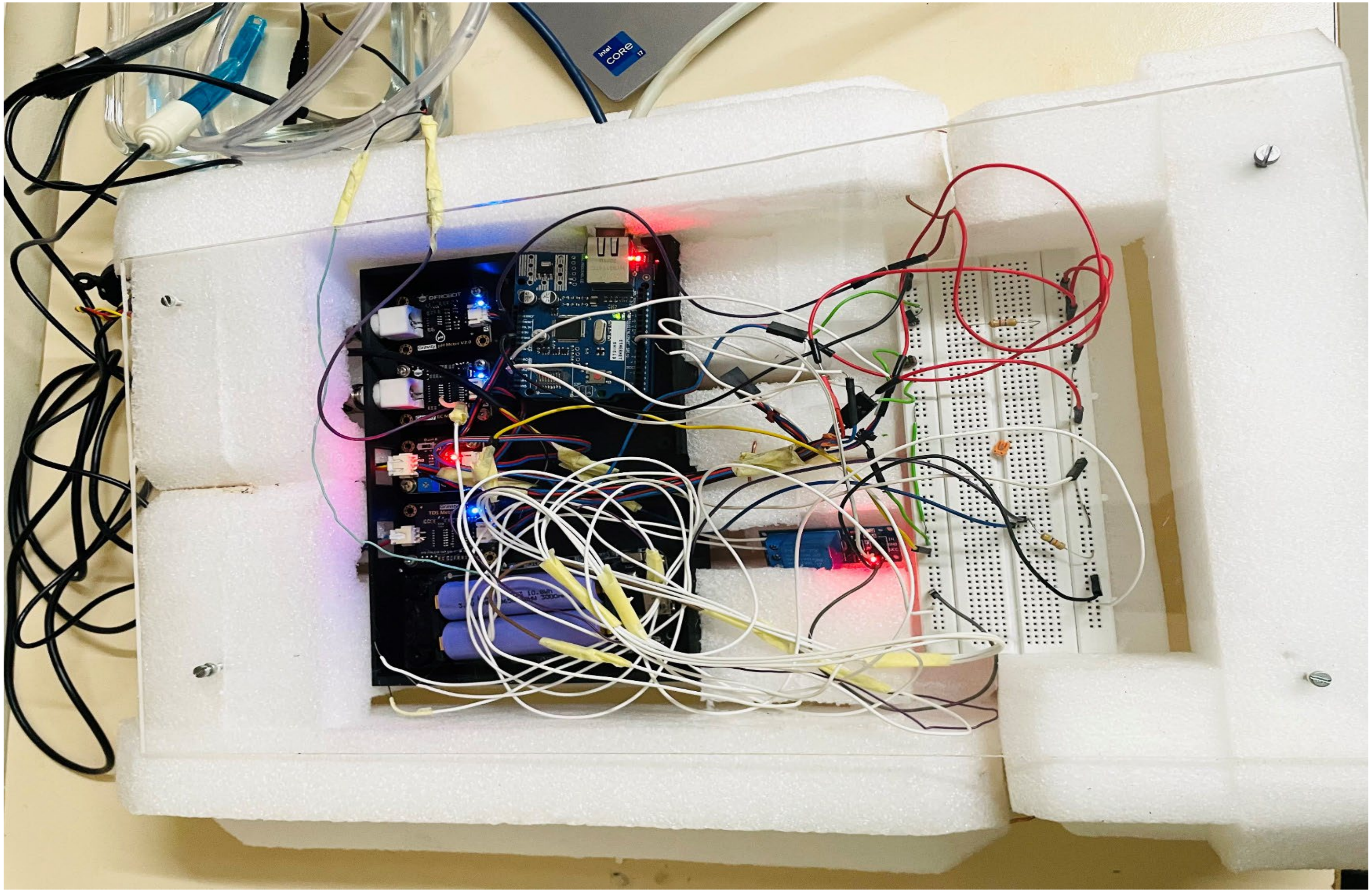


Figure 4.18: Final prototype Top View

4.9 Summary

In this chapter, everything was prepared, organized properly, and all parts of the Smart Monitor and Control System for water quality were tested. Each piece had to be expanded and studied to ensure proper functioning of the system, from calibrating the sensor to attaching it to the cloud, operating together with users, or forming a project prototype.

Part One: Sensor Integration laid out the foundations for accurate data collection using different sensors that analyze water quality parameters. The addition of such sensors into an Arduino microcontroller through successful calibration allowed building truthful real-time monitors.

Part Two: Cloud Integration broadened features such as remote access to data systems and online storage. Therefore, an individual can check on certain aspects like air or ground quality with respect to their location since it’s kept in its original state.

Part Three: Notification Alert was a proactive strategy proposed for the management of our current water quality status quo. For example, IFTTT was integrated, which helped to produce messages that were concerned with possible dangers, intending to address them with speed.

Component 4: User-Friendly Interface emphasizes communication between the system and user. An attempt was made to make a user-friendly interface that can be easily navigated by people who have no or slight knowledge concerning technology, hence being able to manipulate system processes.

Component 5: Final Design reveals a universally packed combination of all these features, allowing us to acknowledge that low-cost water quality monitoring and control systems operate.

CHAPTER FIVE
VALIDATION AND DISCUSSIONS OF RESULTS

5.1 Introduction

This section sets out to demonstrate and evaluate the findings from the water quality tracking system we have developed. The researcher scrutinized and conducted experiments on diverse water samples, utilizing the six sensors to conduct practical tests on variables such as turbidity, TDS, pH, electrical conductivity, flow rate, and temperature. Furthermore, this study will meticulously compare the data collected on Arduino with the illustrations generated on ThingSpeak and then provide commentary on the experiment's outcomes. This work displays the data directly from the Arduino monitor for each sensor, then presents it in graphs on ThingSpeak. The result is an understanding of its real-time performance and precision in measuring different aspects related to water quality. This chapter evaluates each sensor's effectiveness, data confidence, and system operation implications. The aim is to ensure that the system not only functions as intended but also prepares for real-world conditions where water quality monitoring and management occur.

5.2 EC Sensor: Analysis test

In Chapter 4, Figure 4.3, this study calibrated the EC sensor using the known solutions, 12.88 mS/cm and 1413 S/cm. The sensor produced the accurate readings with the solutions, which confirmed proper calibration for the EC sensor. Figure 5.2 displays the results of the tap water test. The researcher conducted the tests on different days in this study to observe any changes in the results and found that they aligned well with the EC ranges of tap water. This study measured the saline solution using 1 liter of distilled water and 1 tablespoon of salt, as depicted in Figure 5.1. Figure 5.2 displays the results, which match the expected saline solution, as depicted in Figure 5.3. During the testing, the temperature varied between 19 and 23 degrees. Measurements reflected these temperature variations, as shown in figures 5.2 and 5.3. Both tap water and saline solution, despite the fluctuation, were consistent with the expected outcome. Figures 5.4 and 5.5 illustrate these variations, while Appendices I, J, K, and L contain the code.

5.2.1 EC Sensor: Practical test

To practical test the analog EC sensor, solutions with known electrical conductivity values were used, including water tap and saline solution. The necessary materials included 12.88 mS/cm and 1413 µS/cm for calibration and as control solutions. Tap water to observe the sensor's response to a typical water sample and a saline solution prepared by dissolving 1 tablespoon salt (NaCl) in 1 liter of distilled water. The testing began by setting up the sensor according to the manufacturer's instructions, and the code was programmed and uploaded so that it could read and print the EC and temperature values on the Arduino IDE monitor. Each time the sensor was used first, the calibration was done using known solutions (12.88 mS/cm, 1413 µS/cm) to ensure accuracy reading across different conductivity levels.

The tests were conducted on different days to track any variations in the results. For tap water, the sensor was rinsed with distilled water before each measurement to avoid contamination. The saline solution was tested similarly, but with the sensor rinsed between different solutions. The results generally matched the EC ranges: Tap water showed EC values between 1.48 and 1.51 mS/cm at the temperature range of 16.62 °C and 19.40 °C. The saline solution was 12.13 and 12.14 mS/cm at the temperature of 15.75°C to 21.75°C. On a different testing, fresh water showed EC values of 1.36 and 1.40 mS/cm at the temperature range of 15.88°C to 21.40°C, and the saline showed 12.11 and 12.14 mS/cm at the temperature range of 16.94°C to 22.00°C.

On the figure, it shows as expected, and accuracy results on the ThingSpeak visualization graph and on the figure below showed an initial high value due to the sensor not being cleaned before immersion. After cleaning, the readings returned to expected values. This testing confirmed the sensor's consistency and reliability, validating its accuracy in measuring various water samples. The results were displayed on the Arduino IDE and sent to ThingSpeak via Ethernet shield to show the corresponding graphs, providing a clear picture of the sensor's performance, and the code is shown on the appendix from I to L.

5.2.2 Different solutions

During the sensor calibration and testing, different solutions were used, as depicted in Figure 5.1. Tap water served as a typical water sample to observe the sensor response. Furthermore, a saline solution was made by dissolving 1 tablespoon of salt (NaCl) in 1 liter of distilled water. Additionally, two buffer solutions with conductivity values of 1413 µS/cm and 12.88 mS/cm, respectively, were utilized for daily calibration of the sensors and are also represented in the illustration.

Figure 5.1: **Water, Saline and buffer solutions for EC sensor**

5.2.3 Temperature and EC indicators and Results on IDE monitor and ThingSpeak day one of testing

According to Figure 5.2, the EC sensor and temperature readings of tap water for the initial day of testing have been presented. The data shows that the measurements made with these sensors tend to exhibit some fluctuations characterized by simultaneous changes in the EC sensor readings as well as temperature readings. Such variabilities may be attributed to differing conditions within a room, which can, in turn, affect water's conductivity and hence the sensors themselves; this phenomenon is illustrated below.

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')

Water type: Freshwater (Low EC)
Temperature: 16.62 °C
EC: 0.00 mS/cm
Water type: Freshwater (Low EC)
Temperature: 16.69 °C
EC: 1.51 mS/cm
Water type: Tap Water or Bottled Water
Temperature: 16.62 °C
EC: 1.48 mS/cm
Water type: Tap Water or Bottled Water
```

Figure 5.2: Results of Tap water or bottled water on the IDE monitor

To indicate the saline water readings and the EC sensor, the figure is as follows, as the results were captured from the Arduino IDE monitor. The results show that there was some degree of change in the sensor measurements as both the EC sensor and temperature varied together with time. This may have been caused by variations in room conditions, which could have changed water conductivity, hence impacting sensor performance. This effect is illustrated in figure 5.3 below.

```
Water type: Brackish or Saline Water
Temperature: 16.94 °C
EC: 12.14 mS/cm
Water type: Brackish or Saline Water
Temperature: 17.00 °C
EC: 12.13 mS/cm
Water type: Brackish or Saline Water
```

Figure 5.3: Results of the Saline water and Temperature

Figure 5.4 displays the graph captured on ThingSpeak, showing the EC sensor reading from the first day of testing. The results on ThingSpeak aligned with the results observed or captured on the Arduino IDE monitor, as shown in Figures 5.2 and 5.3 above. Which confirms the accuracy of the data. The readings were uploaded to ThingSpeak every 15 seconds, as outlined in the code that is in Appendices I to L. This alignment between the monitor and ThingSpeak demonstrates the reliability of data transmission and storage.

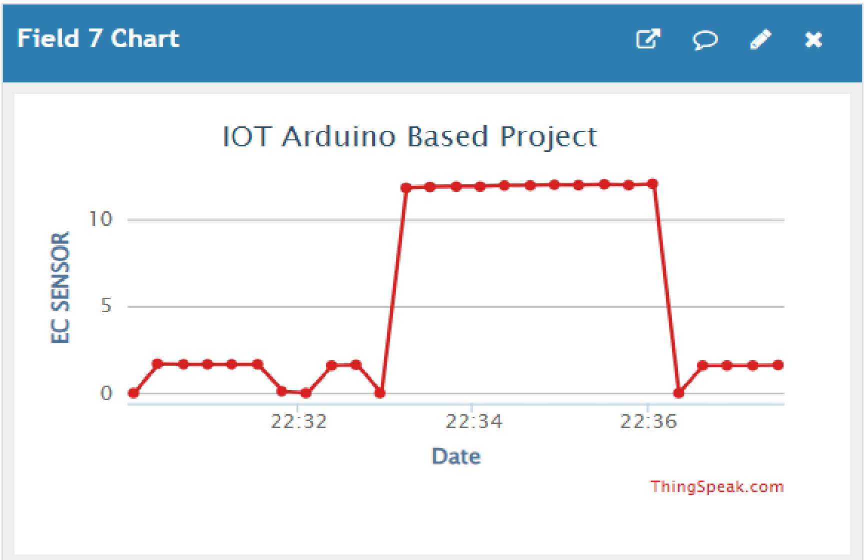


Figure 5.4: Graph Visualisation on ThingSpeak for tap water and Saline solution

On the first day of testing, the temperature readings were not captured on ThingSpeak due to the problem with my temperature connection, and the results did not want to set on ThingSpeak. It was then decided that on the following day, instead of focusing on the EC sensor readings, the attention would be shifted to capturing the temperature data to observe how it correlates with the EC values. The temperature readings continued to fluctuate, like the variations observed on the previous day.

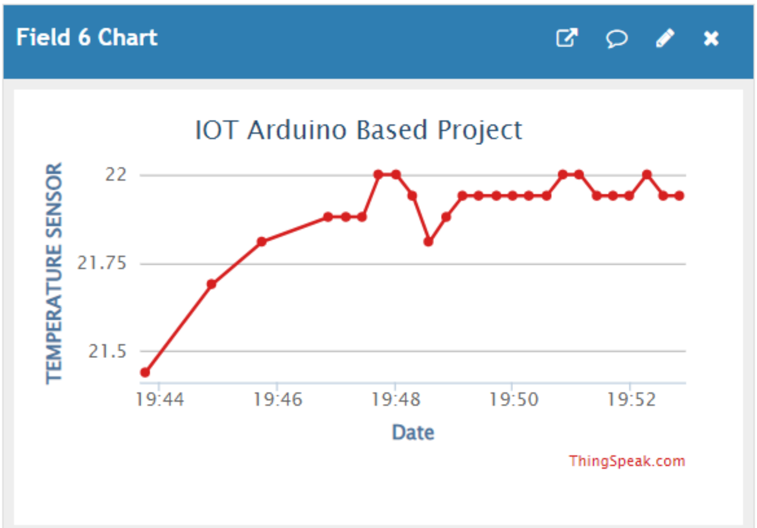


Figure 5.5: Temperature readings captured on ThingSpeak

5.2.4 EC and Temperature Results on Arduino IDE monitor and ThingSpeak on the second day of testing

On the second day of testing, the results were slightly different from previous tests due to the change of temperature or a room temperature, as shown in Figure 5.6, which illustrates the results of the saline water that kept changing.

```
Serial Monitor ×
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')
Water type: Freshwater (Low EC)
Temperature: 16.94 °C
EC: 12.11 mS/cm
Water type: Brackish or Saline Water
Temperature: 16.94 °C
EC: 12.14 mS/cm
Water type: Brackish or Saline Water
Temperature: 17.00 °C
```

Figure 5.6: **Results of the saline solution captured on Arduino IDE**

Compared to the previous day, the results for tap water showed only minor differences. The small changes that were observed were mostly due to fluctuations in room temperature, as shown in Figure 5.7.

```
EC: 1.40 ms/cm
Water type: Freshwater (Low EC)
Temperature: 15.88 °C
EC: 1.36 ms/cm
Water type: Freshwater (Low EC)
Temperature: 15.88 °C
EC: 1.40 ms/cm
Water type: Freshwater (Low EC)
Temperature: 15.94 °C
EC: 1.36 ms/cm
```

Figure 5.7: **Results of a tap water**

Figure 5.8 displays the graph from ThingSpeak showing the EC sensor readings from the first day of testing. The results on ThingSpeak were consistent with what was seen on the Arduino IDE monitor, confirming that the data was accurate, but the sensor started at a huge value due to the sensor's dirty state, and after cleaning, the results were accurate. The readings were uploaded to ThingSpeak every 15 seconds, as detailed in the code in the appendices I to L.

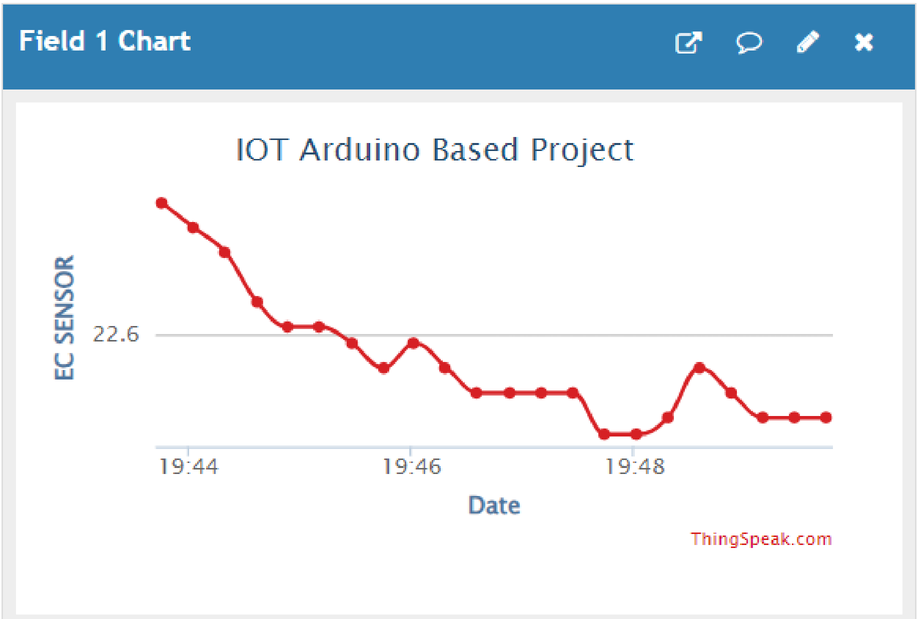


Figure 5.8: Results of Saline and Tap water on ThingSpeak

The results on the ThingSpeak graph in Figure 5.9 were good and accurate, and the temperature kept changing due to the room temperature.

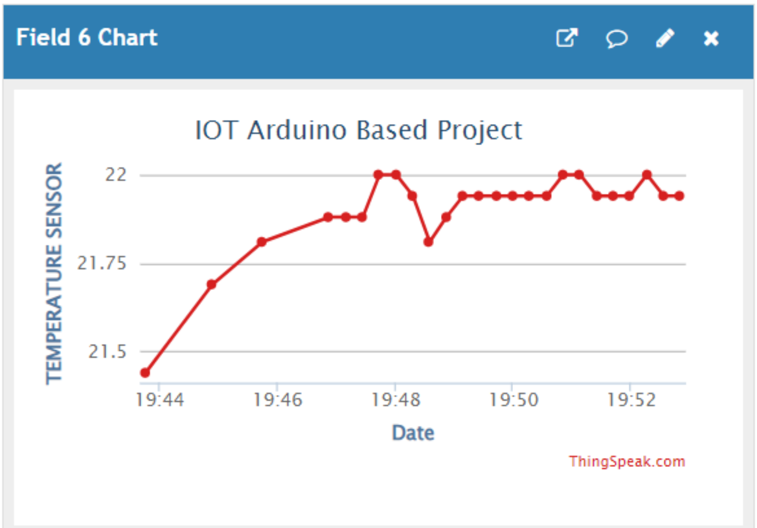


Figure 5.9: Results of the Temperature

5.3 TDS sensor's analysis and testing functionality

During the process of testing the analog TDS sensor, the code was uploaded to Arduino to ensure that the reading was accurate. The sensor was calibrated using a known value, which is typical tap water. The process involved reading analog values from the sensor, converting these to voltage values, and then, after the final reading of ppm's. During the experiment of this study, two solutions were used, which are tap water and 0.3 grams of NaCl mixed inside 1 liter of still water. By using clean tap water, 290–300 ppm was obtained, and for the second solution with NaCl and still water, 964.29–1000 ppm was obtained, which is the highest concentration for this TDS sensor according to the manufacturer. And the measurement was taken on different days to see the changes in TDS values as shown in Figures 5.11 and 5.12.

The sensor's documentation supports these findings by indicating that it is suitable for measuring TDS in domestic water, hydroponics, and other water quality application testing. The TDS value reflects the cleanliness of water, with higher values indicating more dissolved solids and it is less clean. The sensor's performance aligns with the manufacturer's specification, ensuring its accuracy and reliability for water quality. The results for its performance were shown on the IDE monitor, then sent to ThingSpeak for monitoring the sensor's various responses when immersed in these two different solutions, as shown in 5.10 and the code in Appendices G and H.

5.3.1 TDS Solutions

Two solutions were tested: tap water and a saline solution prepared by mixing 0.3 grams of NaCl into 1 liter of still water. These solutions were used to measure TDS, as shown in Figure 5.10.

Figure 5.10: **Tap water and 0.3 grams of Table salt in distilled water**

5.3.2 TDS Level indicator captured on IDE monitor

Figure 5.12 depicts the image acquired from the Arduino IDE, demonstrating TDS values as compared to their respective voltages. From this data, it can be concluded that the water tested is free from impurities and appropriate for human consumption. The test sample was taken from a tap water supply, whose TDS values are subject to variation depending on sources and local conditions. Moreover, bubble interference in the liquid may result in unsteady readings, thus affecting accuracy levels. These contrasts indicate variety in TDS as well as emphasize successful adaptive testing and calibration on accurate water quality analysis.

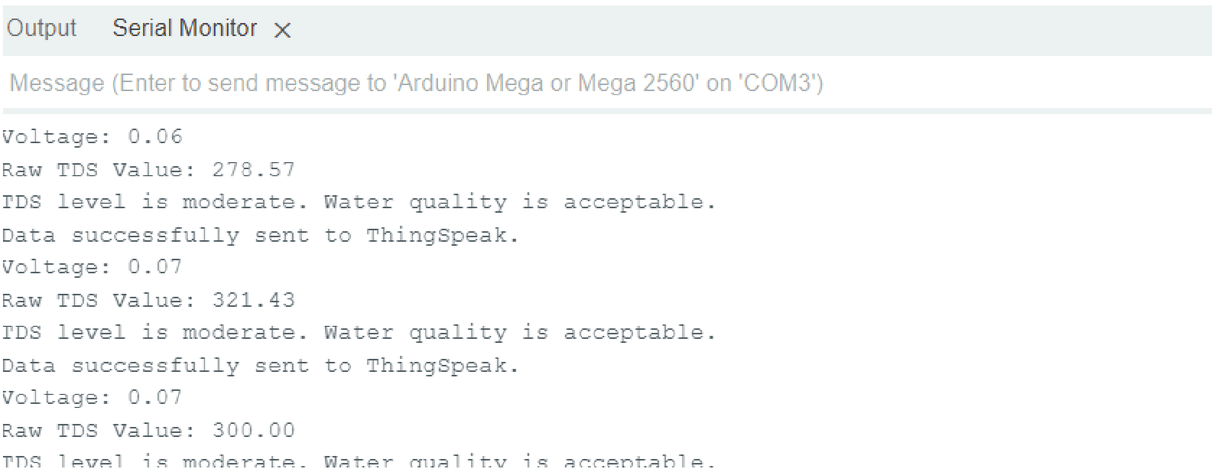


Figure 5.11: **Results of TDS using Tap water**

Figure 5.13 shows an Arduino IDE monitor image of TDS values in a solution obtained by dissolving 0.3 g of common salt in distilled water. This water was found unfit for drinking due to high TDS levels. Since these high TDS values indicate unfit drinking water, the need for careful calibration and accurate testing is emphasized to interpret the TDS values towards water quality. However, variations in readings may result from several factors, including different concentrations, thus demanding the adoption of a consistent measurement method.

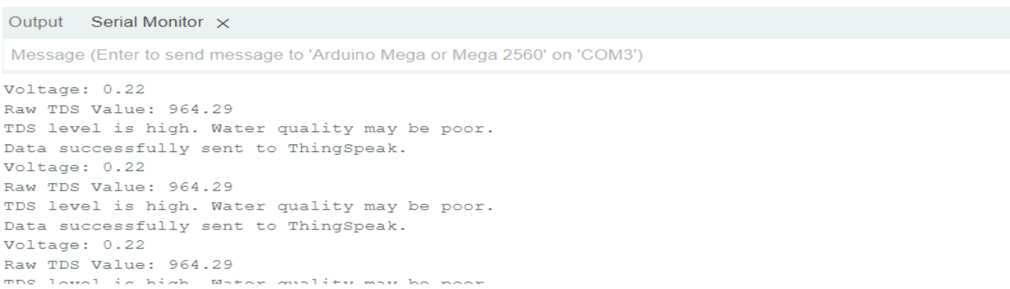


Figure 5.12: **Results using 0.3 grams of Table salt in distilled water solution**

Figure 5.13 displays the results for both tap water and unacceptable water, illustrating the range of TDS values from acceptable to unacceptable levels. The data is presented in a graph on ThingSpeak, clearly indicating the differences in water quality between the two solutions.

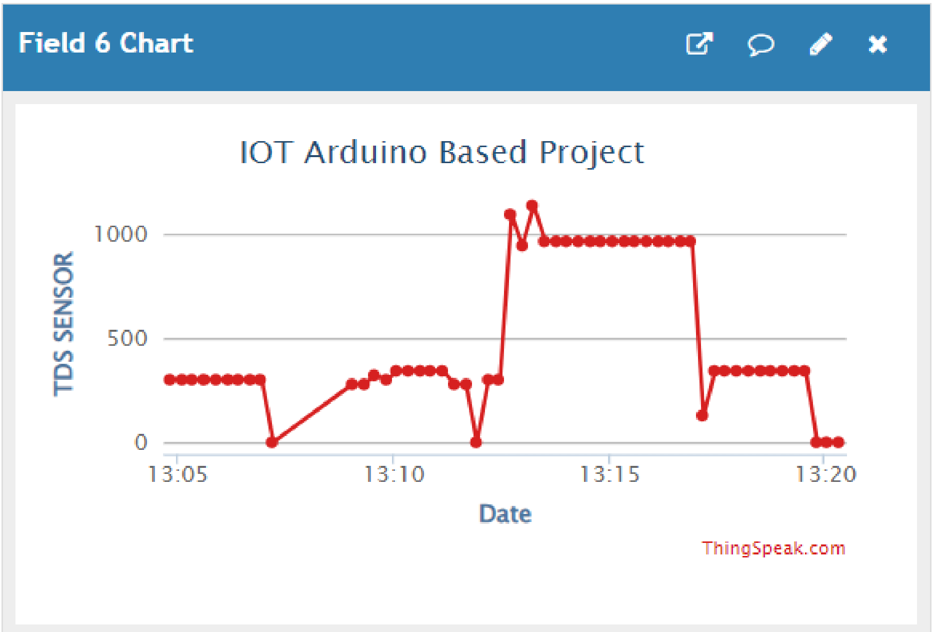


Figure 5.13: **Results of a Tap and unacceptable water**

In Figure 5.14, the tap water was tested again on a different day. The results of this test are displayed on ThingSpeak, providing a clear representation of the water quality measurements for Tap water solution.

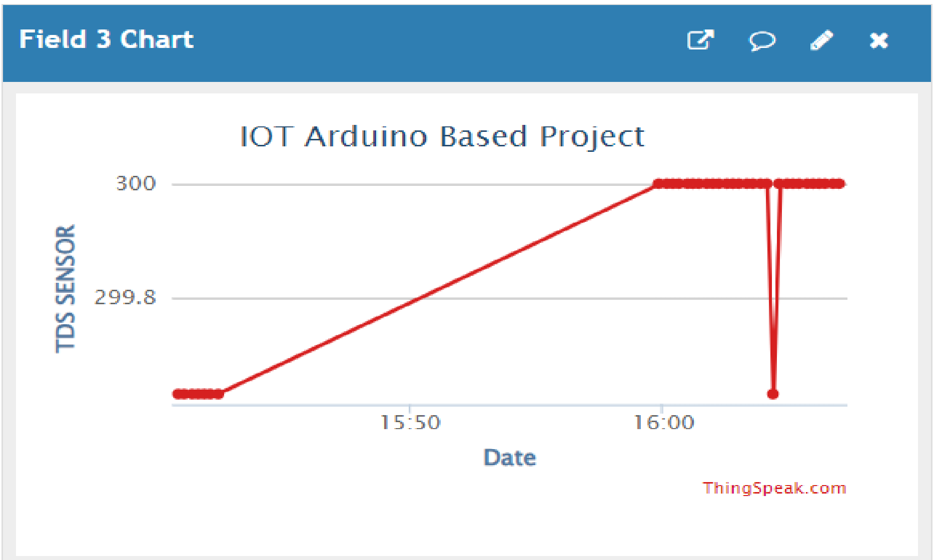


Figure 5.14: **Results of Tap water**

TDS test results collected on a different day are portrayed in figure 5.15. The TDS reading was 0 ppm dissolved solids when the sensor was not placed in the solution. On the other hand, when the sensor was immersed in water, the reading rose to 1000 ppm, which is the upper limit for the sensor. Such high levels lead to the conclusion that the quality of water is poor because of high levels of dissolved solids.

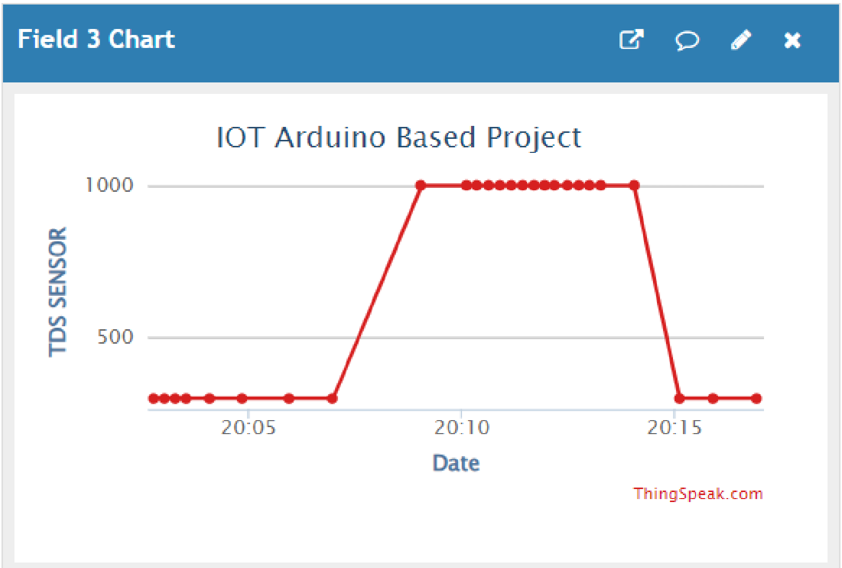


Figure 5.15: **Results of unacceptable water**

5.4 pH sensor Analysis test

In this study, an experiment was conducted to test how well a pH sensor works in our water quality monitoring setup. The goal of this study was to determine how well the sensor measures the pH levels against different known solution levels. According to SANS 241:2015, water is considered safe to drink if it falls between 6.5 and 9.5 pH levels. During the test of this experiment, the pH sensor was dipped into solutions with pH values of 3.10 (vinegar), 7 (natural spring water), and 10:15 (tap water mixed with baking soda). These solutions were chosen to represent acidic, neutral, and alkaline levels. During the calibration, the reference solution was used; it was a clean water tap to check the sensor's accuracy in measuring pH values. The results were verified on ThingSpeak. It showed the sensor's accuracy to capture different pH levels and during different times of the day and the code on Appendices D and E.

5.4.1 pH sensor Practical test

During the calibration process described in Chapter 4, Figure 4.6, the pH was first submerged in tap water to allow it to settle, ensuring accurate readings. Once settled, the sensor was tested into different pH solutions, each placed in different glasses as shown in Figure 5.16. These solutions were chosen to represent a range of pH levels. Acidic, neutral, and alkaline.

The sensor was immersed in each solution at various times and on different days to observe the changes in pH levels and corresponding voltages, as shown in figures 5.17 and 5.18. After each test, the sensor was cleaned throughout before testing another one to prevent contamination. The test was conducted in the following sequence, as shown in Figure 5.16:

- pH Level 3.10
- pH Level 7.00
- pH Level 10.15

The data collected during the tests was shown on the Arduino monitor and sent to ThingSpeak as the graph visualization on different figures. This setup allowed for real-time monitoring of pH levels. To determine where the water reached the drinkable or undrinkable level threshold. By continuously tracking these pH levels to make sure that the water has been certified for drinking.

Figure 5.16: Vinegar, Natural Spring water and Tap water mix with baking soda solutions

5.4.2 pH Level indicators and ThingSpeak graphs

Figure 5.17 shows results captured on an Arduino IDE monitor for vinegar, and the varies because the sensor changes when put on another solution or cleaned.

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')

```
Voltage: 1.15 V, pH Value: 3.10
Acidic
Data successfully sent to ThingSpeak.
Voltage: 1.15 V, pH Value: 3.10
Acidic
Data successfully sent to ThingSpeak.
Voltage: 1.45 V, pH Value: 6.30
Neutral
Data successfully sent to ThingSpeak.
Voltage: 1.45 V, pH Value: 6.30
```

Figure 5.17: Acidic results

Figure 5.18 shows neutral and alkaline results that were captured on the IDE monitor.

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')

```
Data successfully sent to ThingSpeak.
Voltage: 1.45 V, pH Value: 6.30
Neutral
Data successfully sent to ThingSpeak.
Voltage: 2.15 V, pH Value: 10.15
Alkaline
Data successfully sent to ThingSpeak.
Voltage: 2.15 V, pH Value: 10.15
Alkaline
Data successfully sent to ThingSpeak.
```

Figure 5.18: Neutral and Alkaline

The graph is shown below Figure 5.19 as it was captured on ThingSpeak results varies between neutral results.

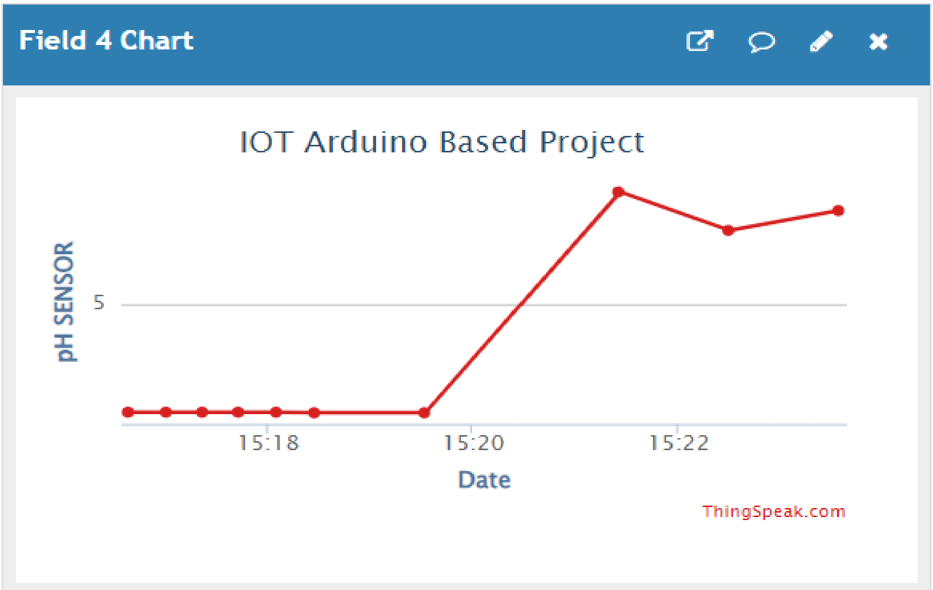


Figure 5.19: **Results of a drinkable water on the pH**

Results on Figure 5.20 on a different day show results of acidic, neutral, and alkaline on the ThingSpeak. That means the sensor emerged to different solutions.

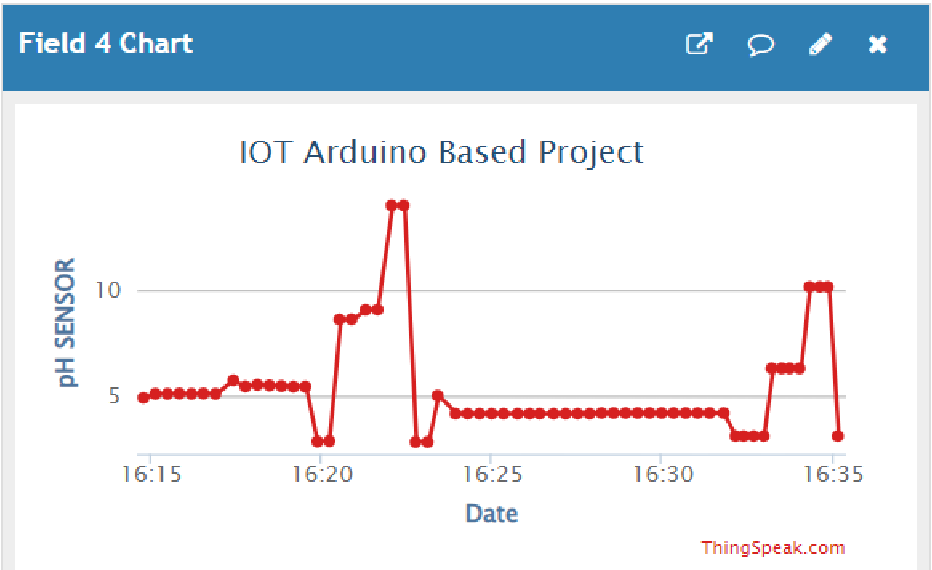


Figure 5.20: **Results of Acidic, Neutral and Alkaline**

Results on Figure 5.21 are for neutral and alkaline, which sensor was immersed in clean and dirty water.

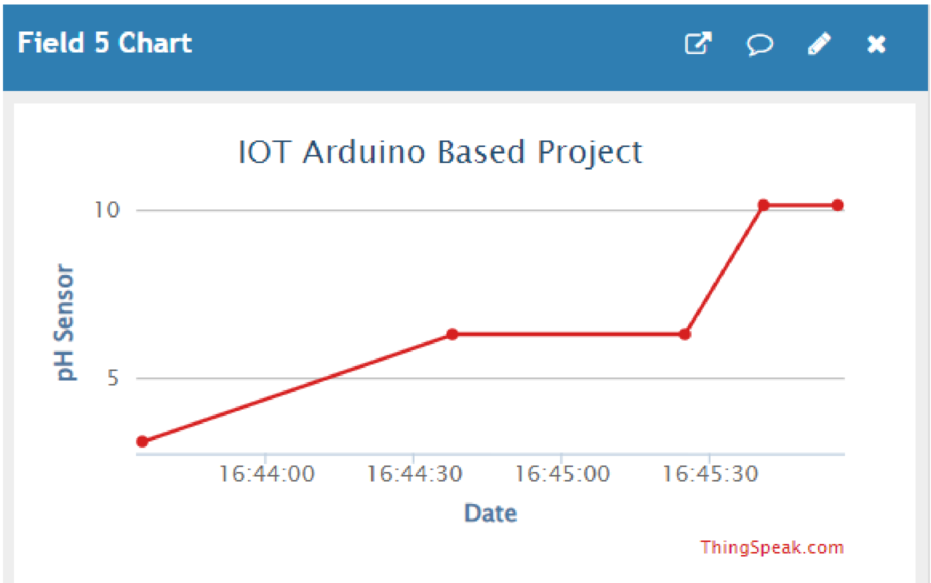


Figure 5.21: Results of Neutral and alkaline

5.5 Turbidity sensor Analysis and practical test

The turbidity sensor SKU SEN0189 was tested using an Arduino microcontroller connected through an analog-to-digital converter (ADC). The sensor's VCC to Arduino 5V, GND to GND, and its analog pin to A0 of Arduino. After uploading and testing the code sensor, the sensor provided consistent readings, confirming it was functioning correctly. The voltage from cloudier to clear water ranges between 3.5 and 1.14 volts, as shown in Figures 5.22 and 5.23. And the turbidity from clear to cloudier ranges from 0 to 20 NTU, covering everything from relatively clear to higher turbidity. Then on the second day, I measured dirty water throughout, as shown on the ThingSpeak graphs. The results were then displayed on ThingSpeak over different days, which showed that the sensor can measure broad ranges of turbidity effectively. The variation showed that this sensor has the capability to detect low and higher turbidity levels in samples, and the code is shown in Appendix M.

Solutions used for a turbidity are shown in Figure 5.22

Figure 5.22: Solutions for testing Turbidity

As illustrated in Figure 5.23, the solutions used were for cloud water, and the results obtained are shown in the below figure.

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')
Channel update successful.
Turbidity: 20.00 NTU    Voltage: 3.08 V Water Type: Cloud Water
Channel update successful.
Turbidity: 20.00 NTU    Voltage: 2.80 V Water Type: Cloud Water
Channel update successful.
Turbidity: 19.44 NTU    Voltage: 3.22 V Water Type: Cloud Water
Channel update successful.
Turbidity: 20.00 NTU    Voltage: 3.28 V Water Type: Cloud Water
Channel update successful.
Turbidity: 20.00 NTU    Voltage: 2.73 V Water Type: Cloud Water
```

Figure 5.23: Dirty water

The results obtained on Figure 5.24 are for the clean turbidity values and voltages.

```
Turbidity: 2.90 NTU     Voltage: 1.50 V Water Type: Clean Water
Channel update successful.
Turbidity: 2.40 NTU     Voltage: 1.14 V Water Type: Clean Water
Channel update successful.
Turbidity: 4.20 NTU     Voltage: 2.42 V Water Type: Clean Water
```

Figure 5.24: Results of a clean turbidity values

The graph captured from ThingSpeak is illustrated in Figure 5.25, and it is the results of a voltage to the measurement of turbidity (dirty water).

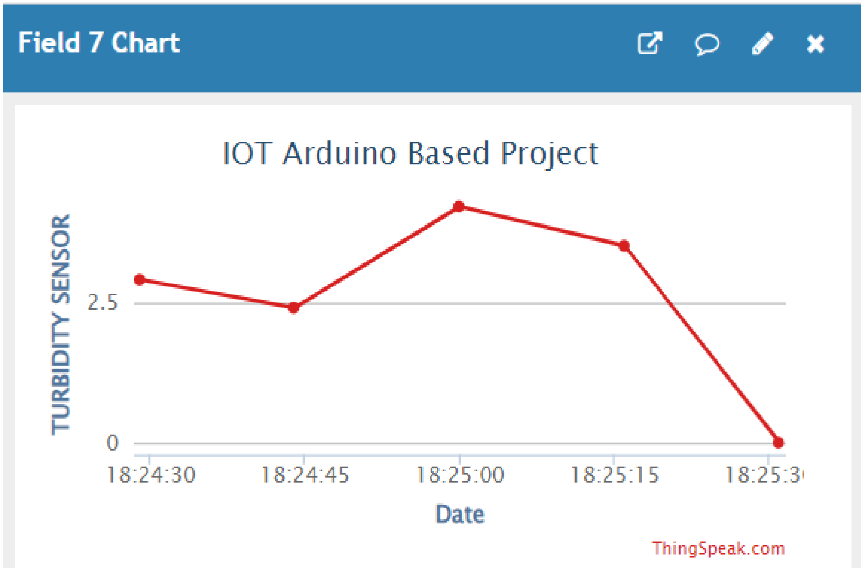


Figure 5.25: **Results of a NTU for clean water using Turbidity sensor**

The graph captured from ThingSpeak is illustrated in Figure 5.26, and it is the results of a voltage to the measurement of turbidity (dirty water).

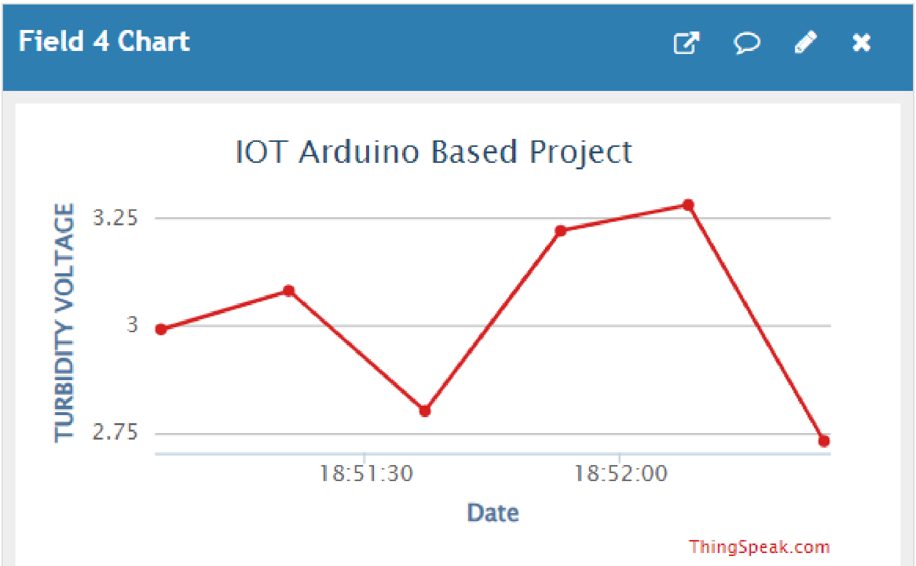


Figure 5.26: **Results Voltage of a Dirty water**

The graph captured from ThingSpeak is illustrated in Figure 5.27, and it is the results of a NTU measurement of turbidity (dirty water).

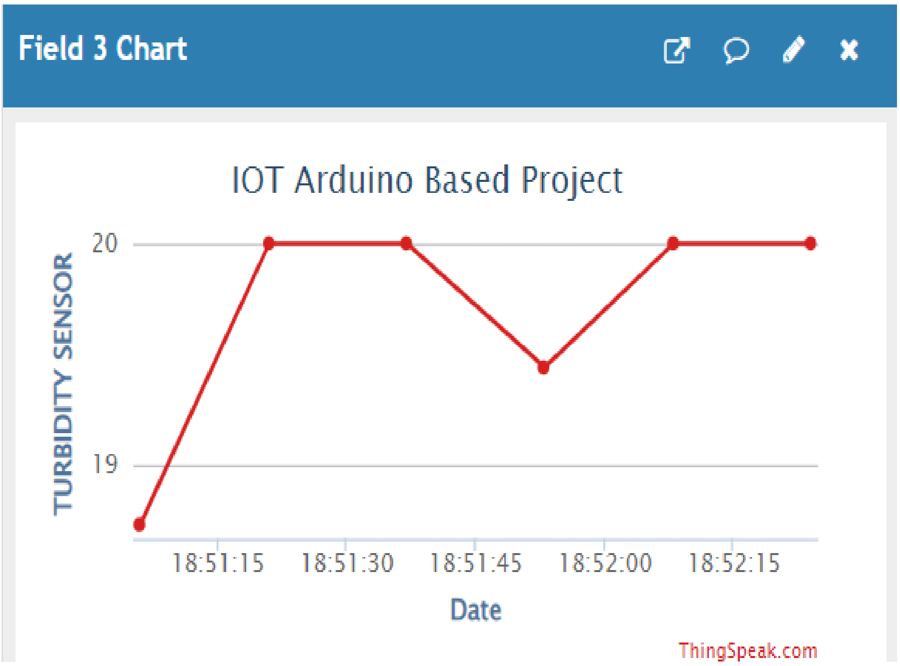


Figure 5.27: **Results of NTU Dirty water using Turbidity sensor**

The graph captured from ThingSpeak is illustrated in Figure 5.28, and it is the results of voltage measurements of turbidity (clean water).

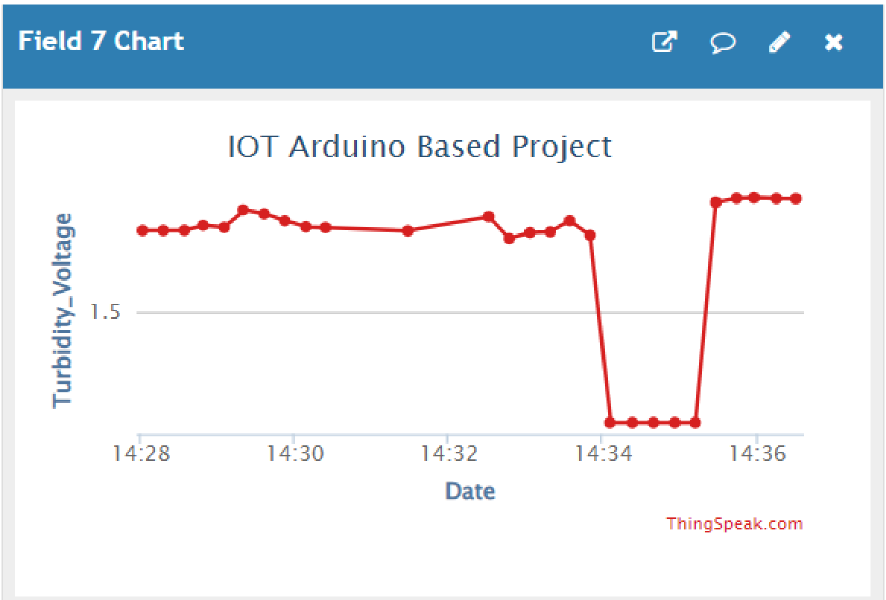


Figure 5.28: **Results of a Voltage for a Clean water using a Turbidity sensor**

The graph captured from ThingSpeak is illustrated in Figure 5.29, and it is the results of voltage measurements of turbidity (clean water).

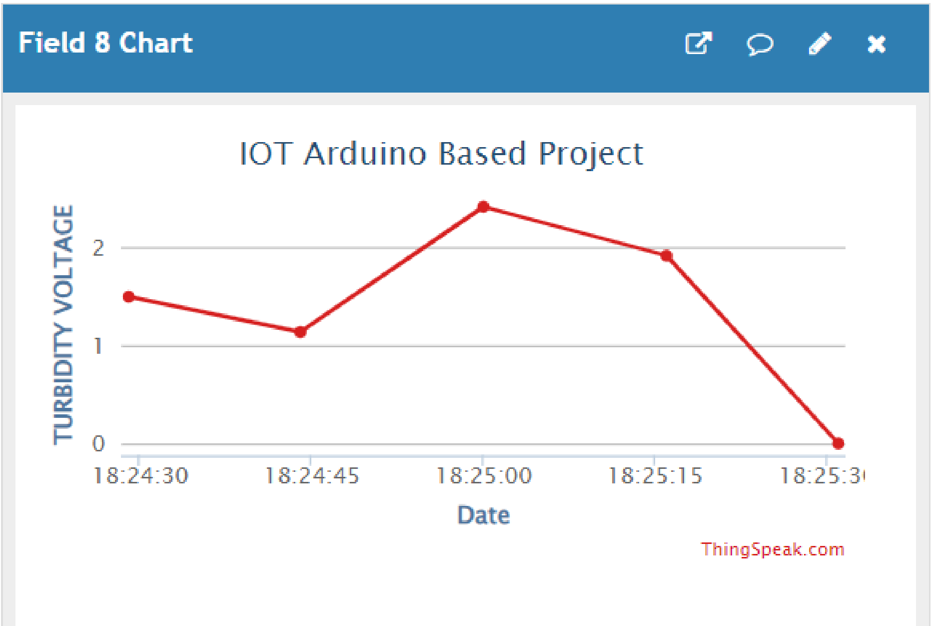


Figure 5.29: Results of a Voltage for a Clean water using a Turbidity sensor

5.6 Water sensor: analysis

The purpose of the analysis test was to assess the accuracy and performance of the YF-S201 water flow sensor using the provided Arduino code. The main goal was to make sure that the sensor functioned as expected and produced accurate flow rate measurements. To start, the sensor was connected to the Arduino according to the instructions; its output pin was connected to digital pin 2 on the Arduino, providing correct power and ground connections; the code was uploaded to the Arduino, which was made to measure the flow rate based on the pulses generated by the sensor. The flow rate was obtained and shown on the serial monitor in liters per minute (L/m). Water was allowed to pass through the sensor during the test while the code was executing. The measurements of the flow rate were monitored and recorded. The code is shown in Appendix F.

5.6.1 YF-S201 water flow sensor: Practical test

The purpose of the practical test was to evaluate the functionality of the YF-S201 water flow sensor in an actual environment. Confirming that the sensor performed as expected and functioned well under normal circumstances required the completion of this test. The sensor was mounted in the intended location within the water flow system and linked to the Arduino via the correct power and ground connections. The flow rate, which was determined in liters per minute (L/m) and displayed on the serial monitor as indicated on Figure 5.30, was read and displayed using the supplied Arduino code. Water was run through the sensor during the practical test, and the flow rate values were recorded. To assess the sensor's performance in a variety of circumstances, it was tested with diverse flow conditions. The results proved the sensor's steady and dependable operation under everyday conditions. The measurements of the flow rate were consistent and in line with expectations. In its intended application, the sensor proved to be a good fit for measuring water flow.

5.6.2 The Process of Testing Flow water sensor

In a serial monitor, flow rate measured in liters per minute (L/m) is displayed like this, as seen in figure 5.30. This figure shows how a flow rate was measured using 1 liter of solution that was precisely processed through a sensor’s flow tube. The outcomes from this regulated on-time measurement then stand for flow rate real-time represents on the serial monitor.

Figure 5.30: **Water flow rate testing.**

The flow rate, measured in Liters per minute (L/m), was displayed on the serial monitor as shown in Figure 5.31.

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')
1.60 L/m
1.60 L/m
1.60 L/m
1.60 L/m
1.76 L/m
1.76 L/m
1.76 L/m
1.76 L/m
1.76 L/m
1.76 L/m
1.92 L/m
```

Figure 5.31: Results of Flow rate on the IDE monitor

5.7 The results analysis for overall system performance

Figure 5.32 shows the results from the overall system. The Electrical Conductivity (EC) readings were stable, ranging from 0.82 to 0.83 mS/cm, indicating that the water's conductivity was consistent. Temperature measurements varied slightly between 22.12°C and 22.19°C, reflecting steady thermal conditions during the test. pH values were also stable, between 5.61 and 5.62, suggesting the water's acidity remained constant. The turbidity sensor gave a consistent reading of 741 NTU, showing a steady level of suspended particles. Flow rate measurements fluctuated between 0.3 and 4 L/min, indicating varying water flow through the system. Since all these measurements stayed within their acceptable ranges, the system's pump was activated to ensure that the water quality remained within the desired parameters, and the code is illustrated on the Appendix from P to V.

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')
EC: 0.83 TDS: 21 pH: 5.61 Turbidity: 741 Temperature: 22.19 Flow Rate: 2.70
Pump is ON
EC: 0.83 TDS: 21 pH: 5.62 Turbidity: 741 Temperature: 22.12 Flow Rate: 2.90
Pump is ON
EC: 0.83 TDS: 21 pH: 5.62 Turbidity: 741 Temperature: 22.12 Flow Rate: 4.00
Pump is ON
EC: 0.82 TDS: 21 pH: 5.61 Turbidity: 741 Temperature: 22.12 Flow Rate: 1.20
Pump is ON
EC: 0.82 TDS: 21 pH: 5.62 Turbidity: 741 Temperature: 22.19 Flow Rate: 0.30
Pump is ON
```

Figure 5.32: Results of all Six and pump on Arduino IDE monitor

On a different testing day, the researcher for this study tested the sensors and communicated their variations to ThingSpeak, as shown in Figures 5.33 and 5.34. Even though the sensors had been appropriately calibrated, and their values were within the anticipated limits, their results differed from the previous ones. There may be changes in Electrical Conductivity (EC) sensor readings due to a varied temperature or composition of water. Temperature fluctuations or changes in the water can also influence the temperature readings. Alterations in water chemistry can also cause changes in pH sensor readings, while the number of suspended particles in the water largely determines turbidity readings. The TDS displayed varying values, indicating the water's impurity, yet the sensor stayed unchanged on ThingSpeak. Creating a new field may be necessary due to its outdatedness. For flow rate measurement, it may depend on alterations in water flow or pressure. The most important aspect here is whether the DC pump is functioning properly despite these differences. The tests revealed that the DC pump performed as expected, indicating that, despite fluctuations in sensor readings, it remained functional. The appendix provides the code from P to V.

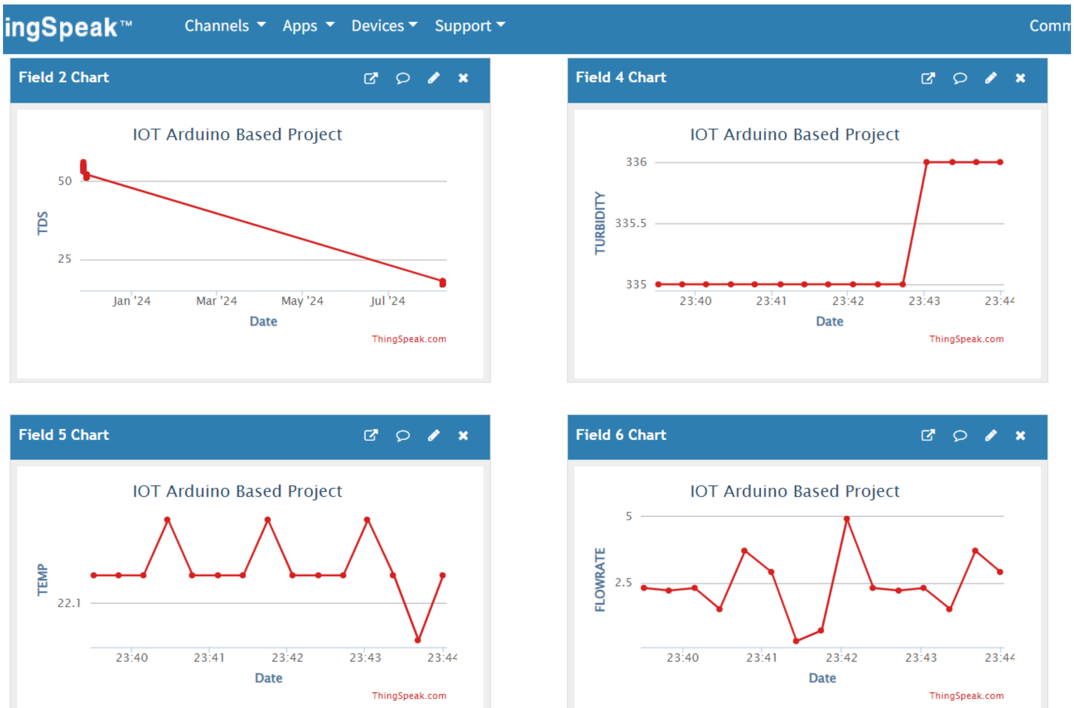


Figure 5.33: Results of TDS, Turbidity, Temperature and Flowrate on ThingSpeak

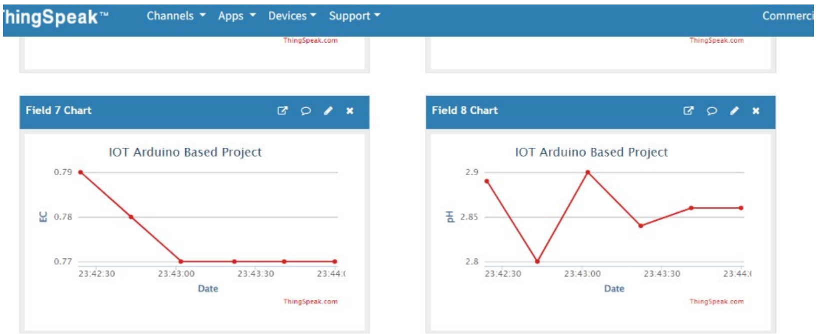


Figure 5.34: **Results of all sensors testing on ThingSpeak**

5.8 Estimated Cost Breakdown for Low-Cost Smart Water Quality Monitoring and Control System

1. Core Components

These are the main sensors, controllers, and connectivity components.

- **Arduino Mega**: R385.00
- **Ethernet Shield**: Free from Department of Electrical, Electronic and Computer Engineering.
- **Water Quality Sensors**:
  - pH Sensor: R1734.49
  - EC Sensor: R1600
  - TDS Sensor: R248.00
  - Turbidity Sensor: R238.33
  - Water Flow Sensor: R190
  - Temperature Sensor: R119.00
- **Total for Core Components**: R3882.49 – R4605.00

2. Control and Power Components

Control elements for water treatment automation and power supply.

- **DC Pump**: R21.74
- **Relay**: R95
- **Lithium Batteries**: R90 – R275
- **Battery Charger**: R60-R120

- **Total for Control and Power Components**: R171-R300

3. Communication and Software Integration

These resources manage data handling and notification systems.

- **ThingSpeak API**: Free
- **IFTTT**: Free
- **Total for Communication and Software Integration**: R0

4. All the additional Materials

These materials were provided at no cost by the Department of Electrical, Electronic, and Computer Engineering.

- **Jumper Wires, Breadboard, Connectors**: Free
- **Enclosure Box for Housing Components**: designed free
- **soft foam padding: free**
- **Total for Additional Materials**: R0

5.9 Comparison of this solution with Existing Water Quality Monitoring Solutions

This research designed a cheap sanitation water quality monitoring facility that was capable of measuring pH, turbidity, TDS, temperature, flow rate, and electrical conductivity among other parameters using cheap sensors and components. As Chapter 3 explained, the design emphasized offering credible monitoring that can be accessible for communities with limited resources. One of the key advantages was that such hardware and data handling tools were also made available at no cost together with items such as jumper wires and a box, therefore lowering expenses greatly. On the other hand, the current systems invariably depended on costly laboratory and specialized equipment, which made them impractical for even basic monitoring in rural areas. In this regard, our solution has been able to provide a low-cost option for water quality monitoring while ensuring that rural communities can still get clean water without compromising on the quality of the data collected. This study has a total estimated cost of R4,053.49. In comparison, off-the-shelf water quality monitoring systems often range from R7000 to R20,000, depending on their capabilities and features.

5.8 Summary

This section evaluates the performance of turbidity, total dissolved solids (TDS), pH, electrical conductivity, flow rate, and temperature sensors applied for real-time water quality monitoring. The system was found to be reliable and accurate in real life, and comparing readings from the Arduino monitor with Thingspeak graphical representations confirmed that the system is reliable and accurate when applied in actual conditions. Each sensor had stable and accurate measurements, indicating effectiveness in continuous water supervision. Indeed, it is an exceptional instrument that ensures the provision of pure and safe drinking water in a variety of situations, with the DC pump and relay operating flawlessly.

CHAPTER SIX
DELIVERABLES, CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This chapter wraps up the research on the creation of a low-cost intelligent water quality monitoring system. Creating a system that could monitor vital indicators like temperature, water flow, pH level, turbidity, electrical conductivity, and total dissolved solids was the primary objective of the project. ThingSpeak, a cloud-based platform for real-time monitoring and analysis, received the thoroughly processed data from these sensors. The study also explored the possibility of automating the system's response to water quality issues through IFTTT, demonstrating the effective use of low-cost technology in water resource management. Throughout this project, this study has focused on enhancing accuracy through proper calibration and appropriate data processing methods, ensuring that sensors record accurate readings. This chapter begins by summarizing the key findings and contributions made by this research work. It then provides recommendations for future research and practical applications, offering insights into potential directions for future work based on the achievements in this project.

6.2 Aims and objectives

This study starts with the assertion that clean and safe water is essential and highlights how smart monitoring systems can fill the gaps left by traditional methods. The main goal of this research is to develop a low-cost, smart system for monitoring and controlling water quality.

A list of the goals for achieving the aim is stated in Chapter One is repeated here for ease of reference:

- Literature Review: Review 20 and more studies on Arduino water quality monitoring systems in eight weeks to find useful ideas.
- Water Quality Parameters: To learn about key water quality measurements (temperature, TDS, pH, turbidity, EC, and water flow) and their importance in eight weeks.
- Monitoring and Control Model: developing a model using sensors and Arduino, Ethernet, Thingspeak, and FTTT for measuring water quality within twenty-eight weeks.

- Automated Control System: Design a system to automatically treat water based on sensor readings, aiming for reliable accuracy in pump operation within eight weeks.

All above listed aims and objected satisfactorily achieved.

6.3 Dissertation deliverables

The work completed to meet the goals and objectives of the research study is summarized in this section.

6.3.1 Chapter 2: Literature Review

The literature review started by exploring the various issues surrounding water access and quality. The data sources used took into consideration low-income community challenges beneath prolonged water scarcity, with diverse evolving water demands. Some of the most significant determinants of water quality were also evaluated. We then reviewed different methods of testing water quality, comparing traditional approaches with the current systems in use for monitoring and identifying problems (Smart). This part offered online water quality assessment basics, a few recent technological changes that have brought about the well-established practice (within its application scope), and how water quality management has been revolutionized by the Internet of Things (IoT). We brought together the features of IoT systems along with their integration into such purification processes and instantaneous examination (real-time analysis) of bathing water cleaning conditions. In conclusion, we summarized how our study could supplement other sectors about purity management that are nowadays directed at advanced technologies in this field even for rural areas.

6.3.2 Chapter 3: Research Methodology

This chapter concentrates on developing the methodology for a low-cost smart water quality monitoring system that is compatible with an Arduino microcontroller. It contains details about the design and planning stages, explaining how key features like water sensors were integrated with the Ethernet shield in the Arduino microcontroller. The overall architecture of the whole system is outlined in this section while explaining why those components were chosen, therefore enabling effective water management. To enhance clarity, a flowchart representing both the design process, and the sequence of activities involved has also been added here to coil everything together clearly. This chapter provided foundational insights into our design decisions and planning that informed the construction of such systems.

6.3.3 Chapter 4: Engineering system design

In this chapter, we focused on setting up, configuring, and testing each component of the automated smart water quality monitoring and control system. We detailed how each piece fit into the overall system, from sensor integration to the final design.
The chapter was organized into six main components:

- Sensor Integration: We described how each sensor was connected and configured to gather data on water quality.
- Cloud Integration: We outlined the process of linking the system to the cloud for data storage and accessibility.
- Notification Alerts: We explained how the system sent notifications to users about water quality issues.
- User-Friendly Interface: We covered how the system's interface was designed to be intuitive for users.
- Final Design: We presented the final integrated system, showcasing how all components worked together.

Each section included explanations and test demonstrations that illustrated how the system met its design objectives. The system codes were included in the Appendix, and sensor connections were detailed in Chapter 3. This chapter provided a comprehensive look at how the system was assembled and tested to ensure it functioned as intended.

6.3.4 CHAPTER FIVE: Validation and Results analysis

Chapter 5 focused on validating and analyzing the performance of our smart water quality monitoring system. We started by thoroughly testing the water sensors to ensure they delivered accurate and reliable readings. This involved a series of analysis tests to assess how well the sensors worked under various conditions. Next, we put the sensors through practical tests to see how they performed in real-world situations. This helped us understand how they would operate outside of a lab environment and provided valuable insights into their real-life performance. We also tested the sensors with different water solutions to confirm their accuracy across a range of water qualities. This step was important to make sure the sensors could handle various conditions and still provide reliable measurements. The chapter also covered a comparison of the results shown on our IDE monitoring system with those displayed on ThingSpeak. We reviewed the graphs and data visualizations on ThingSpeak, checking to see if the results matched what we observed on the IDE. This comparison was crucial for verifying that both systems were in sync. In the end, the validation process confirmed that our system was effectively measuring and displaying water quality parameters. The consistent results between the IDE and ThingSpeak visualizations showed that the system was performing as expected.

6.4 Academic and industrial application

The research presented here is valuable in terms of industry and academia. It provides an affordable approach to monitoring water quality for industries. Utility companies could use the system constructed as a more efficient way of dealing with water quality than what has been done conventionally.

In academic circles, this research forms a basis upon which future intelligent water quality systems can be built. Additionally, it informs new research designs while helping students learn how to design and implement similar systems. The research enhances knowledge in areas related to water management as well as proposing future innovations in the domain.

6.5 Future work

Future researchers can focus on improving the hardware of the smart water quality monitoring system to test its performance in real-world conditions. Additionally, researchers could explore increasing automation to better manage water quality changes. Expanding the system with additional sensors and control features might also improve its effectiveness in various water quality scenarios.

6.6 Recommendations

- Future research should focus on more complete water quality assessment that can be accomplished through extending the range of sensors so that they could include other variables such as ORP or dissolved oxygen.
- Advanced calibration options must be invested in by future development, along with automatic calibration choices to make readings from sensors more accurate and constant.
- Remote regions can benefit from improved network connectivity, such as cellular networks or LoRa, to facilitate reliable data transmission.

- Remote applications should look at other ways of optimizing battery life through the integration of power-saving components and/or solar power options, which will prolong operational time.

6.7 Summary

This section discussed the goals and outcomes of the research project using a low-cost smart water quality monitoring and control system. This section encompasses the pertinent literature, and theoretical frameworks utilized in the execution of this work. It then delves into the design and construction of the system, integrating its sensors with an Arduino unit. Additionally, it discusses the implications of the study for both academic research and practical applications worldwide. This chapter provides recommendations for future improvements and highlights steps such as referring to sources and publications during the research.

REFERENCES

Adjovu, G.E., Stephen, H., James, D. & Ahmad, S. 2023a. Measurement of Total Dissolved Solids and Total Suspended Solids in Water Systems: A Review of the Issues, Conventional, and Remote Sensing Techniques. *Remote Sensing*, 15(14): 3534.

Adjovu, G.E., Stephen, H., James, D. & Ahmad, S. 2023b. Measurement of Total Dissolved Solids and Total Suspended Solids in Water Systems: A Review of the Issues, Conventional, and Remote Sensing Techniques. *Remote Sensing*, 15(14).

Aggarwal Khushboo & Ahmad Nadeem. 2022. FLUORIDE AND CHILDREN'S HEALTH IN INDIA: DISTRICT LEVEL ANALYSIS. , 14(3): 630–647.

Alabrsh, M. 2023. *Agricultural Inputs and Commodity Price Bulleen in the Syrian Arabic Republic-Issue #15*.

Almassri, A.M., Wan Hasan, W.Z., Ahmad, S.A., Ishak, A.J., Ghazali, A.M., Talib, D.N. & Wada, C. 2015. Pressure sensor: State of the art, design, and application for robotic hand. *Journal of Sensors*, 2015.

Anon. 2024. Financing water security and mitigating investment risks. In 137–146.

Anshori, M., Sari, N., Vijai, A., Program, N., Fisika, S., Matematika, F., Ilmu, D. & Alam, P. Rancang Bangun Alat Ukur Kekeruhan Air Layak Pakai Berbasis Arduino Uno R3 Pada Sungai Martapura. *Jurnal Ilmiah Fisika FMIPA Universitas Lambung Mangkurat*, 19(3): 2541–1713. https://doi.org/.

Ashbolt, N.J. 2004. Microbial contamination of drinking water and disease outcomes in developing regions. In *Toxicology*. 229–238.

Astya, P.Nand. 2017. *IEEE International Conference on Computing, Communication and Automation (ICCCA 2017) : proceeding : on 5th-6th May, 2017*. IEEE.

B A Karomah. 2022. Different Methods of Monitoring Water Quality the Use of Water Quality Monitoring in the Field. : 1–5.

Ba, A. & McKenna, S.A. 2015. Water quality monitoring with online change-point detection methods. *Journal of Hydroinformatics*, 17(1): 7–19.

Banna, M.H., Najjaran, H., Sadiq, R., Imran, S.A., Rodriguez, M.J. & Hoorfar, M. 2014. Miniaturized water quality monitoring pH and conductivity sensors. *Sensors and Actuators, B: Chemical*, 193: 434–441.

Barros, M., Granchinho, P., Ferreira, C., Neves, P., Magalhaes, H., Santos, L., Lopes, B., Marques, J., Pinho, H., Mourato, S. & Martins, A. 2019. Study of the requirements of an autonomous system for surface water quality monitoring. *Renewable Energy and Power Quality Journal*, 17: 399–404.

Bonadonna, L., Briancesco, R. & La Rosa, G. 2019. Innovative analytical methods for monitoring microbiological and virological water quality. *Microchemical Journal*, 150.

Butuner, R. & Uzun, Y. *Arduino Microcontroller Card*.

Cong, Y., Li, X., Zhang, S., Zheng, Q., Zhang, Y. & Lv, S.W. 2023. Embedding Carbon Quantum Dots into Crystalline Polyimide Covalent Organic Frameworks to Enhance Water Oxidation for Achieving Dual-Channel Photocatalytic H2O2 Generation in a Wide pH Range. *ACS Applied Materials and Interfaces*, 15(37): 43799–43809.

Dr. Puneeth GJ, Shashank T, Vinit P, S D Muzammil & N R Karthikeya. 2024. Smart Water Pollution Management: IoT for Automatic Detection and Prevention. *International Journal of Advanced Research in Science, Communication and Technology*: 532–548.

Egieya, J.M., Parker, Y., Hofmann, V.S., Daher, B., Gorgens, J. & Goosen, N.J. 2024. Predictive simulation of the water-energy-food nexus for the City of Cape Town. *Science of the Total Environment*, 934.

Enqvist, J.P. & Ziervogel, G. 2019. Water governance and justice in Cape Town: An overview. *WIREs Water*, 6(4).

Faruq, Md.O., Emu, I.H., Haque, Md.N., Dey, Maitry, Das, N.K. & Dey, Mrinmoy. 2017. Design and implementation of cost effective water quality evaluation system. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE: 860–863.

G, A. 2020. . G, Ann. (2020) 7 Ways to Measure, Monitor, and Evaluate Water Quality. [Online] Public Lab. Available at: https://publiclab.org/notes/anngneal/12-08-2017/7-ways-tomeasure-monitor-and-evaluate-water-quality.

Geetha, S. & Gouthami, S. 2016. Internet of things enabled real time water quality monitoring system. *Smart Water*, 2(1): 1.

Grewar, T. 2019. South Africa's options for mineimpacted water re-use: A review. *Journal of the Southern African Institute of Mining and Metallurgy*, 119(3).

Hakim, W.L., Hasanah, L., Mulyanti, B. & Aminudin, A. 2019. Characterization of turbidity water sensor SEN0189 on the changes of total suspended solids in the water. *Journal of Physics: Conference Series*, 1280(2): 022064.

Handandi, Z.Z., Hidayana, E., Setiawan, E., Juniani, A.I., Nugraha, A.T. & Amelia, P. 2024. UTILIZING TOTAL DISSOLVED SOLIDS (TDS) SENSOR FOR DISSOLVED SOLIDS MEASUREMENT IN THE WATER. , 7(1): 22–30.

Heger, S. 2017. An Installer's Guide to Total Suspended Solids.

Hermawan, A., Andrian.Harahap, D.A., Daging, I.K., Dewi, P., Ridhwan, R.Z. & Qadri, M. 2023. Design of a Web-based Cold Storage Temperature Monitor with Arduino Uno for Fish Quality Maintenance: Sensor-based Methodology and Innovative Contribution. *SINTEK JURNAL: Jurnal Ilmiah Teknik Mesin*, 17(2): 161.

Jang M. Y. B., Lee. J & Lee K. 2020. Integration of GPS with remote sensing for environmental monitoring and management. *Environmental Monitoring and Assessment*, 192(2): 1–70.

Jin, D. 2022. *Application of remote sensing technology in water quality monitoring*.

J.K Mwabi, B B Mamba & M B B Momba. 2013. . Removal of waterborne bacteria from surface water and groundwater by cost-effective household water treatment systems (HWTS): A sustainable solution for improving water quality in rural communities of Africa. Water SA, 39(4). : 445–456.

K, G. & S, N. 2017. Gopavanitha, K. and Nagaraju, S., 2017, August. A low-cost system for real time water quality monitoring and controlling using IoT. In 2017 International conference on energy, communication, data analytics and soft computing (ICECDS) (pp. 3222EE. : 3227–3229.

Kirankumar, P., Keertana, G., Abhinash Sivarao, S.U., Vijaykumar, B. & Chetan Shah, S. 2021. Smart Monitoring and Water Quality Management in Aquaculture using IOT and ML. In *Proceedings - 2021 IEEE International Conference on Intelligent Systems, Smart and Green Technologies, ICISSGT 2021*. Institute of Electrical and Electronics Engineers Inc.: 32–36.

Lakshmikantha, V., Hiriyannagowda, A., Manjunath, A., Patted, A., Basavaiah, J. & Anthony, A.A. 2021. IoT based smart water quality monitoring system. *Global Transitions Proceedings*, 2(2): 181–186.

Lambrou, T.P., Anastasiou, C.C., Panayiotou, C.G. & Polycarpou, M.M. 2014a. A low-cost sensor network for real-time monitoring and contamination detection in drinking water distribution systems. *IEEE Sensors Journal*, 14(8): 2765–2772.

Lambrou, T.P., Anastasiou, C.C., Panayiotou, C.G. & Polycarpou, M.M. 2014b. A low-cost sensor network for real-time monitoring and contamination detection in drinking water distribution systems. *IEEE Sensors Journal*, 14(8): 2765–2772.

Manjakkal Libu, Lorenzelli Leandro & Willander Magnus. 2023. Toward Real-Time Water Quality Monitoring Using Wireless Sensor Networks. In Sohail Sarang, Goran M. Stojanović, & Stevan Stankovski, eds. *Toward Real-Time Water Quality Monitoring Using Wireless Sensor Networks*.

Mbana, Z. & Sinthumule, N.I. 2024a. What is there to drink? Water (in)justice in the democratic South Africa. *Frontiers in Water*, 6.

Mbana, Z. & Sinthumule, N.I. 2024b. What is there to drink? Water (in)justice in the democratic South Africa. *Frontiers in Water*, 6.

V Mema. 2010. Impact of poorly maintained wastewater sewage treatment plants-lessons from South Africa: Wastewater management. ReSource. : 60–65.

Miry, A.H. & Aramice, G.A. 2020. Water monitoring and analytic based thingspeak. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(4): 3588.

Mishra, B.K., Kumar, P., Saraswat, C., Chakraborty, S. & Gautam, A. 2021. Water security in a changing environment: Concept, challenges and solutions. *Water (Switzerland)*, 13(4).

Mokoena, A. 2023. QUESTIONING DAY ZERO: RIGHTS, PROVISION, AND WATER INEQUALITY IN SOUTH AFRICA. *Human Organization*, 82(3): 223–234.

Mukta Monira, Samia Islam, Barman Das Surajit, Reza Wasif Ahmed & Khan Saddam Hossain. 2019. IoT based Smart Water Quality Monitoring System. : 1–5.

P Singh & S Saikia. 2016. Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module. . : 1–4.

Parsons, G., Paul, L.A. & Messer, K.D. 2022. Demand for an Environmental Public Good in the Time of COVID-19: A Statewide Water Quality Referendum. *Journal of Benefit-Cost Analysis*, 13(1): 107–119.

Pasika, S. & Gandla, S.T. 2020. Smart water quality monitoring system with cost-effective using IoT. *Heliyon*, 6(7): e04096.

Pichel, N., Vivar, M. & Fuentes, M. 2019. The problem of drinking water access: A review of disinfection technologies with an emphasis on solar treatment methods. *Chemosphere*, 218: 1014–1030.

Postolache', O., Girzo', P., Pereim', M. & Ramos', H. 2002. *An IR Turbidity Sensor: Design and Application*.

Pranta, A.D., Rahaman, Md.T., Ahmed, Md.S. & Arefin Rafi, Md.S. 2023. Navigating Eutrophication in Aquatic Environments: Understanding Impacts and Unveiling Solutions for Effective Wastewater Management. *Research in Ecology*, 5(3): 11–18.

Prapti, D.R., Mohamed Shariff, A.R., Che Man, H., Ramli, N.M., Perumal, T. & Shariff, M. 2022. Internet of Things (IoT)-based aquaculture: An overview of IoT application on water quality monitoring. *Reviews in Aquaculture*, 14(2): 979–992.

Rajalashmi, K., Yugathian, N., Monisha, S. & Jeevitha, N. 2021. IoT based water quality management system. In *Materials Today: Proceedings*. Elsevier Ltd: 502–515.

Rajput Singh Krishnapal, Sikarwar Singh Dharmendra, Jain Sheetal, Mukharya Ayan, Sabre Krati & Batham Deepak. 2022. AI Based Motor/Water Pump Switching System. : 1–6.

Ray, P.P. 2016. A survey of IoT cloud platforms. *Future Computing and Informatics Journal*, 1(1–2): 35–46.

Rich, D., Andiroglu, E., Gallo, K. & Ramanathan, S. 2023. A review of water reuse applications and effluent standards in response to water scarcity. *Water Security*, 20.

Rusydi, A.F. 2018. Correlation between conductivity and total dissolved solid in various type of water: A review. In *IOP Conference Series: Earth and Environmental Science*. Institute of Physics Publishing.

Samsudin, S.I., Salim, S.I.M., Osman, K., Sulaiman, S.F. & Sabri, M.I.A. 2018. A smart monitoring of a water quality detector system. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(3): 951–958.

Sani, S.A., Ibrahim, A., Musa, A.A., Dahiru, M. & Baballe, M.A. 2023. Drawbacks of Traditional Environmental Monitoring Systems. *Computer and Information Science*, 16(3): 30.

Sapkal, R.J., Wattamwar, P., Waghmode, R. & Tamboli, U. 2019. A Review - Water Quality Monitoring System. *International Journal of Scientific Research in Science and Technology*.

Satyanarayana Murthy, N. & Ahamed, S.F. 2023. Water quality monitoring and measuring physicochemical parameters using wireless sensor networks. *African Journal of Aquatic Science*, 48(4): 366–373.

Shamsi, S.M., Abdullah, H.B. & Bakar, L. 2020. Development of Integrated EC and pH Sensor for Low-Cost Fertigation System. In *IOP Conference Series: Earth and Environmental Science*. Institute of Physics Publishing.

Singh Sumer. 2022. Water quality and security . *International journal of multidisciplina*, 2(3): 429–435.

Sisodia, N., McGuinness, K.L., Wadhawan, J.D. & Lawrence, N.S. 2022. In situ recalibration of ion selective electrodes. *Sensors and Diagnostics*, 1(1): 134–138.

Suresh Lal, B. 2018. Water for life : issues and challenges . In *Water for life*. 1949–1957.

Tamayo, A., Campos-Lendinez, Á., Muñoz, J., Crivillers, N. & Mas-Torrent, M. 2023. pH Sensor Based on Supramolecular Host-Guest Interactions and an Organic Field-Effect Transistor with a Magnetic Carbon Gate Electrode. *Chemistry of Materials*, 35(21): 9257–9263.

Tantray, J.A., Mansoor, S., Wani, R.F.C. & Nissa, N.U. 2023. pH meter. In *Basic Life Science Methods*. Elsevier: 9–10.

Thirstrup, C. & Deleebeeck, L. 2021. Review on Electrolytic Conductivity Sensors. *IEEE Transactions on Instrumentation and Measurement*, 70.

Trevathan, J., Read, W. & Schmidtke, S. 2020. Towards the development of an affordable and practical light attenuation turbidity sensor for remote near real-time aquatic monitoring. *Sensors (Switzerland)*, 20(7).

Uddin, M.G., Nash, S. & Olbert, A.I. 2021. A review of water quality index models and their use for assessing surface water quality. *Ecological Indicators*, 122.

Vanshika Ambati, Kuber Ramya B & Poojitha Nalluri. 2023. Innovations in Machine Learning and IoT for Water Management. In Abhishek Kumar, ed. *Revolutionizing Water Quality Monitoring: The Smart Tech Frontier*. 152–171.

Verlicchi, P. & Grillini, V. 2020. Surface Water and Groundwater Quality in South Africa and Mozambique—Analysis of the Most Critical Pollutants for Drinking Purposes and Challenges in Water Treatment Selection. *Water*, 12(1): 305.

Van Der Vyver, C. 2015. *Improving Awareness of the Water Crisis in South Africa*. https://www.researchgate.net/publication/281436406.

Wang, Y., Rajib, S.M.S.M., Collins, C. & Grieve, B. 2018. Low-Cost Turbidity Sensor for Low-Power Wireless Monitoring of Fresh-Water Courses. *IEEE Sensors Journal*, 18(11): 4689–4696.

Wiranto Goib, Mambu A Grace, Hiskia, Hermida Putu I Dewa & Widodo Slamet. 2015. Design of Online Data Measurement and Automatic Sampling System for Continuous Water Quality Monitoring.

Wu, D., Wang, H. & Seidu, R. 2020. Smart data driven quality prediction for urban water source management. *Future Generation Computer Systems*, 107: 418–432.

Y Chen & R Compton. 2022. Chen, Y. and Compton, R., 2022. A Calibration-Free pH Sensor Using an In-Situ Modified Ir Electrode for Bespoke Application in Seawater. Sensors, 22(9), p.3286. : 3826.

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).

APPENDICES

APPENDIX A: ThingSpeak and Ethernet shield(1 of 3)

```
#include <SPI.h>

#include <Ethernet.h>

#include <ThingSpeak.h>

// ThingSpeak channel's write API key

const char* apiKey = "X426TIF6UXYCJB2V"; // ThingSpeak write API key

const char* server = "api.thingspeak.com";

// Ethernet shield's MAC address

byte mac [] = {0xE4, 0xA8, 0xDF, 0xED, 0x10, 0xF6};

EthernetClient client;

void setup () {

 Serial.begin(9600);

 // Initialize the Ethernet shield

 Ethernet.begin(mac);

 delay (1000); // Wait for Ethernet shield is fully initialized

// ThingSpeak communication

 ThingSpeak.begin(client);

}
```

APPENDIX B: ThingSpeak and Ethernet shield (2 of 3).

```
void loop () {
int relayStatus = digitalRead (RELAY_PIN);
  // Read sensor values
  float sensor1 pH = analogRead(A0); // Read from analog pin A0
  float sensor2 TDS = analogRead(A1); // Read from analog pin A1
  float sensor3 EC = analogRead(A2); // Read from analog pin A2
  float sensor4 Temperature = analogRead(A3); // Read from analog pin A3
  float sensor5 Turbidity = analogRead(A4); // Read from analog pin A4
int relayStatus = digitalRead (8); // Read relay status from digital pin 8
float temperature = digitalRead (2); // Read temperature sensor from digital pin 2
float water Flow = digitalRead (7); // Read water flow sensor from digital pin 7
  // Send the sensor data to ThingSpeak
ThingSpeak.setField(1, pH);
 ThingSpeak.setField(2, TDS);
 ThingSpeak.setField(3, EC);
ThingSpeak.setField(4, Temperature);
ThingSpeak.setField(5, Turbidity);
ThingSpeak.setField(6, waterFlow);
ThingSpeak.setField(7, relayStatus);
```

APPENDIX C: ThingSpeak and Ethernet shield(3 of 3).

```
// Upload the data to ThingSpeak
  int responseCode = ThingSpeak.writeFields(2586881, apiKey); // ThingSpeak channel ID
  // Print the result to the Serial Monitor
  if (responseCode == 200) {
    Serial.println("Data successfully updated on ThingSpeak.");
  } else {
    Serial.print("Error updating data. Response code: ");
    Serial.println(responseCode);
  }
// wait for 15 seconds before sending the next update
  delay (15000);
}
```

APPENDIX D: **pH code (1 of 2).**

```
// Pin where the pH sensor is connected
#define PH_PIN A0 // Analog pin A0
// Calibration values
float pH_baseline = 6.9; // pH of natural spring water (baseline)
float voltage_baseline = 1.97; // Voltage corresponding to pH 6.9
// Calibration points
float pH_low = 3.10; // pH for acidic solution
float voltage_low = 2.33; // Voltage for pH 3.10
float pH_high = 10.15; // pH for alkaline solution
float voltage_high = 1.59; // Voltage for pH 10.15
void setup () {
  Serial.begin(115200); // Start serial communication.
  pinMode (PH_PIN, INPUT); // Set the pH pin as input
}
void loop () {
  // Read the voltage from the pH sensor
  float voltage = analogRead (PH_PIN) * (5.0 / 1023.0); // Convert to 0-5V range
  float pH = convertVoltageToPH(voltage); // Convert voltage to pH value
// Print the voltage and pH value to the Serial Monitor
  Serial.print("Voltage: ");
  Serial.print(voltage, 2); // Print with 2 decimal places
  Serial.print(" V, pH: ");
  Serial.println(pH, 2); // Print pH value with 2 decimal places
 delay (2000); // Wait for 2 seconds before the next reading
}
```

APPENDIX E: **pH code (2 of 2).**

```
// Function to convert voltage to pH value
float convertVoltageToPH (float voltage) {
  float pH;
if (voltage >= voltage_low && voltage <= voltage_baseline) {
    // Interpolate pH for values between low and baseline
     pH = pH_low + (voltage - voltage_low) * (pH_baseline - pH_low) / (voltage_baseline - voltage_low);
  } else if (voltage <= voltage_baseline && voltage >= voltage_high) {
    // Interpolate pH for values between baseline and high
    pH = pH_baseline + (voltage - voltage_baseline) * (pH_high - pH_baseline) / (voltage_high - voltage_baseline);
  } else {
    // Voltage out of calibration range
    pH = 0; // Indicates an invalid pH reading
  }

  return pH;
}
```

APPENDIX F: Flow rate measurement code.

```
int sensorPin = 2;
volatile long pulse;
unsigned long lastTime;
float volume;
void setup() {
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(sensorPin), increase, RISING);
}
void loop () {
  volume = 2.663 * pulse / 1000 * 60;  // Adjusted for 1-second interval
  if (millis () - lastTime > 1000) {   // 1-second interval
    pulse = 0;
    lastTime = millis();
  }
Serial.print(volume);
  Serial.println(" L/m");
    }
    void increase () {
      pulse++;
    }
```

APPENDIX G: **TDS sensor code (1 of 2).**

```
#define TDS_PIN A1 // Analog pin

float voltage;

float tdsValue;

float calibrationFactor;

void setup() {

  Serial.begin(9600);

// Calibrate using a known TDS value

  float knownTDS = 300.0; // Known TDS value of tap water in ppm

  int sensorValue = analogRead(TDS_PIN); // Read the sensor's analog value

  voltage = sensorValue * (5.0 / 1023.0); // Convert analog reading to voltage

  // Check if initial voltage reading is reasonable

  if (voltage <= 0.0) {

    Serial.println("Error: Initial voltage reading is too low for calibration.");

    while (true); // Halt the program if calibration fails

  }

  calibrationFactor = knownTDS / (voltage * 1000.0); // Calculations of the calibration factor

  Serial.print("Calibration Factor: ");

  Serial.println(calibrationFactor, 2);

}
```

APPENDIX H: **TDS sensor code (2 of 2).**

```
void loop () {
  // Read TDS sensor value
  int sensorValue = analogRead (TDS_PIN);
  voltage = sensorValue * (5.0 / 1023.0); // Convert analog reading to voltage
 // Calculations of TDS value
  float tdsValueRaw = voltage * calibrationFactor * 1000; // Convert voltage to TDS (ppm)
 // Printing voltage and TDS value
  Serial.print("Voltage: ");
  Serial.println(voltage, 2);
  Serial.print("Raw TDS Value: ");
  Serial.println(tdsValueRaw, 2);
  // The conditional statements to handle different TDS ranges
  if (tdsValueRaw < 100) {
    Serial.println("TDS level is low. Water quality is good.");
  } else if (tdsValueRaw >= 100 && tdsValueRaw < 500) {
    Serial.println("TDS level is moderate. Water quality is acceptable.");

 } else if (tdsValueRaw >= 500) {
   Serial.println("TDS level is high. Water quality may be poor.");
 }
// Waiting 15 seconds before the next reading
 delay (15000);
}
```

APPENDIX I: **EC and Temperature sensors code (1 of 4).**

```
#include <OneWire.h>

#include <DallasTemperature.h>

#include <DFRobot_EC.h>

#include <EEPROM.h>

// We define the pins where our sensors are connected

#define ONE_WIRE_BUS 2    // This is the pin where the temperature sensor is connected

#define EC_PIN A2          // This is the pin where the EC (electrical conductivity) sensor is connected

// Setting up objects for the sensors

OneWire oneWire (ONE_WIRE_BUS);        // Setting up the OneWire bus for the temperature sensor

DallasTemperature sensors(&oneWire);     // Creating a DallasTemperature object to interact with the temperature sensor

DFRobot_EC ec;                          // Creating a DFRobot_EC object to interact with the EC sensor

void setup () {

  // Start communication with the computer

  Serial.begin(9600);

  Serial.println("Water Quality Monitoring System is starting...");
```

APPENDIX J: **EC and Temperature sensors code (2 of 4).**

```
  // Initialize the sensors
  sensors. begin ();
  ec. begin ();
  // Let's start by calibrating the EC sensor
  calibrateEC();
}
void loop () {
  // Request the temperature from the sensor
  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0); // Get the temperature in Celsius
 // Read the EC sensor's voltage and calculate the EC value
  float voltage = analogRead(EC_PIN) / 1024.0 * 5000; // Convert the analog reading to
millivolts
  float ecValue = ec.readEC(voltage, temperatureC);   // Convert the voltage to an EC value in
mS/cm
// Display the temperature and EC readings
  Serial.print("Temperature: ");
  Serial.print(temperatureC);
```

APPENDIX K: **EC and Temperature sensors code (3 of 4).**

```
  Serial.println(" °C");
  Serial.print("EC: ");
  Serial.print(ecValue, 2); // Print the EC value with 2 decimal places
  Serial.println(" mS/cm");
  // Depending on the EC value, we determine the type of water we're dealing with
  if (ecValue < 1.0) {
    Serial.println("Water type: Freshwater (Low EC) - Suitable for most uses.");
  } else if (ecValue >= 1.0 && ecValue <= 10.0) {
    Serial.println("Water type: Tap Water or Bottled Water - Generally safe for drinking.");
  } else {
    Serial.println("Water type: Brackish or Saline Water - Not suitable for drinking.");
  }
// Wait for 15 seconds before taking the next reading
  delay (15000);
}
void calibrateEC () {
  // This function helps us calibrate the EC sensor
Serial.println("Let's calibrate the EC sensor...");
// Step 1: Calibrate with a solution of known conductivity (1314 µS/cm)
  Serial.println("Please place the sensor in a 1314 µS/cm calibration solution and press
```

APPENDIX L: **EC and Temperature sensors code (4 of 4).**

```
Enter.");
  while (Serial.available() == 0) {} // Wait until the user presses Enter
  Serial.read(); // Clear the input (i.e., the Enter key press)
  float voltage = analogRead(EC_PIN) / 1024.0 * 5000; // Convert the analog reading to
millivolts
  float temperatureC = sensors.getTempCByIndex(0); // Get the current temperature
  ec. calibration (voltage, temperatureC, "1314"); // Calibrate using the 1314 µS/cm solution
  delay (5000); // Wait a few seconds to allow the sensor reading to stabilize
 // Step 2: Calibrate with a higher conductivity solution (12.88 mS/cm)
  Serial.println("Now place the sensor in a 12.88 mS/cm calibration solution and press Enter.");
  while (Serial.available() == 0) {} // Again, wait until the user presses Enter
  Serial.read(); // Clear the input
  voltage = analogRead(EC_PIN) / 1024.0 * 5000; // Convert the analog reading to millivolts
  temperatureC = sensors.getTempCByIndex(0); // Get the current temperature
  ec.calibration(voltage, temperatureC, "12880"); // Calibrate using the 12.88 mS/cm solution
  delay(5000); // Wait for stable readings
  Serial.println("Calibration is complete! Your sensor is ready to use.");
}
```

APPENDIX M: **Turbidity sensor code (1 of 3).**

```
#include <SPI.h>

// These variables to calculate turbidity from the sensor readings.

float turb_voltx = 0.0; // This stores the turbidity voltage.

int turb = 0; // This stores the calculated turbidity value.

float voltTurbRef = 4.5; // This is a reference voltage for the turbidity sensor.

// Function declarations

void read_tur (); // This function will handle reading and calculating the turbidity.

void setup () {

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

}

void loop () {

  read_tur (); // Call the function to read and calculate the turbidity.

// Display the turbidity results on the Serial Monitor.

  Serial.print("Turbidity Voltage: ");

  Serial.print(turb_voltx);

  Serial.print(" V\tTurbidity: ");

  Serial.print(turb);

  Serial.println(" NTU");
```

APPENDIX N: **Turbidity sensor code (2 of 3).**

```
// Based on the turbidity value, we classify the water and print the result.
  if (turb >= 0 && turb <= 5) {
    Serial.println("Water Type: Clean Water"); // If turbidity is between 0 and 5 NTU, the water is clean.
  } else if (turb > 5 && turb <= 20) {
    Serial.println("Water Type: Cloudy Water"); // If turbidity is between 5 and 20 NTU, the water is cloudy.
  } else {
    Serial.println("Water Type: Very Cloudy or Dirty Water"); // If turbidity is above 20 NTU, the water is very cloudy or dirty.
  }
 delay (15000); // Wait for 15 seconds
}
// This function reads the turbidity sensor and calculates the turbidity value.
void read_tur () {
  turb_voltx = 0.0; // Reset the turbidity voltage accumulator.
// Take multiple readings from the sensor to get a more accurate average value.
  for (int mk = 0; mk < 810; mk++) {
    turb_voltx += (float)analogRead(A0) / 1023.0 * voltTurbRef; // Convert the sensor's analog reading to a voltage and add it to the total.
  }
```

APPENDIX O: Turbidity sensor code (3 of 3).

```
    turb_voltx /= 810.0; // calculations of the average voltage from all the readings.
// Use the average voltage to calculate the turbidity.
    // the coefficients can be adjusted based on calibration.
    if (turb_voltx < 2.5) {
        turb = 3000; // If the voltage is very low, we assign a high turbidity value (e.g., very dirty water).
    } else {
        turb = (-1000 * turb_voltx * turb_voltx) + (5000 * turb_voltx) - 3000; // Calculate turbidity using a quadratic formula.
    }
}
```

APPENDIX P: Combination of sensors and DC pump and relay (1 of 7).

```
#include <OneWire.h>

#include <DallasTemperature.h>

// Pin connected to the relay module that controls the pump

const int relayPin = 7;

// Pin assignments for different sensors

#include <OneWire.h>

#include <DallasTemperature.h>

// Pin assignments

const int relayPin = 7;          // Relay pin to control the pump

const int ecSensorPin = A0;        // Electrical Conductivity (EC) Sensor connected to analog
pin A0

const int tdsSensorPin = A1;       // Total Dissolved Solids (TDS) Sensor connected to analog
pin A1

const int phSensorPin = A2;        // pH Sensor connected to analog pin A2

const int turbiditySensorPin = A3; // Turbidity Sensor connected to analog pin A3

const int tempSensorPin = 2;       // Digital pin 2 connected to DS18B20 temperature sensor

const int flowSensorPin = 4;       // Digital pin 4 connected to water flow sensor

// Define acceptable ranges for sensor readings

const float ecThresholdLow = 0.50;            // Minimum acceptable value for EC (Electrical
Conductivity)

const float ecThresholdHigh = 10.25;       // Maximum acceptable value for EC

const int tdsThresholdLow = 20;            // Minimum acceptable value for TDS (Total Dissolved
Solids)

const int tdsThresholdHigh = 60;           // Maximum acceptable value for TDS

const float phThresholdLow = 2.00;         // Minimum acceptable pH value

const float phThresholdHigh = 7.00;        // Maximum acceptable pH value

const int turbidityThresholdLow = 200;     // Minimum acceptable turbidity level
```

APPENDIX Q: **Combination of sensors and DC pump and relay (2 of 7).**

```
const int turbidityThresholdHigh = 800;    // Maximum acceptable turbidity level
const float temperatureThresholdLow = 19.00; // Minimum acceptable temperature
const float temperatureThresholdHigh = 30.00; // Maximum acceptable temperature
const float flowThresholdMin = 0.1;        // Minimum acceptable flow rate
const float flowThresholdMax = 5.0;        // Maximum acceptable flow rate
// Set up the DS18B20 temperature sensor
OneWire oneWire(tempSensorPin);          // Create an instance of OneWire for the temperature sensor
DallasTemperature sensors(&oneWire);         // Create an instance of DallasTemperature for reading temperatures
volatile int pulseCount = 0;   // Variable to count pulses from the flow sensor
float flowRate = 0.0;          // Variable to store the calculated flow rate in Liters per minute
void setup() {
  // Initialize the relay pin to control the pump
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);  // Start with the pump turned off
// Begin serial communication for debugging
  Serial.begin(9600);
// Set sensor pins as input
  pinMode(ecSensorPin, INPUT);
```

APPENDIX R: Combination of sensors and DC pump and relay (3 of 7).

```
  pinMode(tdsSensorPin, INPUT);
  pinMode(phSensorPin, INPUT);
  pinMode(turbiditySensorPin, INPUT);
  pinMode(flowSensorPin, INPUT);
 // Initialize the DS18B20 temperature sensor
  sensors. begin();
 // Set up an interrupt to count pulses from the flow sensor
  attachInterrupt(digitalPinToInterrupt(flowSensorPin), countFlowPulse, RISING);
}
void loop() {
  // Read and average sensor values
  float ecValue = averageSensorReadings(ecSensorPin) * (5.0 / 1023.0) * 3.5; // Convert analog reading to EC in mS/cm
  int tdsValue = averageSensorReadings(tdsSensorPin); // Read TDS in ppm
  float phValue = averageSensorReadings(phSensorPin) * (5.0 / 1023.0) * 3.5; // Convert analog reading to pH
  int turbidityValue = averageSensorReadings(turbiditySensorPin); // Read turbidity in NTU
// Calculate flow rate in Liters per minute
  flowRate = pulseCount / 7.0;  // Calculate flow rate based on pulse count
```

APPENDIX S: Combination of sensors and DC pump and relay (4 of 7).

```
  // Get temperature from DS18B20 sensor
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);  // Read temperature in Celsius
 // Adjust EC value based on temperature
  float compensatedECValue = compensateECForTemperature(ecValue, temperature);
// Print sensor values for monitoring
  Serial.print("EC: "); Serial.print(compensatedECValue);
  Serial.print(" TDS: "); Serial.print(tdsValue);
  Serial.print(" pH: "); Serial.print(phValue);
  Serial.print(" Turbidity: "); Serial.print(turbidityValue);
  Serial.print(" Temperature: "); Serial.print(temperature);
  Serial.print(" Flow Rate: "); Serial.println(flowRate);
 // Check if all sensor readings are within the acceptable ranges
  if (isWithinRange(compensatedECValue, ecThresholdLow, ecThresholdHigh) &&
      isWithinRange(tdsValue, tdsThresholdLow, tdsThresholdHigh) &&
      isWithinRange(phValue, phThresholdLow, phThresholdHigh) &&
      isWithinRange(turbidityValue, turbidityThresholdLow, turbidityThresholdHigh) &&
      isWithinRange(flowRate, flowThresholdMin, flowThresholdMax) &&
```

APPENDIX T: Combination of sensors and DC pump and relay (5 of 7).

```
        isWithinRange (temperature, temperatureThresholdLow, temperatureThresholdHigh)) {
        // Turn on the pump if all values are within range
        turnPumpOn ();
    } else {
        // Turn off the pump if any value is out of range
        turnPumpOff ();
    }

    // Reset pulse count for the next measurement period
    pulseCount = 0;
// Wait for 5 seconds before repeating the loop
    delay(5000);
}
// Function to turn the pump on
void turnPumpOn() {
    digitalWrite(relayPin, HIGH);  // Activate the relay to turn on the pump
    Serial.println("Pump is ON");
}
```

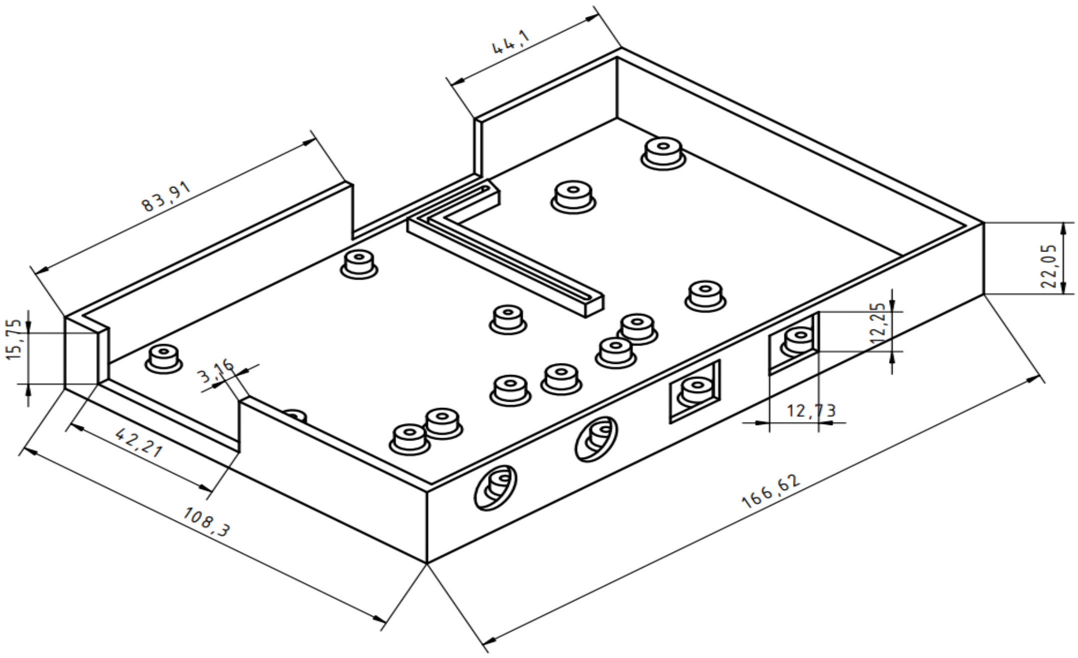
APPENDIX U: Combination of sensors and DC pump and relay (6 of 7).

```
// Function to turn the pump off
void turnPumpOff() {
  digitalWrite(relayPin, LOW);   // Deactivate the relay to turn off the pump
  Serial.println("Pump is OFF");
}
// Function to read multiple values from a sensor and return the average
float averageSensorReadings(int pin) {
  const int numReadings = 10;  // Number of readings to average
  int sum = 0;
  for (int i = 0; i < numReadings; i++) {
    sum += analogRead(pin);  // Read the sensor value
    delay(10);  // Short delay between readings
  }
  return sum / (float)numReadings;  // Return the average value
}
// Function to check if a sensor value is within the acceptable range
 isWithinRange(float value, float low, float high) {
```

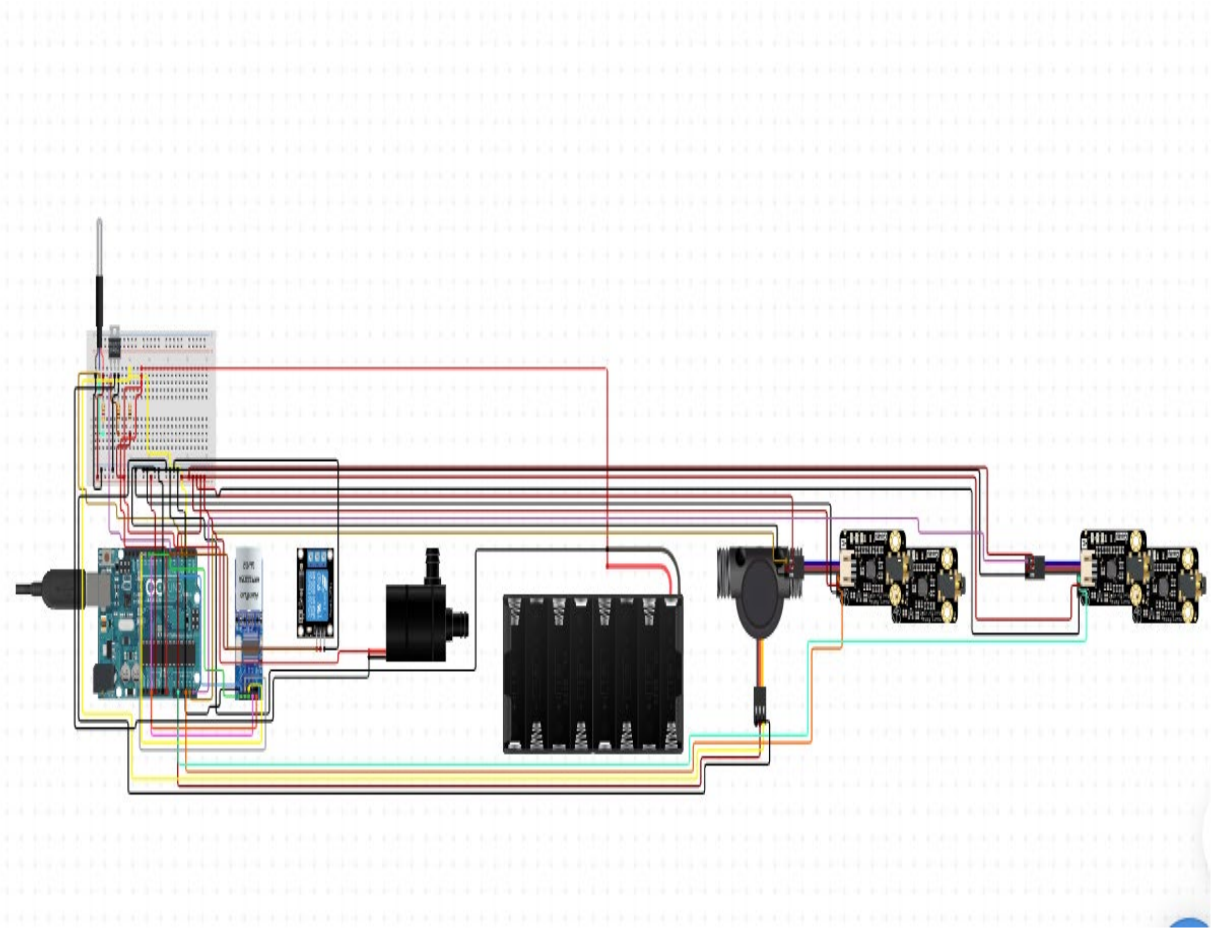
APPENDIX V: Combination of sensors and DC pump and relay (7 of 7).

```
    return (value >= low && value <= high);  // Return true if value is within range
}
// Function to compensate the EC value based on temperature
float compensateECForTemperature(float ecValue, float temperature) {
    // Adjust EC value based on temperature deviation from a reference
    float referenceTemperature = 30.0;  // Reference temperature in Celsius
    float temperatureCoefficient = 0.02; // Temperature coefficient (2% per °C)
    float temperatureDifference = temperature - referenceTemperature;
    float compensatedECValue = ecValue * (1 + temperatureCoefficient * temperatureDifference);
    return compensatedECValue;
}
// Interrupt service routine to count pulses from the flow sensor
void countFlowPulse() {
    pulseCount++;  // Increment pulse count
}
```

APPENDIX W: 3D Box Design



APPENDIX X: **Simulation of this Monitoring and control system using circuito**



APPENDIX Y: ATmega2560 Arduino Specifications

ARDUINO.CC | PROFESSIONAL | EDUCATION

STORE PRODUCTS EDUCATION PROFESSIONAL

Overview Tech Specs Conformities Documentation FAQs

Tech specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

APPENDIX Z: Specifications of water flow sensor from Communica

Overview

Water flow sensor consists of magnetic core, rotating impeller, external casing and sensor and a hall-effect sensor. When water flows through the rotor, rotor rolls, it activates the magnetic core to trigger switch action speed changes with different rate of flow. The hall-effect sensor outputs the corresponding pulse signals, users can get the flow speed via detecting the pulse. It is suitable to detect flow in water dispenser or coffee machine.

There are also lots of other water flow sensors in other diameters for your choice.

The water flow sensor outputs pulse proportional to the water flow with: Pulse Frequency = Flow *7.5. That is, if the output frequency is 48, then the water flow: 48/7.5= 6.4(L/min)

Specifications

- Mini. Wokring Voltage : DC 4.5V
- Max. Working Current : 15mA(DC 5V)
- Working Voltage : 5V~24V
- Flow Rate Range : 1~30L/min
- Load Capacity : ≤10mA(DC 5V)
- Operating Temperature : ≤80°C
- Liquid Temperature : ≤120°C
- Operating Humidity : 35%~90%RH
- Water Pressure : ≤2.0MPa
- Storage Temperature : -25~+80°C
- Storage Humidity : 25%~95%RH