

LoRa Water Quality Management System

Cole Hadley, Patricio Benitez Palermo, Matthew Sharp

Instructor: Dr. Aldo Morales

Technical Advisors: Dr. Sedig Agili and Dr. Aldo Morales

Technical Report Submitted in Partial Fulfillment of the
Senior Electrical Capstone Design Course at Penn State Harrisburg

EE 406W, Section 1

Fall Semester, 2025

Submitted December 12, 2025

Abstract

To provide high-quality water to their customers, utilities must monitor key parameters which contribute to the overall health, safety, and taste of drinking water. This paper proposes a water-quality management architecture that uses a distributed network of low-power, battery-operated sensor nodes that report turbidity, pH, and temperature to a central gateway via LoRa communications, a wireless communications standard that features significantly lower power consumption and similar range compared to cellular at the trade-off of lower data rates. The gateway detects anomalies, visualizes real-time data through a dashboard, and applies an artificial neural network to forecast future water quality metrics.

Based on this proposal, a fully integrated prototype of the system is presented. Sensor nodes acquire environmental data and transmit it over LoRa to the gateway, which updates a visual dashboard and issues mobile alerts to field personnel. A prototype artificial neural network predicts future water conditions using historical telemetry. To facilitate sensing of the physical water environment, the design of a complete signal conditioner subsystem is presented; this subsystem integrates an analog frontend with digital post-processing to produce highly accurate sensor telemetry data for use with the entire system.

Acknowledgements

We extend our gratitude to our capstone project technical advisors, Dr. Aldo Morales and Dr. Sedig Agili for contributing their expertise to our project. Both of our advisors were crucial for the development and exploration of sensor interfacing, wireless communication, and artificial neural networks. Additionally, we would like to thank Dr. Yen-Chih Chen and J. Mitchell Spear for their consultation on the water treatment process and helping us understand how water quality is measured in the context of a water utility company. We would also like to recognize the technicians and engineers of the Reading, Muhlenberg, and Maiden Creek water authorities for taking the time to complete our survey, helping us better understand the challenges that water utilities face when evaluating water quality. For the valuable feedback and support during both the senior electrical capstone proposal and design phases of our project, we would like to thank Dr. Kiana Karami and Dr. Aldo Morales. Finally, we are extremely grateful for the support of Terry Tirko, for facilitating lab equipment and space for the completion of our project and supporting our efforts through numerous technical challenges.

Table of Contents

Table of Figures	viii
Table of Tables	xii
Table of Equations	xiv
1. Introduction	1
1.1. Introduction	1
1.2. Technical History	1
1.2.1. Early Developments in Water Quality Monitoring	2
1.2.2. Legislative Milestones: Clean Water Act and Safe Drinking Water Act	2
1.2.3. Technological Advancements in the 1970s and 1980s.....	3
1.2.4. Emergence of Wireless Sensor Networks in the 1990s and 2000s.....	4
1.2.5. Advancements in Low-Power, Long-Range Communication Technologies	4
1.2.6. Current State and Gaps in Water Quality Monitoring	5
1.3. Review of Technical Literature.....	7
1.3.1. IoT in Modern Water Utilities	7
1.3.2. Applications of LoRa in IoT Water Systems	8
1.3.3. Wireless Water Quality Monitoring.....	9
1.3.4. Usage of Artificial Neural Networks in Water Quality Management	10
1.4. Lifecycle of Similar Products	11
2. Experimental Method	13
2.1. Engineering Requirements Development.....	13
2.1.1. Environmental and Safety Concerns	13
2.1.2. Legal, Ethical, and Political Concerns	14
2.1.3. Sustainability	15
2.1.4. Customer Requirements	15

2.1.5. Constraints	20
2.1.6. Standards	21
2.2. Engineering Requirements	22
2.3. Level 1 and 2 Functional Decomposition	25
2.3.1. Level 1 Decomposition	25
2.3.2. Level 2 Decomposition	26
2.4. Subsystem Development.....	31
2.4.1. Signal Conditioner Circuit Design <Matthew Sharp>	31
2.4.2. Signal Conditioner Theory	35
2.4.3. Signal Conditioner Simulation.....	37
2.4.4. pH Sensor Preamplifier Circuit Design <Cole Hadley>	41
2.4.5. pH Sensor Preamplifier Theory	43
2.4.6. pH Sensor Preamplifier Simulation.....	45
2.4.7. RP2350 Microcontroller	46
2.4.8. Buck-Boost Regulator	50
2.4.9. Switched Capacitor Voltage Converter	53
2.4.10. LoRa Modulator	54
2.4.11. Turbidity Sensor.....	60
2.4.12. Temperature Sensor	60
2.4.13. Multiplexer	61
2.4.14. Analog to Digital Converter (ADC).....	61
2.4.15. Digital Potentiometers.....	63
2.4.16. SPI Flash Memory	64
2.4.17. Logic Level Translator.....	64
2.4.18. Neural Net Theory	66

2.4.19. Neural Net Design	66
2.4.20. Gateway	67
2.4.21. Gateway Receiver	69
2.4.22. Gateway Dashboard	71
2.4.23. Sensor Node PCB	75
2.4.24. Sensor Node Power Management.....	84
3. System Integration and Test Results	87
3.1. Digital Potentiometer Operation	87
3.2. Level Translator Operation.....	89
3.3. Dual 4:1 Analog Multiplexer Operation.....	91
3.4. Switched Capacitor Operation	93
3.5. Buck-Boost Regulator Operation	93
3.6. 16-bit Analog-to-Digital Converter Test	97
3.7. SPI Flash Operation.....	100
3.8. LoRa Modulator Test	101
3.9. LoRa Range Test.....	103
3.10. pH Sensor Pre-Amplifier Operation.....	104
3.11. Turbidity Sensor Operation	104
3.12. RTD Test	105
3.13. Software-Defined Instrumentation Amplifier Test.....	106
3.14. Complete Operational Test of the Signal Conditioner Subsystem.....	110
3.15. RP2350 Power Management Test.....	113
3.16. Full Node Firmware Test.....	114
3.17. Neural Net Prediction Test for Water Quality Metrics.....	115
3.18. Gateway Dashboard Operation.....	117

3.19. Gateway Alert Generation Test.....	118
3.20. Full System Integration Test.....	119
4. Project Management	124
4.1. Project Work Breakdown Structure (WBS)	124
4.2. Project Schedule (Gantt Chart).....	126
4.3. Project Budget.....	128
5. Summary and Conclusion	130
5.1. After-Action Report.....	131
5.2. Future Work.....	133
References	136
Appendix A: Customer Survey	145
Appendix B: Customer Survey Results	148
Appendix C: Sensor Node Firmware & PCB Design Access	150
Appendix D: Test Plan Tables.....	151
Appendix F: Sensor Node PCB Schematic.....	177

Table of Figures

Figure 1: Example LoRa Water Quality Monitoring Architecture [11]	5
Figure 2: Honeywell Merlin NB-IoT Clip-On Water Meter [13]	7
Figure 3: Block Diagram of Proposed LoRa-Based Water Quality Monitoring System [16]....	9
Figure 4: In-Situ Real-Time Water Quality Monitoring System [18].	10
Figure 5: Estimated lifecycle of LoRa Water Quality Monitoring System	12
Figure 6: Hierarchical Representation of Marketing Requirements.....	20
Figure 7: Level 1 Decomposition of the LoRa Water Quality Management System.....	26
Figure 8: Level 2 Decomposition of the LoRa Water Quality Management System.....	27
Figure 9: Block Diagram of Signal Conditioner Subsystem	32
Figure 10: Signal Conditioner Circuit.....	34
Figure 11: Stage 1 of Software-Defined Instrumentation Amplifier.....	34
Figure 12: Stage 2 of Software-Defined Instrumentation Amplifier.....	35
Figure 13: Digital Potentiometers for Software-Defined Instrumentation Amplifier.....	35
Figure 14: PSpice Simulation Schematic of Software-Defined Instrumentation Amplifier ..	38
Figure 15: Simulated Temperature Sensor Plot.....	39
Figure 16: Simulated pH Sensor Plot.....	40
Figure 17: Simulated Turbidity Sensor Plot.....	41
Figure 18: pH Preamplifier Circuit.....	43
Figure 19: PSpice Simulation Schematic for pH preamplifier	45
Figure 20: Simulated pH Preamplifier Plot.....	46
Figure 21: High-Level Sensor Node Finite State Machine	49
Figure 22: Typical Application Circuit for the TPS63000 Buck-Boost Regulator	51
Figure 23: LM2660 Simplified Internal Design [46]	54
Figure 24: LM2660 Typical Application Circuit [46]	54
Figure 25: Example SPI Transaction for the sx1262 IC [48]	55
Figure 26: SX1262 Transmit and Receive Operation Initialization Process.....	56
Figure 27: RDT3.0 Transmitter FSM Diagram	57
Figure 28: RDT3.0 Receiver FSM Diagram	57
Figure 29: LoRa Packet Structure [48]	58

Figure 30: Packet Data Structure Used to Encode Sensor Node Telemetry Data	59
Figure 31: Typical I ² C Bus Topology [54]	63
Figure 32: Typical Timing Diagram of an I ² C Transaction [54]	63
Figure 33: Internal Driving Circuitry of the LSF0102 Logic Level Translator [57].....	65
Figure 34: Raw Water Quality Data Set for use with Neural Network Training	67
Figure 35: Cleaned Data for use with Neural Network Training.....	67
Figure 36: Gateway Design with Receiver PCB	69
Figure 37: Gateway Receiver Board PCB Schematic	70
Figure 38: Gateway Receiver PCB Layout	70
Figure 39: Gateway Receiver PCB, Populated.....	71
Figure 40: LoRa Water Quality Management System Dashboard Design.....	72
Figure 41: Dashboard Node Data History Window	72
Figure 42: Dashboard with Alert Banner On-Screen	73
Figure 43: Mobile Water Quality Alert via Email.....	74
Figure 44: Customizable Parameter Settings in Dashboard	74
Figure 45: JLC04161H-7628 PCB Stackup [60].....	77
Figure 46: Buck-Boost Power Regulation PCB Layout	78
Figure 47: Switched Capacitor Voltage Converter Layout.....	79
Figure 48: RP2350 and Peripherals PCB Layout	79
Figure 49: Signal Conditioner Subsystem PCB Layout	80
Figure 50: RF Frontend PCB Layout	80
Figure 51: Full 2D Render of Design as produced in KiCAD.....	81
Figure 52: Full 3D Render of PCB, Populated.....	81
Figure 53: Full 3D Render of PCB, Unpopulated	82
Figure 54: 3D Render of Reverse side of PCB, Populated.....	82
Figure 55: Final Ordered PCB, Unpopulated	83
Figure 56: Final Ordered PCB, Populated	84
Figure 57: Alkaline D-Cell Battery Discharge Curve under a 10Ω Load [61].....	86
Figure 58: Digital Potentiometer Testing Code.....	88
Figure 59: Serial Data Transaction over I ² C with Oversized Pull-Up Resistors.....	90
Figure 60: Serial Data Transaction over I ² C with Well-Sized Pull-Up Resistors	90

Figure 61: Noise Floor of the TMUX1309 and Oscilloscope	92
Figure 62: Attenuation of Sinusoidal Signal through TMUX1309	92
Figure 63: TPS63000 Very Thin Small Outline No-Lead (VSON) Package [45].....	94
Figure 64: Schematic for the TPS63000 Custom Evaluation Module	94
Figure 65: PCB Layout for the TPS63000 Custom Evaluation Module	95
Figure 66: 3D Render of the TPS63000 Custom Evaluation Module	95
Figure 67: Final Assembled TPS63000 Evaluation PCB In Test Setup.....	96
Figure 68: TPS63000 Regulator Output vs. Input Voltage.....	97
Figure 69: MCP3425 ADC Measurement Error	98
Figure 70: MCP3425 ADC Test Code	99
Figure 71: Flash Memory Dump of First 256-Byte Page after Chip Erasure.....	101
Figure 72: Flash Memory Dump of First 256-Byte Page after Write Operation.....	101
Figure 73: LoRa Packet Capture on SDR, as Produced in SDR++.....	102
Figure 74: LoRa Range Testing Map, as Produced in Google Earth.....	103
Figure 75: DC Offset Voltages versus Potentiometer Wiper Step Location.....	106
Figure 76: DC Offset Voltage Error with Respect to Ideal, and Each Other	107
Figure 77: Gain Resistance Measurements Versus Wiper Step Location.....	107
Figure 78: Software-Defined Instrumentation Amplifier Gain Resistance Error	108
Figure 79: pH Sensor Voltage as measured by the Signal Conditioner versus Actual pH....	111
Figure 80: RTD Temperature vs. Input Voltage	111
Figure 81: Turbidity Sensor Measurement vs. Input Voltage.....	112
Figure 82: Gateway Packet Data Console Log	114
Figure 83: RMSE for Each Time Step, with Open-Loop Forecasting	115
Figure 84: RMSE for the 18 Test Batches, with Normalized Data.....	116
Figure 85: Plot of open-Loop forecast for a single batch with normalized data	116
Figure 86. Plot of closed loop forecast for a single batch with normalized data.....	117
Figure 87: Dashboard Alert Displays	118
Figure 88: Water Quality Email Alerts.....	119
Figure 89: Full System Integration Test Setup for Sensor Node	120
Figure 90: Full System Integration Test Setup for Gateway.....	121
Figure 91: Full System Integration Test Water Turbidity, pH, and Temperature	122

Figure 92: Work Breakdown Structure	125
Figure 93: Star and Mesh Networking Topologies [67]	133

Table of Tables

Table 1: Customer Primary Requirements	16
Table 2: System Capabilities Subcategory Breakdown	17
Table 3: System Feedback & Notifications Subcategory Breakdown	18
Table 4: Ease of Use Subcategory Breakdown.....	18
Table 5: Deployment Flexibility Subcategory Breakdown	19
Table 6: Approved Methodologies and Relevant Standards for pH and Turbidity Analysis	21
Table 7: Engineering Requirements for the LoRa Water Quality Management System	23
Table 8: pH Sensor Function Table	27
Table 9: Turbidity Sensor Function Table.....	27
Table 10: Temperature Sensor Function Table.....	27
Table 11: Signal Conditioning Function Table.....	28
Table 12: Data Collection Wireless Transceiver Function Table.....	28
Table 13: Data Processing Wireless Transceiver Function Table	29
Table 14: Final Data Processing Function Table.....	29
Table 15: Anomaly Detection Function Table.....	29
Table 16: Predictive Neural Network Function Table	29
Table 17: Decision-making Logic Function Table	30
Table 18: Dashboard Generation Function Table.....	30
Table 19: Alert Generation Function Table.....	30
Table 20: Comparison of Microcontroller Platforms	47
Table 21: Sensor Node Power Consumption, Active State.....	85
Table 22: Sensor Node Power Consumption, Dormant State	85
Table 23: Component Value Selections for TPS63000 Evaluation Module Feedback Path ..	96
Table 24: Gain Selection of the Software-Defined Instrumentation Amplifier	109
Table 25: DC Offset Selection of the Software-Defined Instrumentation Amplifier	110
Table 26: Output Voltage of the Software-Defined Instrumentation Amplifier	110
Table 27: Input Voltage Calculation for the Software-Defined Instrumentation Amplifier	110
Table 28: Water Quality Telemetry from Full-System Integration Test.....	122
Table 29: Gantt Chart.....	127

Table 30: Engineering Labor Cost of Project.....	128
Table 31: Parts Cost of Project.....	129
Table 32: Total Cost of Project.....	129

Table of Equations

Equation 1: Water Quality Index (WQI) Calculation used in [19]	11
Equation 2: Output Voltage of the Software-Defined Instrumentation Amplifier	33
Equation 3: Input Voltage of the Software-Defined Instrumentation Amplifier	33
Equation 4: Cutoff Frequency of a First-Order Passive Low Pass Filter.....	36
Equation 5: Software-Defined Instrumentation Amplifier First Stage Output Voltage.....	36
Equation 6: Instrumentation Amplifier Gain.....	36
Equation 7: pH Sensor Pre-Amplifier Gain.....	44
Equation 8: First Stage Output Voltage of the pH Sensor Pre-Amplifier	44
Equation 9: Second-Stage Output Voltage of the pH Sensor Pre-Amplifier	44
Equation 10: Maximum Duty Cycle of TPS63000 in Boost Mode.....	51
Equation 11: Peak Inductor Current of TPS63000 In Boost Mode	52
Equation 12: Feedback Resistor Selection for TPS63000	52
Equation 13: Feedforward Capacitor Selection for TPS63000	52
Equation 14: CVD Temperature-Resistance Relationship for $T > 0 \text{ }^{\circ}\text{C}$ [52].....	60
Equation 15: CVD Temperature-Resistance Relationship for $T < 0 \text{ }^{\circ}\text{C}$ [52].....	60
Equation 16: Voltage to Temperature Relationship of the PT100 RTD	61
Equation 17: Sensor Node Average Current Consumption	86
Equation 18: Buck-Boost Regulator Input Current.....	86
Equation 19: Minimum Pull-Up Resistor Value for Logic Level Translation [57]	89

1. Introduction

1.1. Introduction

To consistently deliver high-quality water to their customers, water utilities must monitor the quality of their water in every part of their system. This often requires labor-intensive manual testing, or sensors which provide this data in real-time. These systems often require a direct wired connection between each sensor and a Programmable Logic Controller (PLC), making retrofitting older plants with them more difficult. Though wireless options for these sensors are available, they communicate over cellular networks, requiring substantial power dissipation thus limiting battery life.

Using LoRa, a wireless communication technology whose name is an abbreviation of “Long Range,” water quality sensors can be implemented with a significantly reduced power consumption and communicate over long distances without needing access to cellular equipment. Our LoRa-based Water Quality Management System will implement a network of sensors which communicate real-time water quality data back to a central gateway. The gateway will integrate this data with AI-driven analytics, and relay system-wide information to technicians and engineers through a visual dashboard and real-time alerts, enabling water authorities to gain valuable insights into the status of their water systems with reduced setup time and impact on legacy operations.

1.2. Technical History

Monitoring water quality has evolved from simple manual sampling methods to more polished, real-time, and automated technology. This is a result of the increased need for accurate, efficient, and comprehensive water quality analysis to ensure the safety of public health and the environment. As demand for clean water increases due to population growth, industrialization, and climate change, the need for innovation in water quality monitoring systems has become more critical.

1.2.1. Early Developments in Water Quality Monitoring

The quality of water has been monitored since ancient history with the Greeks and Romans using visual observation of clarity and building structures such as aqueducts to supply clear water. The ancient cultures appreciated the value of clean water but were only able to use primitive observation methods [1]. After Antonie van Leeuwenhoek discovered the microscope in the 17th century, scientists could now examine waterborne organisms for the first time, paving the way for a better understanding of possible pollutants [2].

With the Industrial Revolution in the 19th century, there was a speeding up of industrialization that resulted in serious water pollution, and hence the development of chemical analysis techniques to measure pollutants like dissolved solids [3]. These advances paved the way for more systematic water quality monitoring. By the mid-20th century, techniques such as the Most Probable Number (MPN) method emerged, which allowed for the estimation of bacterial populations by serial dilution and observation of growth in culture tubes [4]. This was soon followed by the membrane filtration method, which counted bacteria directly on filter membranes, increasing the efficiency and precision of measurements of bacterial contamination [5].

1.2.2. Legislative Milestones: Clean Water Act and Safe Drinking Water Act

Two landmark legislations in the United States played a crucial role in strengthening water quality monitoring. The Clean Water Act (CWA) of 1972 enacted broad-spectrum regulations to avert water pollution, to safeguard the physical, chemical, and biological integrity of the waters of the United States [6]. It required monitoring of industrial waste and sewage treatment facilities while promoting the development of standard methods for measuring water quality. CWA also focused on the restoration of polluted waters, which led to the overall implementation of monitoring practices.

Similarly, the Safe Drinking Water Act of 1974 directly focused on drinking water quality, mandating that water suppliers meet health-based standards for contaminants [7]. The act initiated more testing demands on the public water systems regarding microbial, chemical,

and radiological parameters. Both acts together established a legislative climate in which water quality monitoring became a regulated and organized endeavor in the United States.

1.2.3. Technological Advancements in the 1970s and 1980s

The 1970s and 1980s witnessed the use of advanced analytical instrumentation in water quality analysis. The most notable development during this time was the release of the computerized Gas Chromatography-Mass Spectrometry (GC/MS). This device enabled the precise identification and quantification of organic pollutants in water samples, a major leap in analysis [8]. The United States Environmental Protection Agency (EPA) implemented GC/MS as a standard analytical procedure in the determination of environmental pollutants, and this saw more accurate and detailed water quality determinations. The technology, apart from identifying pollutants more, also enabled the setting of regulatory standards of water quality for the entire country.

During this time, the Supervisory Control and Data Acquisition (SCADA) system deployment began to transform water quality monitoring. SCADA systems enable remote monitoring and management of water treatment processes with real-time data acquisition and process control [9]. SCADA systems enabled utilities to detect issues such as equipment failures or contamination events more quickly, improving operational efficiency and system reliability. Early SCADA systems often relied on wired connections for communication, which were effective for centralized facilities but presented challenges in monitoring widely distributed or remote locations. While these systems were capable of remote monitoring, reliance on wired infrastructure could limit scalability and increase costs. Over time, advancements in wireless communication technologies, such as radio, cellular, and satellite networks, have significantly enhanced SCADA's ability to monitor and control remote assets, reducing infrastructure costs and improving flexibility.

1.2.4. Emergence of Wireless Sensor Networks in the 1990s and 2000s

In the 1990s and early 2000s, the emergence of wireless sensor networks (WSNs) was a groundbreaking environmental monitoring approach. WSNs are made up of widely distributed sensors that wirelessly transmit data on various environmental factors, including water quality. This made it ideal for tracking water quality in vast, remote areas where traditional wired systems would be impractical and costly [10]. These systems were very effective for surveillance of large bodies of water, for instance, rivers, lakes, and reservoirs, where older methods were not very practical.

1.2.5. Advancements in Low-Power, Long-Range Communication Technologies

Over the past few years, there has been an evolution of low-power, wide-area network (LPWAN) technologies that have overcome most of the shortcomings of previous WSNs. One such LPWAN technology is LoRa, which facilitates long-distance communication with minimal power usage. LoRa technology is particularly well-suited for deploying sensor networks in remote areas, allowing devices to operate efficiently on battery power. This makes LoRa a cost-effective solution for monitoring water quality in areas with limited access to traditional infrastructure.

The use of LoRa technology for water quality monitoring has simplified it to develop affordable and scalable solutions. The technology can send data in real-time across vast geographical areas without the need for heavy infrastructure, hence, why it is finding the interest of water management authorities [11]. These are technologies that will improve the measurement of water quality to augment post-treatment monitoring and confirm water for safe consumption before intake. Figure 1 illustrates the architecture of a LoRa-based water quality monitoring system.

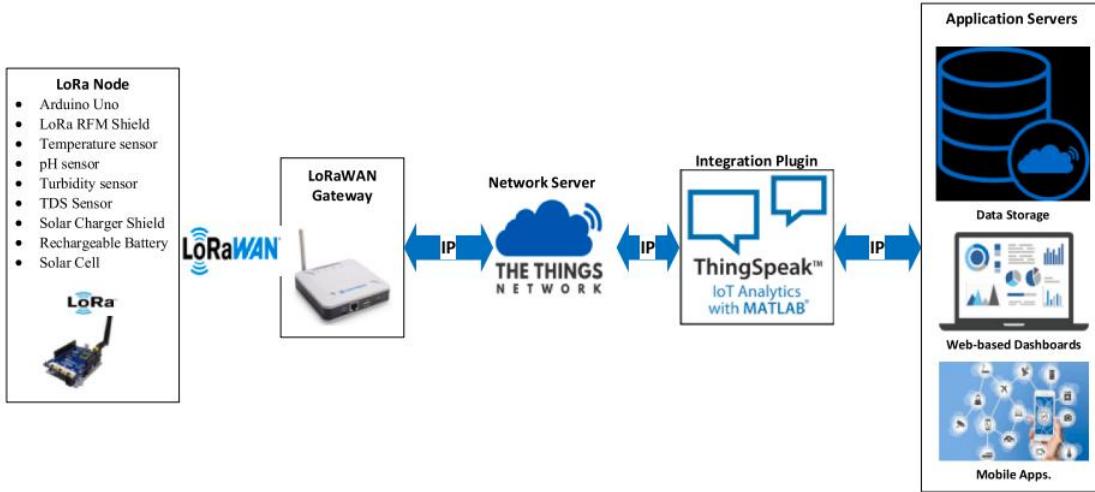


Figure 1: Example LoRa Water Quality Monitoring Architecture [11]

1.2.6. Current State and Gaps in Water Quality Monitoring

Today, water quality monitoring systems have become increasingly sophisticated, incorporating Internet of Things (IoT) devices, advanced sensors, and real-time data analytics. These systems can continuously monitor various parameters such as pH, turbidity, dissolved oxygen, and the presence of specific contaminants [12]. Data collected is transmitted to centralized platforms where it is analyzed to inform decision-making processes. Moreover, AI-driven analytics are beginning to play a critical role in anomaly detection and predictive maintenance, helping to identify issues before they escalate.

Despite these advancements, several gaps and challenges persist:

- 1. Integration with Legacy Systems:** Many existing water treatment facilities operate with legacy SCADA systems that rely on hard-wired connections. Integrating modern wireless sensor networks with these systems can be complex and costly, often requiring significant retrofitting. Overcoming this challenge requires innovative solutions that ensure compatibility without extensive infrastructure overhauls.
- 2. Power Efficiency:** Many wireless communication methods, such as cellular, Wi-Fi, and Bluetooth, consume a significant amount of power to maintain connectivity.

This high-power demand drastically reduces battery life, requiring frequent battery replacements or consistent access to electrical infrastructure. In remote or off-grid environments, these limitations can make long-term sensor deployment impractical. For battery-powered or solar-powered systems, power efficiency is critical to ensure sustained operation without constant maintenance.

3. **Data Management and Analytics:** Traditional SCADA systems often overwhelm operators with large volumes of data, much of it lacking context or immediate relevance. Without clear interpretation, this data becomes difficult to act on and offers little practical value. Effective systems should prioritize delivering focused, actionable insights over raw data streams.
4. **Ensuring Consistent Data Transmission:** Existing technologies such as cellular and Wi-Fi rely on network coverage, which can be unreliable or nonexistent in remote or rural areas. This dependence limits their effectiveness for continuous monitoring in hard-to-reach locations. For applications like water quality monitoring, where consistent data transmission is critical, these coverage gaps pose a serious challenge and can compromise real-time visibility and response.
5. **Cost Constraints:** Budget limitations, especially in smaller towns or developing regions, can delay the adoption of advanced monitoring technologies. Addressing this requires developing scalable solutions that balance performance with affordability, enabling broader access to clean water monitoring.

Water quality monitoring has evolved from basic visual inspections to advanced real-time systems, driven by the need for accuracy and efficiency. Innovations like SCADA, wireless sensor networks, and LoRa technology have improved monitoring capabilities, enabling cost-effective and scalable solutions. However, challenges remain in integrating modern systems with legacy infrastructure, managing data, ensuring power efficiency, and addressing cost constraints. Continued advancements and collaboration are essential to overcoming these barriers and ensuring reliable, real-time water quality monitoring for the future.

1.3. Review of Technical Literature

The integration of wireless Internet of Things (IoT) technology into the daily operations of modern water utility companies has revolutionized how water usage is monitored, managed, and optimized. At the center of this integration are smart metering devices which provide real-time water consumption data for billing and ensure efficient use of company resources. Additionally, wireless technology has also seen applications in the collection and transmission water quality data, particularly for testing and compliance-related applications. This section explores the role of IoT in water quality and usage management, current disadvantages of existing devices, and potential opportunities for improvement and integration of new technologies into utility operations.

1.3.1. IoT in Modern Water Utilities

IoT-based water management systems have become ubiquitous in modern water utilities, primarily with the advent of smart metering. These devices communicate real-time individual usage data back to utility-owned software, where that data can be used for applications such as billing or usage monitoring system-wide. One such IoT device, the Honeywell Merlin NB-IoT, is shown below in Figure 2.



Figure 2: Honeywell Merlin NB-IoT Clip-On Water Meter [13]

This Honeywell IoT Device clips onto existing water meters and transmits real-time water usage data over Narrowband IoT (NB-IoT), a subset of low power wide-area networks (LPWAN) that utilizes existing 3G and 4G cellular infrastructure and frequency bands [13], [14]. Using a low-power communication technology such as NB-IoT allows this device to have a battery life of up to 15 years, making it advantageous over traditional cellular communications, which is quite power-hungry [13], [15].

1.3.2. Applications of LoRa in IoT Water Systems

The low power consumption advantages of NB-IoT systems can also be found in LoRa-based communications systems. Compared to the NB-IoT communication technology used in the Honeywell Clip-On NB-IoT smart meter, LoRa technology enables IoT communications at an even lower power consumption. A typical NB-IoT node has a peak current draw of 100-320 mA and a sleep current of $5\mu\text{A}$, while a comparable LoRa node has a peak current and sleep current of 32mA and $1\mu\text{A}$, respectively. LoRa obtains these increased power savings at the cost of a significantly reduced data rate, with a maximum speed of just 20% of an NB-IoT-based system, although with a much higher immunity to interference [14].

Thus, the power-saving capabilities of LoRa are quite attractive for use in IoT applications where battery life is of greater concern than the data rate, such as in water quality monitoring. One such LoRa-based water quality monitoring system was proposed during the 3rd International Conference on Advancement in Electronics and Communication Engineering in 2023 [16]. The system utilizes a series of sensor nodes that communicate with a central gateway which processes the raw data and communicates with an external microcontroller unit to upload the information to an IoT cloud server over Wi-Fi. A block diagram of the system is shown below in Figure 3.

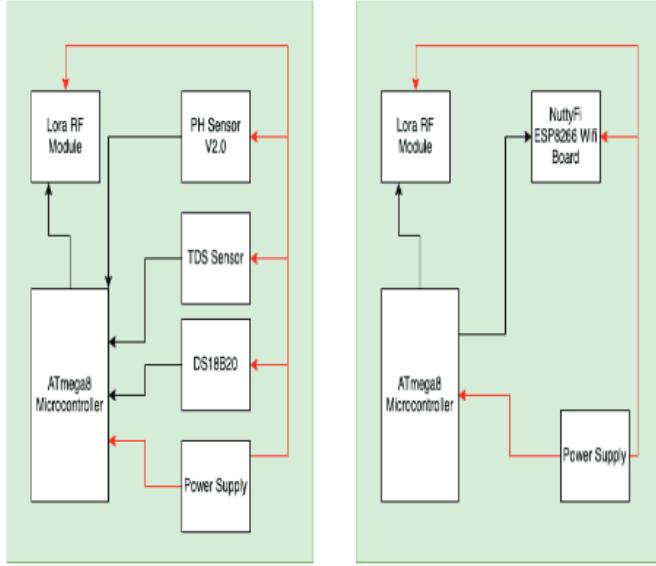


Figure 3: Block Diagram of Proposed LoRa-Based Water Quality Monitoring System [16].

Though this system provides a promising proof-of-concept solution for a LoRa-based water quality management system, the proposed sensor nodes do not support several key metrics of concern for water utility companies, such as turbidity, total suspended solids, or contaminant levels. Furthermore, the system described in the conference paper has not been implemented beyond the prototyping phase, and the raw data is not appreciably processed or analyzed by any software tools, requiring constant human monitoring of the data to identify anomalies or issues in the system. Finally, the system's use of an ATmega8 microcontroller unit in addition to an ESP8266 development board in the gateway adds unnecessary redundancy and failure points to the system, and reduces the overall power savings of the design [16].

1.3.3. Wireless Water Quality Monitoring

Traditional water quality monitoring systems, such as those produced by In-Situ, report real-time data over traditional cellular networks, requiring increased power consumption and therefore larger battery capacity to reach an industry-acceptable battery life. The In-Situ VuLink wireless probe adapter requires three D-cell Lithium Ion Manganese Oxide batteries for a total capacity of 39,000 mAh, to achieve a battery life of up to 12 years with data reporting only once every 24 hours. Using a traditional Alkaline battery formulation

reduces this battery life to three years at the same reporting interval [17]. The VuLink wireless probe adapter is designed to work with a multiparameter sonde, which supports measurements such as temperature, conductivity, pH, turbidity, chloride concentration, nitrate levels, and contaminant concentration [18]. Figure 4 below shows a picture of this system, where the Aqua TROLL Multiparameter Sonde is on the left, and the VuLink Wireless Probe Adapter is on the right.



Figure 4: In-Situ Real-Time Water Quality Monitoring System [18].

In the absence of the wireless probe adapter, the multiparameter sonde can write data to an RS485, Modbus, SDI-12, or Bluetooth output, enabling integration with existing PLC infrastructure. Additionally, the multiparameter probe can be powered by an external DC input, meaning battery life is not a concern for permanent applications where power accessibility is not a concern [6].

1.3.4. Usage of Artificial Neural Networks in Water Quality Management

Though there are several commercially available water quality sensors capable of providing real-time data on a variety of metrics, the raw telemetry provided by these sensors still needs to be manually reviewed and monitored by a human to determine where issues or anomalies may arise within the water system. Thus, one significant area for improvement with these sensors is the integration of artificial intelligence and artificial neural networks (ANNs) with this raw data, meaning anomalies and issues with the system

can be detected outside of a simple check of the incoming data against some pre-set threshold. A potential implementation of neural networks in the prediction of water quality index was proposed in a 2023 paper by Murivhami, Tartibu, and Olayode [19], in which previously-obtained water quality data on seven key metrics was used to train an ANN. The ANN was then used to predict the future water quality index of the system, which was computed using a weighted average calculation shown below in Equation 1.

Equation 1: Water Quality Index (WQI) Calculation used in [19]

$$WQI = \frac{\sum_{i=1}^N q_i \times w_i}{\sum_{i=1}^N w_i}$$

Once trained, the ANN was able to predict the water quality of the system with a typical deviation of less than 1%. Using an ANN to predict future water quality metrics based on real-time data allows water utility companies to adjust equipment and take corrective action on their systems before problems arise, ensuring continuous high-quality water delivery to their customers. Additionally, the forecasted data can be used to create advanced warning systems and execute remedial actions.

1.4. Lifecycle of Similar Products

The lifecycle for an electronic product can be portrayed with the following four stages: Development, Growth, Maturity, and Decline. The development stage includes the design and manufacturing of the product. In the growth stage, the product is introduced to the market and there is an increasing demand for the product. It includes the distribution, operation, and maintenance of the product [20]. The product reaches maturity when sales reach their peak, manufacturing begins to be phased out and replaced by improved products, and maintenance stops being offered. Finally, the decline or end stage occurs when the product is no longer manufactured and the remaining products in the field are replaced, until there are no products in operation, making the end-of-life of the product.

The design and manufacturing stage time depend on the complexity of the electronic product. A similar product to our LoRa Water Quality Monitoring System is the TX3100

Wireless Transmitter from Sensorex, which replaces the TX3000. It measures pH and Oxidation-Reduction Potential (ORP) and transmits data wirelessly [21]. The TX3100 to have been introduced around 2018 [22], while the TX3000 appears to have been introduced before 2012 [23]. This gives the product a lifecycle, from design to end-of-life, of about 10 years, with an additional estimated 10 years of decline, before the products in operation are replaced. This gives a total life cycle of about 20 years for water quality monitoring products similar to our LoRa Water Quality Monitoring System. The figure below shows an estimated time for each of the four stages. In reality, the decline and maturity of an older product and the development of a newer product happen simultaneously, thus the time from introduction of an older product to a newer product is less than the whole lifecycle of a product. The LoRa Water Quality Management System is expected to have a slightly shorter life cycle than previous technologies, due to rapid acceleration of technical development, making older products obsolete more quickly.

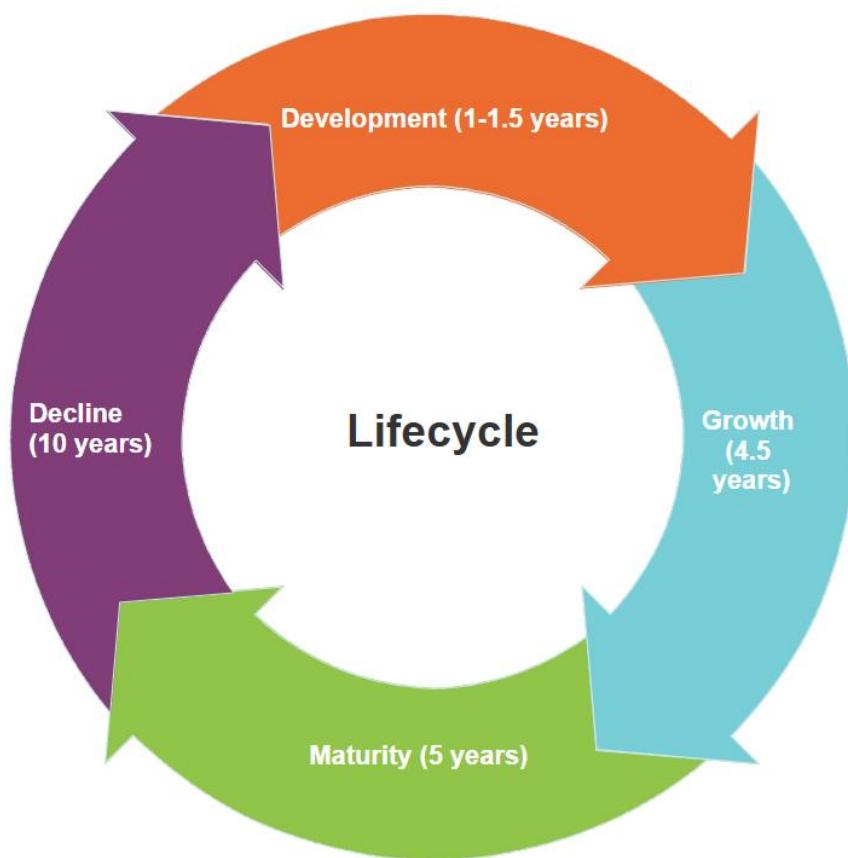


Figure 5: Estimated lifecycle of LoRa Water Quality Monitoring System

2. Experimental Method

2.1. Engineering Requirements Development

2.1.1. Environmental and Safety Concerns

The environmental impact of the LoRa Water Quality Management system is driven mainly by the sensing hardware and radio frequency (RF) electronics. This project uses off-the-shelf sensors sourced from DFRobot and Amazon rather than custom parts. These commercially available sensors are designed for water-borne applications, but if used in real installations, the utility must verify that any materials in contact with drinkable water must meet its own standards and relevant government regulations, such as the Safe Drinking Water Act [24], the EPA's Lead and Copper Rule [25], and NSF/ANSI Standard 61 [26], which control substances allowed to be in contact with drinking water in a utility company's distribution system. The sensor nodes are intended to be mounted indoors within water treatment facilities or customer water utility connection points, which reduces weather exposure but still requires some form of protection against humidity and dust. At end-of-life, proper recycling of PCBs and batteries should be undertaken instead of discarding them as general waste.

Safety concerns for this system mainly involve placing electronics in damp industrial settings and operating RF devices around humans. Each sensor node that is placed in line with the water system is being operated by low-voltage rails generated by D-cell batteries. This keeps shock risk low compared to mains-powered equipment. Additionally, the electronics are designed to be enclosed so that no live conductors are exposed during operation. On the RF side, the sensor node is utilizing LoRa technology and will be adhering to FCC standards which rules limits on exposure and emissions. According to FCC guidance, for any device that fall under these regulations, there is a very low level of exposure to RF radiation. When there are high RF levels, this can cause heating of biological tissue which can be harmful. However, the sensor nodes transmit at low power, meaning very little heating of tissue and effectively no harmful biological effects [27].

2.1.2. Legal, Ethical, and Political Concerns

The LoRa Water Quality Management system must comply with both drinking water regulations and radio communication regulations. For water standards, the system is designed such that it follows EPA drinking water regulations and Pennsylvania Maximum Contaminant Levels for both turbidity and pH. This is done by providing frequent and accurate measurements from the sensors. On the other hand, the nodes and gateway of the system will communicate on the 902-928MHz band. Thus, they must comply with FCC Title 47 Part 15.247 standards [28]. Key requirements include limits on transmission power, pseudo-random frequency hopping between channels, and restrictions on how long transmissions should last.

Ethically, the management system is intended to support safe drinking water by providing more frequent monitoring, rather than replacing lab certified measuring. The only information that will be wirelessly transmitted is the node identification number and the associate sensor readings of that node. So, there is no personal or customer identifying information that is being collected. The transmission is not encrypted; however, the data packets are processed such that they are difficult to interpret without knowledge of the system, as the packet data itself is encoded in its raw binary format, rather than an easily decoded plain-text packet. Furthermore, the system uses a unique symbol to identify itself over other LoRa-based networks like LoRaWAN, meaning that dedicated LoRa receiver devices would also need knowledge of this symbol to receive data transmitted within this network. Therefore, the main ethical concern is that the measurements that are being collected are accurate and reliable so that operators are not mislead.

Politically, this system is aimed at being a low-cost option that does not depend on Wi-Fi or cellular infrastructure, which can be appealing to small borough water authorities or facilities with limited funds. Since the nodes utilizes an unlicensed spectrum, they can provide continuous monitoring of a water system even in the event of traditional network interruptions. As long as power is supplied to the nodes and gateway, communications of the data can continue. Decisions about whether the data that is collected is reported to the public or kept internally would be decided by the water authority using the system. However, the basic idea of this system is to provide another tool that they can adopt

without subscriptions or major infrastructure changes, making the system easily integrated with existing data-reporting and management infrastructure.

2.1.3. Sustainability

A main sustainability consideration of the LoRa Water Quality Management System was the design choice of using three common D-cell batteries per node. The node hardware and firmware were designed with low power consumption and power management in mind. This includes putting the microcontroller into its lowest-power state and turning off all sensors between measurements. By combining a relatively large battery capacity with a low duty cycle of transmitting every 15 minutes, each node can operate for an extended period before a battery replacement is needed. Using standard alkaline D-cell batteries which can be easily purchased at a hardware store with no need for specialized charge or custom battery packs makes the system easily maintainable in the field. At the end-of-life, batteries and electronics should be disposed of in accordance with local regulations and the guidance of the waste disposal company servicing the water utility. For alkaline batteries, it is recommended to check with local or state solid waste authority for proper recycling [29].

The sensor PCB is specifically designed to interface with a DFRobot turbidity sensor, DFRobot turbidity sensor, and a PT100 RTD for temperature sensing, so the sensing side is not highly modular. However, the design provides a fourth, unpopulated sensor input, meaning that one additional sensor could be integrated with the design without any further hardware changes. Since the design of the analog frontend of the sensor node is unified for all sensor inputs, further revisions of the design requiring more sensor inputs could be easily implemented by simply increasing the width of the analog multiplexing scheme.

2.1.4. Customer Requirements

The marketing requirements for this project were established based on data collected from a survey distributed to three water authorities, resulting in a total of seven responses from water quality technicians. The survey itself can be found in Appendix A. Additionally, Dr. Yen-Chih Chen, an associate professor of environmental engineering at Penn State Harrisburg, provided expert insights on water quality treatment and industry standards.

Survey results were analyzed by calculating averages and creating graphs to better understand the key requirements for this project. These findings are summarized in Appendix B. Respondents expressed strong interest in a wireless, battery-powered design. Notably, chlorine monitoring was rated as the highest priority. However, due to budget limitations and sensor availability, a chlorine sensor could not be obtained. As a result, turbidity and pH were selected as the two most important parameters to monitor.

Based on the data collected, four primary marketing requirements were identified: system capabilities, system feedback & notifications, ease of use, and deployment flexibility. The weights assigned to these categories, based on their importance, are shown in Table 1. The highest priority requirement is system capabilities, as it defines the core functions of the product, forming the foundation upon which all other features depend. The second most critical requirement is system feedback & notifications, as survey results indicated that technicians require continuous visibility into water quality levels. From this, deployment flexibility was ranked as the third priority, with technicians expressing interest in a versatile, battery-powered solution that offers long operational life. Finally, while ease of use remains important, it was determined to have the lowest relative priority, as it was considered less critical than the other primary requirements.

Table 1: Customer Primary Requirements

Category	System Capabilities	System Feedback & Notifications	Ease of Use	Deployment Flexibility	Total	Weight
System Capabilities	1	4/3	4/1	2/1	8.33	0.41
System Feedback & Notifications	3/4	1	3/1	3/2	6.25	0.31
Ease of Use	1/4	1/3	1	1	2.58	0.13
Deployment Flexibility	1/2	2/3	1	1	3.17	0.16
				Total	20.33	1.00

The system capabilities category consists of the following subcategories: turbidity monitoring, pH monitoring, wireless communication, AI analytics & predictions, and PLC integration, as displayed in Table 2. Given that the primary function of the product is water quality monitoring, turbidity and pH sensing were identified as the highest-priority features. Turbidity monitoring was ranked the most critical, as it serves as a federally recognized standard for determining water quality. Following this, wireless communication ranked as the next priority, with survey results demonstrating strong technician interest in long-range wireless communication, such as LoRa, that does not rely on Wi-Fi or cellular networks. The lowest-priority subcategories were PLC integration and AI analytics, with technicians showing minimal interest in AI-driven predictions, making it the least essential function of the system.

Table 2: System Capabilities Subcategory Breakdown

System Capabilities	Turbidity Monitoring	pH Monitoring	Wireless Communication	AI Analytics & Predictions	PLC Integration	Total	Weight
Turbidity Monitoring	1	5/4	5/3	5	5/2	11.42	0.34
pH Monitoring	4/5	1	4/3	4	2	9.13	0.27
Wireless Communication	3/5	3/4	1	3	3/2	6.85	0.20
AI Analytics & Predictions	1/5	1/4	1/3	1	1	2.78	0.08
PLC Integration	2/5	1/2	2/3	1	1	3.57	0.11
Total						33.75	1.00

The system feedback & notifications category is composed of dashboard display, alerts, and battery life indicator, as shown in Table 3. Among these, dashboard display was ranked as the highest priority, followed by alerts. These features are essential to technicians' workflow, as they must be able to visually assess water quality levels in real time and receive instant alerts if readings fall below critical thresholds. The battery life indicator was deemed the lowest priority, as it is not a strong need for the feature.

Table 3: System Feedback & Notifications Subcategory Breakdown

System Feedback & Notifications	Dashboard Display	Alerts	Battery Life Indicator	Total	Weight
Dashboard Display	1	5/4	3	5.25	0.48
Alerts	4/5	1/1	2	3.80	0.35
Battery Life Indicator	1/3	1/2	1	1.83	0.17
			Total	10.88	1.00

The ease-of-use category includes low maintenance, user interface clarity, and field replaceable components, as presented in Table 4. Technicians expressed a strong preference for a low-maintenance product, making low maintenance the highest priority. Closely related to this, field replaceable components, including easily swappable batteries and sensors, were also deemed essential for reducing downtime and simplifying maintenance. User interface clarity, while still relevant, was ranked as the lowest priority in this category.

Table 4: Ease of Use Subcategory Breakdown

Ease of Use	Field Replaceable Components	User Interface Clarity	Low Maintenance	Total	Weight
Low Maintenance	1	8/5	4/3	3.93	0.42
User Interface Clarity	5/8	1	2/3	2.29	0.24
Field Replaceable Components	3/4	3/2	1	3.25	0.34
			Total	9.48	1.00

The deployment flexibility category includes long battery life, scalability for expansion, and environmental durability, as shown in Table 5. Survey results indicated that technicians prefer a battery-powered product, with a desired operational lifespan of at least one to three years before requiring battery replacement. Resulting in long battery life being ranked as the highest priority. Scalability for expansion, allowing for the addition of future sensors, was identified as the second priority. Environmental durability was ranked as the

lowest priority, as the product will most likely be placed indoors, reducing the need for extreme weather resistance.

Table 5: Deployment Flexibility Subcategory Breakdown

Deployment Flexibility	Long Battery Life	Scalability for Expansion	Environmental Durability	Total	Weight
Long Battery Life	1	3/2	3	5.50	0.50
Scalability for Expansion	2/3	1	2	3.67	0.33
Environmental Durability	1/3	1/2	1	1.83	0.17
Total			Total	11.00	1.00

Overall, the survey responses and follow-up analysis allowed us to separate the features that must be present in the first version of the system from those that can reasonably be deferred as enhancements. The required capabilities are turbidity and pH monitoring, wireless communication between nodes and the gateway, a dashboard display with alerting, low maintenance, long battery life, and scalability to support additional nodes or sensors. Together, these define the core functionality that technicians expect from the system. The desired features include AI-based analytics and predictions, PLC integration with existing SCADA infrastructure, a battery-life indicator, improved user-interface clarity, field-replaceable components, and enhanced environmental durability. These items add value and align with technician preferences.

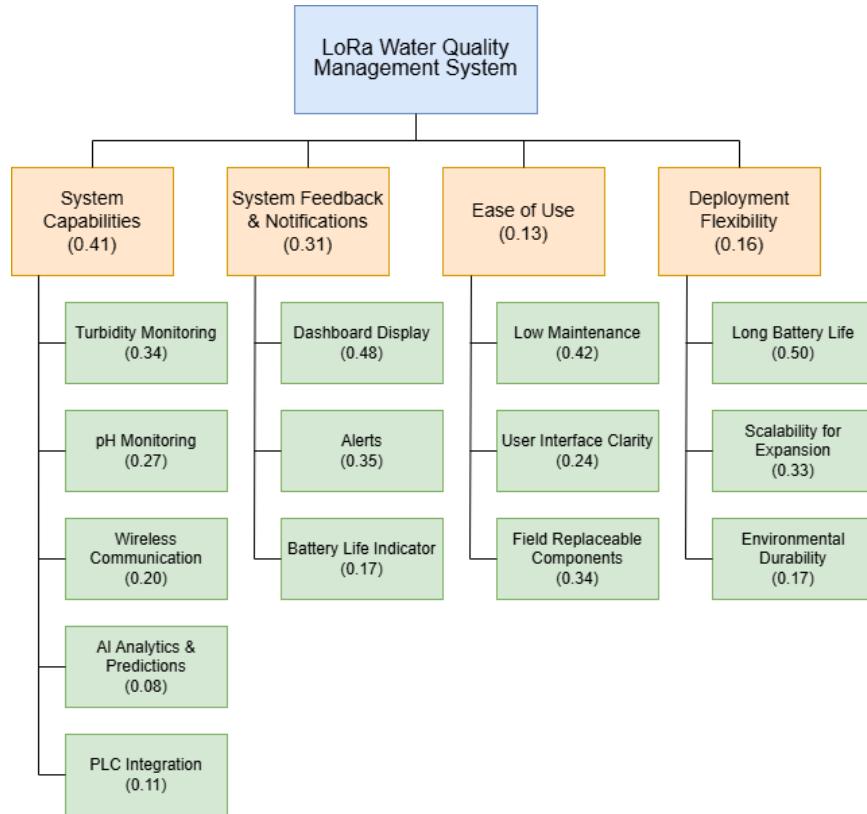


Figure 6: Hierarchical Representation of Marketing Requirements

2.1.5. Constraints

The design of the LoRa Water Quality Management System must meet several key constraints to ensure regulatory compliance, efficient operation, and seamless integration with existing infrastructure. These constraints define the boundaries for the design decision of the system. When defining design constraints, we consider ideal user experience, reliability, and regulatory compliance. Below is a list of constraints that are vital to the operation and functionality of the technology:

- Project must be defined in the first 15-week semester and built/tested in the second 15-week semester
- Team will design a signal conditioner from scratch to satisfy capstone course requirement
- Budget is limited to 500 USD, as allocated by the capstone course
- Low-power radio must operate in a free, unlicensed frequency band

- Sensor nodes must be battery powered with a desired service life of roughly three to five years
- Gateway must function as a stand-alone monitor and data logger, without modifying existing plant systems

2.1.6. Standards

Public water systems must adhere to strict regulatory standards to ensure the safety and quality of drinking water. The Environmental Protection Agency (EPA) has established the National Primary Drinking Water Regulations, commonly referred to as the primary standard, which set these regulatory guidelines. The primary standard specifies the approved methodologies for conducting pH and turbidity analysis. The permitted methodologies for these analyses and their relevant standards are outlined in Table 6 [30]. Additionally, the state of Pennsylvania has a MCL for turbidity that supersedes the EPA MCL standard. It requires that, the turbidity not exceed 0.3 NTUs in 95% of the monthly samples for public water systems with conventional or direct filtration, 1.0 NTUs for slow sand filtration or diatomaceous earth, and 0.15 NTUs for membrane filtration [31].

Table 6: Approved Methodologies and Relevant Standards for pH and Turbidity Analysis

Contaminant	Methodology	EPA Standard	ASTM Standard	SM (18th, 19th ed.)	SM (20th ed.)	SM Online
pH	Electrometric	150.1, 150.2 ¹	D1293-95, 99	4500-H ⁺ B	4500-H ⁺ B	4500-H ⁺ B-00
Turbidity	Nephelometric Method	180.1	-	2130 B	-	-

In addition to the primary standard, the EPA has established the National Secondary Drinking Water Standards, which define Secondary Maximum Contaminant Levels (SMCLs). These secondary standards are non-mandatory but help ensure better color, odor, and taste of drinking water. Specifically, the secondary standards recommend maintaining a pH range between 6.5 and 8.5 [32].

Furthermore, the standard establishes record maintenance requirements. Turbidity analysis records must be retained for at least five years. Additionally, specific data must be

collected and documented, including the date, location, and time of sampling, along with the analysis results [30].

From an RF perspective, key standards for the LoRa Water Quality Management System include the Code of Federal Regulations (CFR) Title 47 Part 15, which governs the operation of wireless devices within the 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz frequency bands. Specifically, the standard enshrines the 902-928 MHz band as an unlicensed frequency band and specifies power output limits for transmitters in this frequency range. Most importantly, subsection 247 specifies time-of-occupancy limits for frequency hopping systems on this band: devices using a transmission bandwidth of less than 250kHz must not occupy one single frequency for more than 400ms within a 20 second period, while devices using more than 250kHz of transmission bandwidth must not occupy one single frequency for more than 400ms within a 10 second period [28].

Another relevant standard governing the wireless communications of the sensor node is IEEE 1528.7, which is used to determine if further Specific Absorption Rate (SAR) testing is required for wireless IoT devices based on power output. Most importantly for this project, devices with less than 20mW of time-averaged power output do not require comprehensive SAR testing under this standard [33]. Thus, to avoid complex and expensive SAR testing to comply with IEEE 1528.7, the system's RF interface must be configured in software to produce a time-averaged power output of less than 20mW.

2.2. Engineering Requirements

The engineering requirements for the LoRa Water Quality Management System were developed from the customer requirements, constraints, and standards found in previous sections of this proposal. The engineering requirements for the project are listed in Table 7 below. Each row of the center column of the table lists one of the engineering requirements, with the corresponding marketing requirements denoted in the left column, and a justification in the right column.

Table 7: Engineering Requirements for the LoRa Water Quality Management System

Marketing Requirements	Engineering Requirements	Justification
10, 12, 13, 23	Uses readily commercially available batter(ies) to produce 3.3V and 5V DC input power.	Provides easy replacement, wide availability, and a voltage level compatible with a wide range of microcontrollers and sensor equipment.
4, 13, 14, 16, 17, 23, 25	Should utilize wireless communication technology with a transmit current consumption of under 150mA, and under 5mA idle current consumption, on the US915 (902-928 MHz) frequency band.	Minimizes power consumption, operates within legally permissible frequencies (902-928 MHz in the United States and Canada), and does not rely on cellular or Wi-Fi.
1, 2, 3, 19	Each sensor node must be capable of producing real-time pH data within ± 0.3 of actual, and turbidity within 0.1NTU of actual under 1.0 NTU, and temperature within ± 0.35 °C of actual.	Turbidity and pH are regulated by Pennsylvania drinking-water standards, so these tolerances support legal compliance. Temperature, while not part of the MCL, was identified by technicians as an important parameter. Together, these three measurements provide both regulatory confidence and practical insight into overall water-quality conditions.
5, 7, 8, 13, 19, 24	System must be able to process data with backend software at minimum every 15 minutes, and alert technicians when either the turbidity or pH is within 15% of thresholds set by Pennsylvania, which are 1.0 NTU and between 6.5-8.5, respectively, or those set by the Water Utility.	Processing the real-time data and providing an overview of the system with some analytics assists technicians and engineers with making determinations regarding the water quality of the system. Processing this data every 15 minutes keeps the system consistent with Pennsylvania Regulations. To help utilities maintain compliance, the system should alert technicians when their water is within 15% of the Pennsylvania Turbidity and pH limits, which are 1.0 NTU maximum, and 6.5-8.5, respectively.

6, 10, 11, 14, 18, 24	System should be easily integrated with existing utility infrastructure and minimize the impact on existing operations.	This makes the system more desirable for utilities and more useful to technicians already familiar with existing plant systems.
1, 2, 3, 14, 21, 22	A custom signal conditioning subsystem shall filter, offset, and amplifier sensor outputs before being presented to the analog to digital converter.	The sensors produce small or bipolar voltages that cannot be read directly by an analog to digital converter. The custom signal conditioner shifts and scales the signals into a usable range while minimizing added error, allowing sensors to still meet water quality accuracy required.

Customer Requirements

1. Turbidity Monitoring [R]
2. pH Monitoring [R]
3. Temperature Monitoring [R]
4. Wireless Communication [R]
5. AI Analytics & Predictions [D]
6. PLC Integration [D]
7. Dashboard Display [R]
8. Alerts [R]
9. Battery Life Indicator [D]
10. Low Maintenance [R]
11. User Interface Clarity [D]
12. Field Replaceable Components [D]
13. Long Battery Life [R]
14. Scalability for Expansion [R]
15. Environmental Durability [D]

Standards

16. Code of Federal Regulations (CFR) Title 47, Part 15.247: Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz
17. IEEE Std 1528.7 - Guide for EMF Exposure Assessment of Internet of Things (IoT) Technologies and Devices
18. USB 2.0 – Universal Serial Bus Specification, Rev. 2.0
19. Pennsylvania Code, Title 25, Chapter 109: Maximum Contaminant Level (MCL)

Constraints

20. Project must be defined in the first 15-week semester and built/tested in the second 15-week semester
21. Team will design a signal conditioner from scratch to satisfy capstone course requirement

- 22.** Budget is limited to 500 USD, as allocated by the capstone course
- 23.** Sensor nodes must be battery powered with a desired service life of roughly three to five years
- 24.** Gateway must function as a stand-alone monitor and data logger, without modifying existing plant systems
- 25.** Low-power radio must operate in a free, unlicensed frequency band

2.3. Level 1 and 2 Functional Decomposition

The Level 1 and Level 2 decompositions are shown in this section. The Level 1 decomposition shows the two modules of the LoRa Water Quality Management System: the sensor node and the central gateway. The Level 2 decomposition is the detailed design level, where circuit and software components are shown.

2.3.1. Level 1 Decomposition

The Level 1 decomposition of the LoRa Water Quality Management System is shown in Figure 7 below. The system is comprised of two main functional blocks: the Sensor Node and the Central Gateway. These two functional blocks are connected via a wireless communication link, shown with a dashed arrow between both wireless transceiver blocks. Power to the system is supplied by an internal battery in each sensor node and externally at the central gateway of the system. Each sensor node has a direct connection to the water and outside environment to take measurements. Additionally, the Central Gateway is connected to Wi-Fi. From there, the sensor node data is processed and analyzed to make decisions about the overall health of the system. The Central Gateway then provides decision making logic and generates alerts and a dashboard that displays water quality metrics for the use of technicians and engineers.

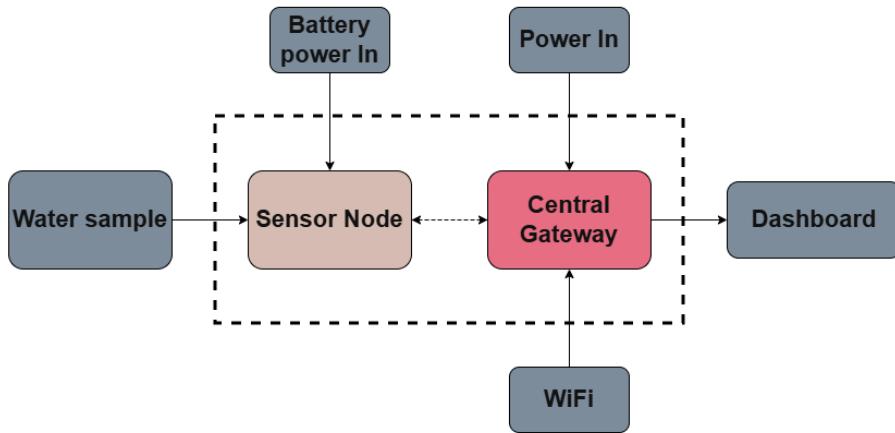


Figure 7: Level 1 Decomposition of the LoRa Water Quality Management System

2.3.2. Level 2 Decomposition

The level 2 decomposition of the LoRa Water Quality Management System, Figure 8, shows the detailed view of the Sensor Node and the Central Gateway. Each sensor node, which is depicted in the left subsystem of the diagram, is responsible for collecting the raw data from the pH, turbidity, and temperature sensors, interpreting the sensor output (e.g. analog voltage value) as a measurement, and formatting the data to be sent to the central gateway over the wireless communication link. Once the interpreted and formatted data is received by the central gateway, it is further formatted so that it can be analyzed through an Artificial Neural Network, compared against thresholds, and be reported to an external dashboard to be used by Technicians and Engineers. Based on the analyzed data output, decision making logic is used to raise alerts to notify technicians and engineers of issues in the system, or post warnings and other valuable information to the system dashboard.

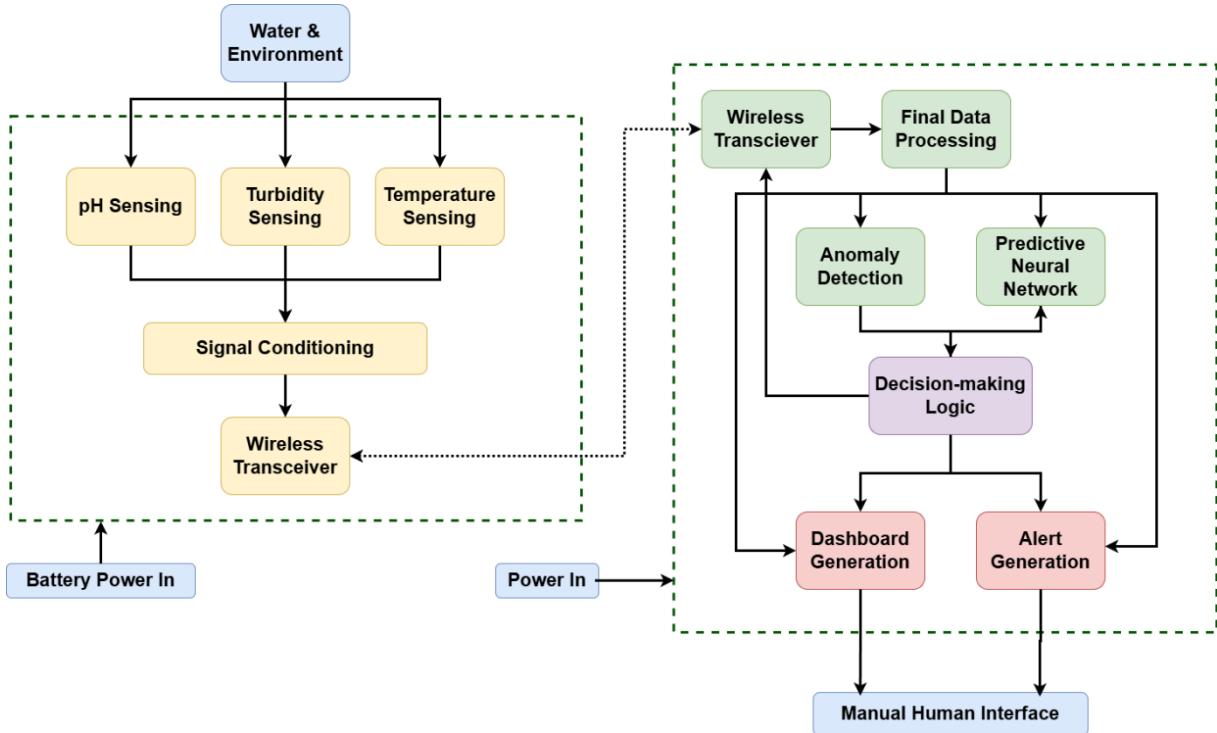


Figure 8: Level 2 Decomposition of the LoRa Water Quality Management System

Table 8: pH Sensor Function Table

Module	pH Sensor
Inputs	<ul style="list-style-type: none"> Raw water sample pH
Outputs	<ul style="list-style-type: none"> Analog voltage corresponding to the pH of the water
Functionality	<ul style="list-style-type: none"> Converts the pH of the water to an analog voltage value interpretable by a microcontroller

Table 9: Turbidity Sensor Function Table

Module	Turbidity Sensor
Inputs	<ul style="list-style-type: none"> Raw water sample Turbidity
Outputs	<ul style="list-style-type: none"> Analog voltage corresponding to the Turbidity of the water
Functionality	<ul style="list-style-type: none"> Converts the Turbidity of the water to an analog voltage value interpretable by a microcontroller

Table 10: Temperature Sensor Function Table

Module	Temperature Sensor
Inputs	<ul style="list-style-type: none"> Raw water sample temperature
Outputs	<ul style="list-style-type: none"> Analog voltage corresponding to the temperature of the water
Functionality	<ul style="list-style-type: none"> Converts the temperature of the water to an analog voltage value interpretable by a microcontroller

Table 11: Signal Conditioning Function Table

Module	Signal Conditioning
Inputs	<ul style="list-style-type: none"> Analog pH data Analog turbidity data Analog temperature data
Outputs	<ul style="list-style-type: none"> Formatted digital data for the pH, turbidity, and temperature
Functionality	<ul style="list-style-type: none"> Interprets the analog voltages produced by the sensors as digital data Converts the digitized analog voltage readings to relevant measurement units for pH, turbidity, and temperature Formats the data to be sent to the central gateway module

Table 12: Data Collection Wireless Transceiver Function Table

Module	Wireless Transceiver (Data Collection)
Inputs	<ul style="list-style-type: none"> Formatted digital pH, turbidity, and temperature readings
Outputs	<ul style="list-style-type: none"> Wireless signal to be sent to central gateway wireless transceiver
Functionality	<ul style="list-style-type: none"> Broadcasts the formatted sensor data wirelessly to be read in by the central gateway

Table 13: Data Processing Wireless Transceiver Function Table

Module	Wireless Transceiver (Data Processing)
Inputs	<ul style="list-style-type: none"> Wireless signal from sensor node containing formatted digitized sensor data Inputs from decision making logic to mediate data transfer
Outputs	<ul style="list-style-type: none"> Formatted, digitized sensor data ready for final processing Acknowledge signal to sensor node transceiver indicating successful data transmission
Functionality	<ul style="list-style-type: none"> Interprets the broadcasted wireless data into useable data for further data processing

Table 14: Final Data Processing Function Table

Module	Final Data Processing
Inputs	<ul style="list-style-type: none"> Formatted, digitized sensor data originating from sensor nodes
Outputs	<ul style="list-style-type: none"> A .CSV file containing updated sensor data from all nodes
Functionality	<ul style="list-style-type: none"> Combines and formats data from all sensor nodes into a commonly used file type, which can be interpreted by the dashboard generation software, neural network, and anomaly detector

Table 15: Anomaly Detection Function Table

Module	Anomaly Detection
Inputs	<ul style="list-style-type: none"> CSV data with updated sensor readings from all sensor nodes
Outputs	<ul style="list-style-type: none"> A simplified report of anomalous sensor readings and readings outside the thresholds set by the utility and legal regulations
Functionality	<ul style="list-style-type: none"> Looks through incoming sensor data to identify readings that are outside of legal compliance and compliance with utility specifications

Table 16: Predictive Neural Network Function Table

Module	Predictive Neural Network
Inputs	<ul style="list-style-type: none"> CSV data with updated sensor readings from all sensor nodes Signal to cause the neural net to begin running from decision-making logic
Outputs	<ul style="list-style-type: none"> Predictions for future system water quality based on training data
Functionality	<ul style="list-style-type: none"> Makes predictions for future temperature, pH, and turbidity of the system based on incoming and past sensor data

Table 17: Decision-making Logic Function Table

Module	Decision-making Logic
Inputs	<ul style="list-style-type: none"> Anomalous sensor readings Sensor readings outside legal or utility-set thresholds System water quality predictions from Artificial Neural Network
Outputs	<ul style="list-style-type: none"> Selected anomalous readings to report on the dashboard Readings and predictions requiring attention from technicians and engineers Wireless data reception mediation signals
Functionality	<ul style="list-style-type: none"> Takes in all available system data and makes determinations on what requires attention from engineers and technicians, and the severity of system-wide issues

Table 18: Dashboard Generation Function Table

Module	Dashboard Generation
Inputs	<ul style="list-style-type: none"> CSV data with updated sensor readings from all sensor nodes Sensor readings and system water quality predictions of note as determined by decision-making logic
Outputs	<ul style="list-style-type: none"> A visual display of current system-wide data, anomalous readings of note, and future system water quality predictions
Functionality	<ul style="list-style-type: none"> Summarizes system-wide data in a visual format easily readable by technicians and engineers

Table 19: Alert Generation Function Table

Module	Alert Generation
Inputs	<ul style="list-style-type: none"> Sensor readings falling outside legal or utility-set thresholds Significant anomalous sensor readings Water quality predictions of concern as determined by decision-making logic
Outputs	<ul style="list-style-type: none"> Automated email sent to technicians and engineers of concern
Functionality	<ul style="list-style-type: none"> Informs technicians and engineers immediately when issues arise in the system, or when issues are predicted to arise, so that proactive corrective action may be taken

2.4. Subsystem Development

Throughout the design and development phase of the LoRa Water Quality Management System, numerous design decisions and trade-offs were made to maximize the system's performance according to the engineering and customer requirements as outlined above. This section details the design and implementation of the numerous subsystems and components employed for use with the LoRa Water Quality Management System, including the Signal Conditioner Subsystem, which was designed "from scratch" to meet the constraints set forth by the Senior Electrical Capstone Design Course.

2.4.1. Signal Conditioner Circuit Design <Matthew Sharp>

The signal conditioner subsystem selects a sensor input, then amplifies and offsets the signal as needed to prepare it for analog-to-digital conversion. This subsystem is divided into an analog frontend, which conditions the incoming analog signal using a fully-parameterizable software-defined instrumentation amplifier, and a digital post-processing layer, where the resulting analog voltage is converted into a meaningful measurement value. This subsystem is comprised of the Texas Instruments TMUX1309 dual 4:1 analog multiplexer [34], which selects between differential sensor inputs, providing a differential input for use with the instrumentation amplifier, while also allowing all sensor inputs to the device to be used with the same analog circuitry, increasing modularity of the design while reducing layout complexity. A fully-customizable DC offset is then applied to the positive end of the signal using a summing amplifier and digital potentiometers to produce the DC levels. The output of the summing amplifier is cascaded with an instrumentation amplifier, which is comprised of the Texas Instruments TLV9144 4-channel rail-to-rail input and output operational amplifier [35]. The input DC offset levels, gain, and output DC offset are all controlled by Microchip Technology MCP4651 dual $50k\Omega$ digital potentiometers [36]. The output of the instrumentation amplifier is then read by the Microchip Technology MCP3425 16-bit Sigma-Delta Analog-to-Digital Converter [37]. A block diagram of this subsystem is shown in Figure 9 below.

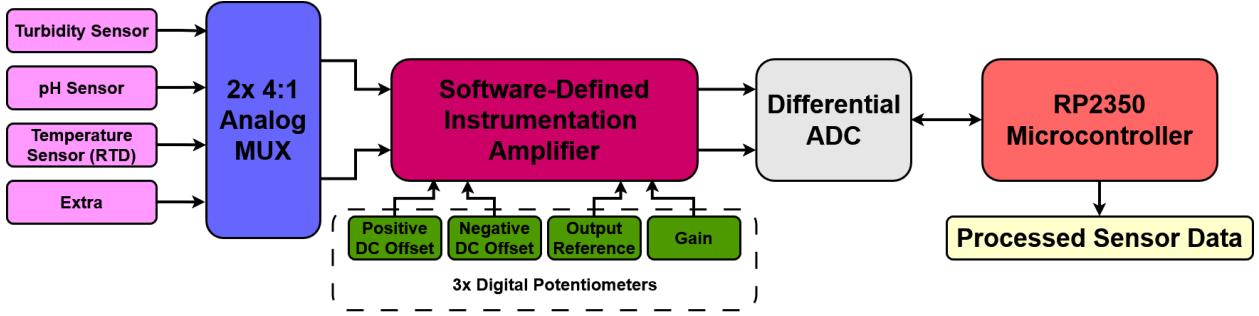


Figure 9: Block Diagram of Signal Conditioner Subsystem

Figure 10 shows the multiplexer, a symbol representing the software-defined instrumentation amplifier/offset circuit, and the ADC. A single sensor (either temperature, pH, or turbidity) is first selected through the multiplexer. The chosen sensor's output is fed into the amplification section, then routed to the ADC for conversion and subsequent processing by the microcontroller.

Figure 11, Figure 12, and Figure 13 illustrate the internal structure of the amplification section. Figure 11 presents the first stage: a summing op-amp that applies either a positive or negative DC voltage offset to the signal. This stage also includes a passive first-order RC low-pass filter to remove high-frequency noise. The conditioned signal is then passed to the second stage, shown in Figure 12, which is an instrumentation amplifier providing any required gain.

Both the summing and instrumentation amplifiers are controlled by three digital potentiometers, shown in Figure 13. The left potentiometer adjusts the DC offset in stage one, the right potentiometer sets the reference voltage for the instrumentation amplifier, and the middle potentiometer determines the gain in stage two. All three potentiometers are managed by the microcontroller, making this a fully software-controlled signal conditioner.

To account for nonlinearities and resistance deviations between digital potentiometers discovered during initial testing, an automatic calibration routine was incorporated into the sensor node's firmware. The calibration measures the DC offsets and Gain Resistor values at each potentiometer wiper step location (0-256). This data is then stored on the NOR flash IC on the PCB, which the microcontroller then reads into memory, using it to

accurately set the digital potentiometers according to desired gain and DC offset values.

Following from the engineering requirements of the signal conditioner subsystem:

- Interprets the analog voltages produced by the sensors as digital data
- Converts the analog voltage readings to units for pH, turbidity, and temperature
- Formats the data to be sent to the central gateway module

Accurate and reliable operation of the software-defined instrumentation amplifier, which pre-processes incoming analog sensor voltages to be compatible with the ADC and provide increased resolution in the measurement region of interest, are necessary for full operation of this subsystem. Since the calibration data of the software-defined instrumentation amplifier exposes known resistance and DC offset values with respect to the potentiometer steps, the sensor voltage at the input of the amplifier can be easily determined following from the output voltage equation of the amplifier, given in Equation 2 below. This gives an input voltage equation as seen in Equation 3 below, which the microcontroller uses with the known resistance and DC offset values to compute the original sensor voltage.

Equation 2: Output Voltage of the Software-Defined Instrumentation Amplifier

$$V_{out} = (V_{in} + V_{off}^+ - V_{off}^-) \left(\frac{2 * R_0}{R_G} + 1 \right) + V_{REF}$$

Equation 3: Input Voltage of the Software-Defined Instrumentation Amplifier

$$V_{in} = \frac{V_{out} - V_{REF}}{\frac{2R_0}{R_G} + 1} - V_{off}^+ + V_{off}^-$$

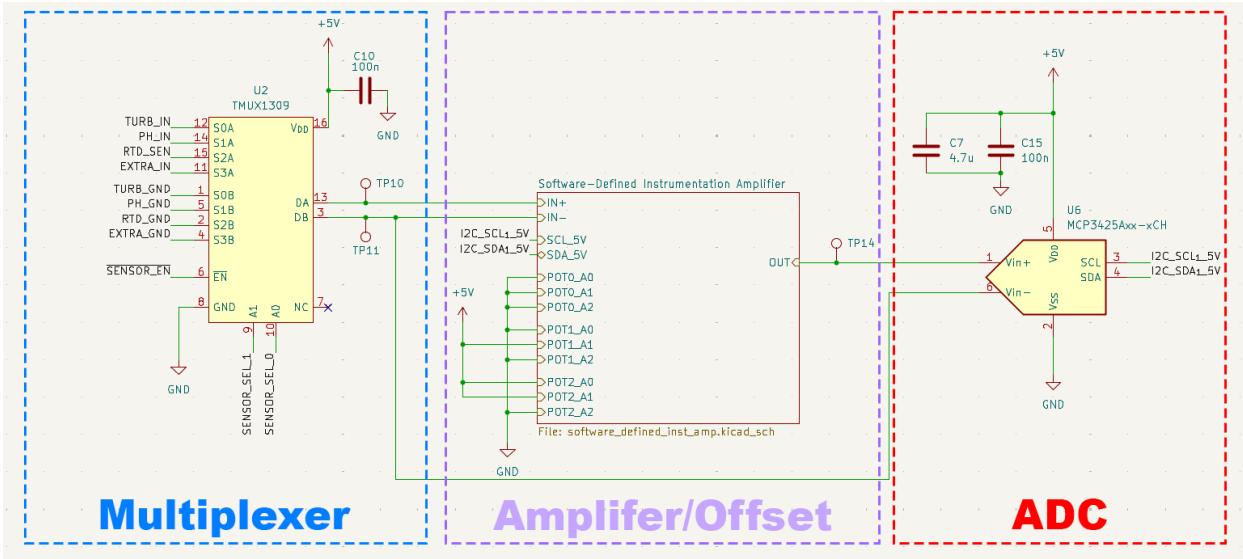


Figure 10: Signal Conditioner Circuit

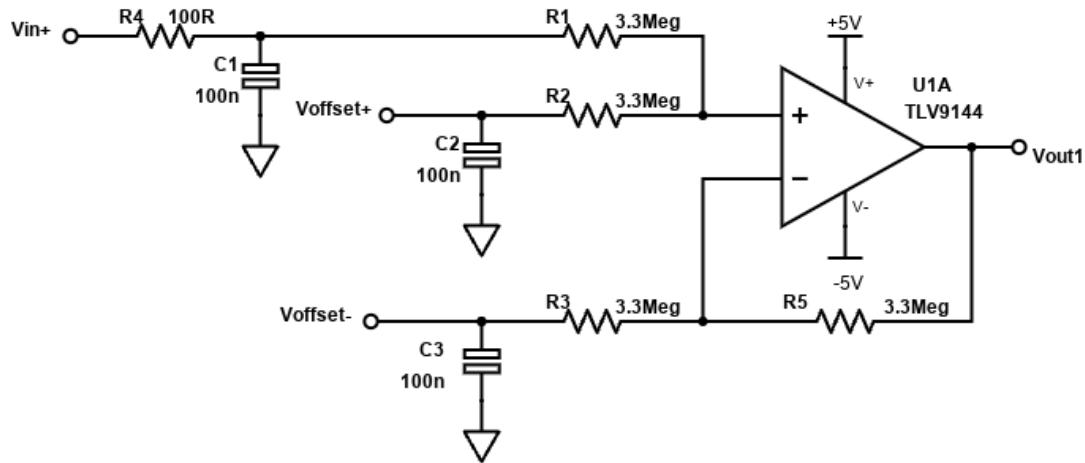


Figure 11: Stage 1 of Software-Defined Instrumentation Amplifier

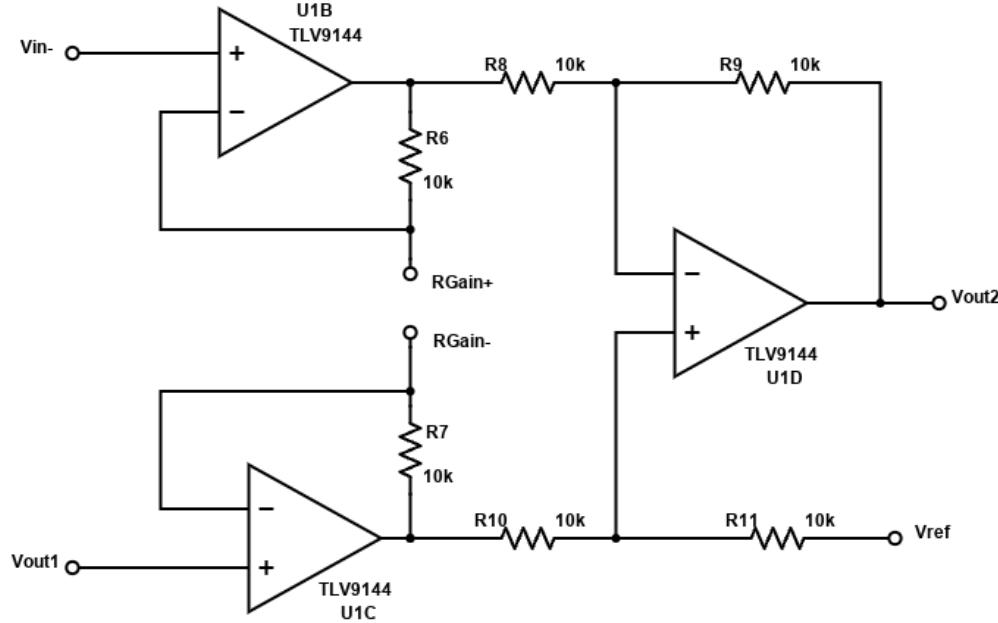


Figure 12: Stage 2 of Software-Defined Instrumentation Amplifier

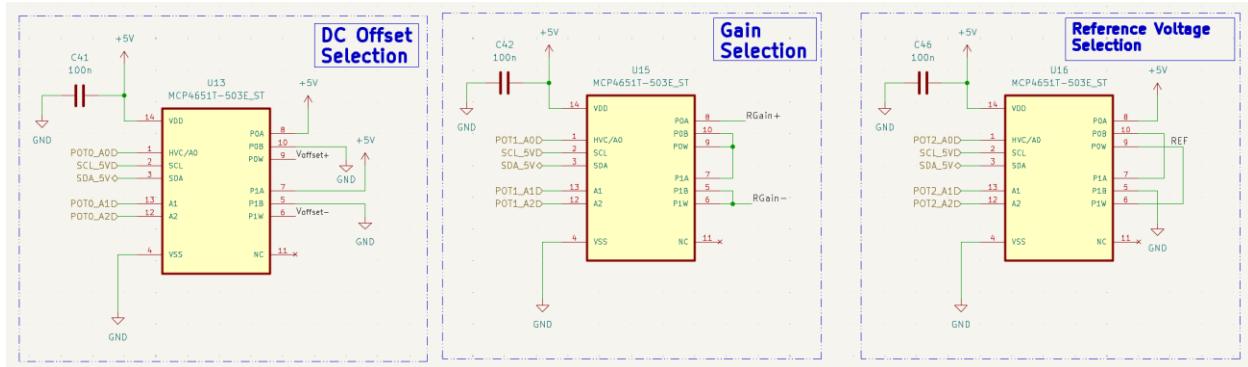


Figure 13: Digital Potentiometers for Software-Defined Instrumentation Amplifier

2.4.2. Signal Conditioner Theory

The main section that requires calculations is the software-defined instrumentation amplifier. The input signal for this circuit originates from the positive output of a sensor, which provides a varying voltage corresponding to the measured quantity. Beginning with stage one as shown in Figure 10, this input signal is first passed through a passive first-

order RC low-pass filter. The cut-off frequency for this type of filter is given by Equation 4 below.

Equation 4: Cutoff Frequency of a First-Order Passive Low Pass Filter

$$f_{cutoff} = \frac{1}{2\pi RC}$$

Using this equation, R and C were chosen to be 100Ω and 100nF , respectively. This results in a cutoff frequency of approximately 16kHz , making it an effective filter for attenuating high-frequency noise while allowing the desired signal to pass through.

Next, the summing amplifier was configured. All resistors in this stage were set to $10\text{k}\Omega$ resulting in unity gain so that the amplifier introduces no additional amplification. With V_{in} as the sensor input, $V_{offset+}$ as the positive DC offset, and $V_{offset-}$ as the negative DC offset, the stage-one output (V_{out1}) can be expressed using Equation 5 below.

Equation 5: Software-Defined Instrumentation Amplifier First Stage Output Voltage

$$V_{out1} = V_{in}^+ + V_{offset+} - V_{offset-}$$

The digital potentiometers used in Figure 12 each contain two $50\text{k}\Omega$ potentiometers per IC. The leftmost digital potentiometer is configured such that one wiper is connected to $V_{offset+}$ and the other to $V_{offset-}$. This arrangement forms a voltage divider between the 5V supply and ground, allowing the circuit to apply up to a $\pm 5\text{V}$ DC offset to the input signal.

In the second stage, the signal is processed by an instrumentation amplifier configured with $10\text{k}\Omega$ resistors. This allows the gain to be reduced to a simple closed form and set by a single resistor R_G , as shown in Equation 6 below.

Equation 6: Instrumentation Amplifier Gain

$$Gain = \frac{20\text{k}\Omega}{R_G} + 1$$

The non-inverting input of the instrumentation amplifier receives the stage-one output, the inverting input is tied to the sensor's negative terminal V_{in}^- , and the REF pin is driven by V_{ref} , and the gain is set by the variable resistor R_{gain} . The resulting stage-two (V_{out2}) output is given in Equation 2 from the previous section.

Here, R_G is determined by the middle digital potentiometer shown in Figure 12. The two potentiometers are cascaded to form a single $100\text{k}\Omega$ variable resistor. This allows for a variable gain from 1.2 to 103. The rightmost digital potentiometer in Figure 12 sets the reference voltage for the instrumentation amplifier. The two internal potentiometers' wipers are tied together and used as a voltage divider, providing an adjustable reference voltage between 0 V and 5 V.

In summary, the theory behind the signal conditioner demonstrates how each stage contributes to accurate and adjustable signal conditioning. The first stage filters noise and applies a controllable DC offset, while the second stage amplifies the conditioned signal with a tunable gain and reference voltage. Together, these stages enable precise software control of the signal's amplitude and baseline, ensuring reliable analog-to-digital conversion and compatibility with the microcontroller described earlier.

2.4.3. Signal Conditioner Simulation

The signal conditioner was simulated in PSpice for TI. This tool was selected because it includes device models for Texas Instruments operational amplifiers that closely represent the parts used in the physical circuit. Figure 13 shows the simulated schematic. The objective of the simulation is to evaluate the two stages that provide offset and amplification.

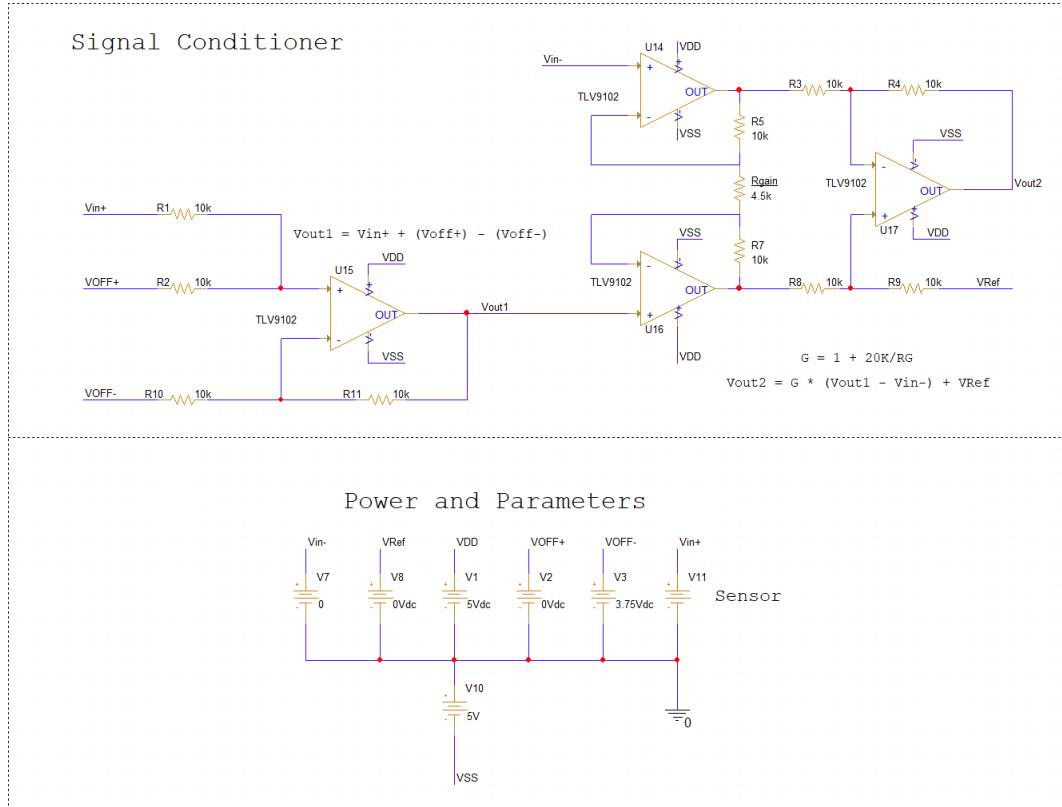


Figure 14: PSpice Simulation Schematic of Software-Defined Instrumentation Amplifier

A DC voltage source emulates the sensor output over the range of voltages expected from the sensors. Additional DC sources were used to set parameters such as voltage offset and reference. These sources represent the digital potentiometers that generate the DC offset and the reference voltage set by the digital potentiometers. A discrete resistor (R_{gain}) sets the instrumentation amplifier gain and represents the digital potentiometer used for gain control. The TLV9102 op-amp model was used because a TLV9144 model is not available, and the TLV9102 provides a reasonable substitute, as it has similar input, output, bandwidth, and CMRR characteristics to the TLV9144, per manufacturer datasheets.

Three simulations were conducted with inputs that are the approximate voltage ranges expected for the temperature, pH, and turbidity sensors. Each case uses a DC sweep of the sensor source so that the x-axis of the plot corresponds to the sensor input voltage.

The first simulation models the temperature sensor. As the temperature sensor used is a PT100 RTD, which is a resistive device, we expect the sensor resistance to range from 100Ω

to 137Ω , corresponding to $0^\circ C$ and $100^\circ C$, respectively. Forming a voltage divider with a $4.7k\Omega$ resistor connected to our 5V input, we expect the sensor voltage to be between 107mV and 143mV. The ADC following these stages is designed for an input of approximately 0V to 2V. The simulation therefore used the parameters: $V_{off+} = 0V$, $V_{off-} = 0V$, $V_{ref} = 0V$, and $R_{gain} = 1.5k\Omega$. The expected output is a swing from 1.498 V to 2.002 V. Figure 14 shows the results. The green trace is the sensor input, the pink trace is the stage-one output, and the blue trace is the stage-two output.

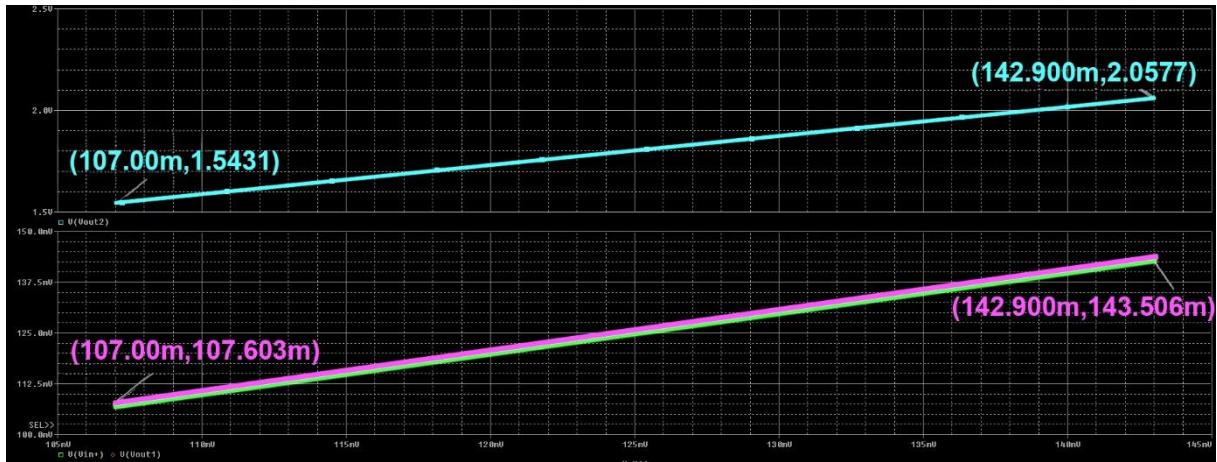


Figure 15: Simulated Temperature Sensor Plot

The stage-one output exhibits a small offset of a few microvolts, which is attributable to non-ideal op-amp behavior. The stage-two output swings from 1.5431V to 2.0577V. This differs from the expected range by about 3%, which is acceptable and indicates correct operation.

The second simulation models the pH sensor. This sensor is connected to a pre-amplifier, and over the full 0 to 14 pH range the preamp output is expected to swing from 1.5V to 3.5V. To obtain an output near 0V to 2V, the following parameters were applied: $V_{off+} = 0V$, $V_{off-} = 1.5V$, $V_{ref} = 0V$, and $R_{gain} = 100 k\Omega$. The expected output is a swing from 0 V to 2.4 V. Figure 15 presents the results. Trace colors match those in the previous simulation.

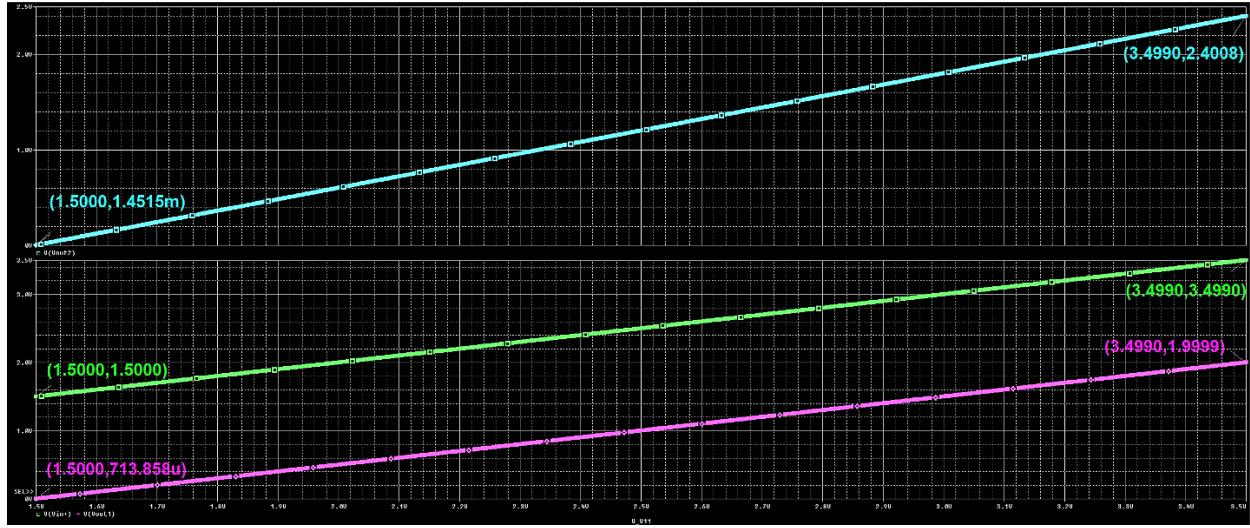


Figure 16: Simulated pH Sensor Plot

Stage one shifts the input downward by approximately 1.5V as intended. The stage-two output swings from 1.4515 mV to 2.4008 V. The difference from the expected range is about 0.14%, which demonstrates strong agreement.

The final simulation models the turbidity sensor. This sensor is expected to produce approximately 3.996V to 4.117V, which corresponds to roughly 0 to 100 NTU at the probe output. To obtain an output near 0V to 2V, the following parameters were used: $V_{off+} = 0V$, $V_{off-} = 3.75V$, $V_{ref} = 0V$, and $R_{gain} = 4.5 k\Omega$. The expected output is a swing from 1.341 V to 2.000 V. Figure 16 shows the results, with the same trace color assignments.

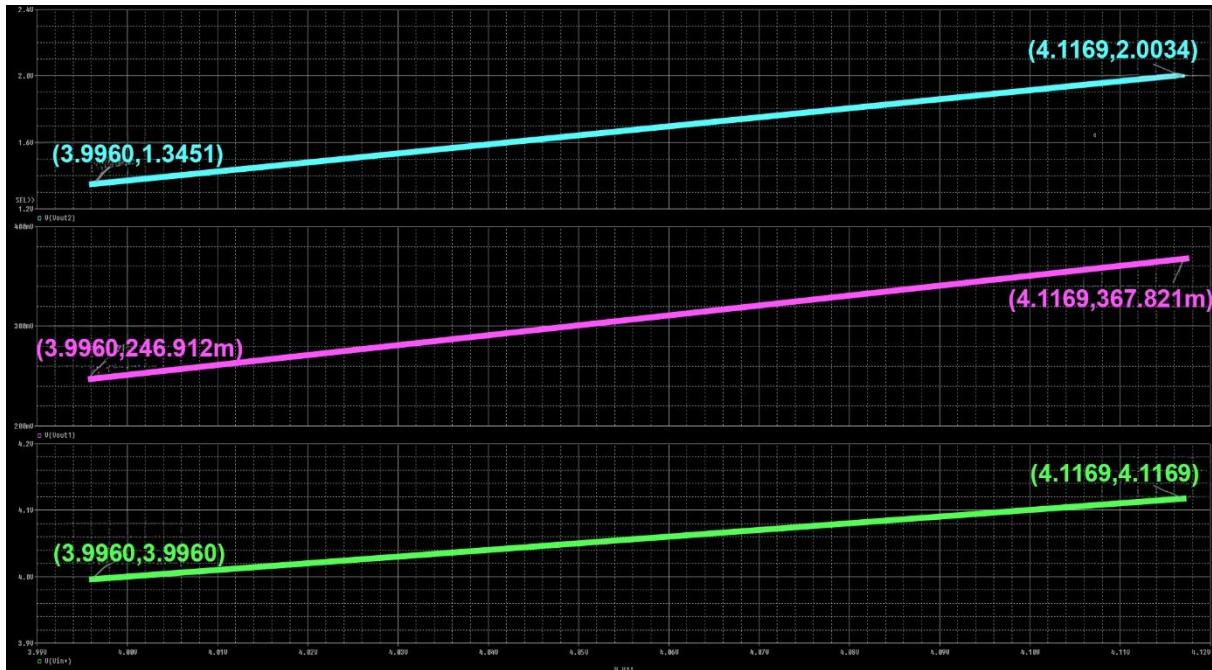


Figure 17: Simulated Turbidity Sensor Plot

Stage one shifts the input downward by approximately 3.75 V. The stage-two output swings from 1.3451 V to 2.0034 V. This differs from the expected range by about 0.3%, which indicates strong performance of the circuit in this configuration.

Overall, the simulations behaved as expected and validated the design. Across the temperature, pH, and turbidity cases, the stage-one offset and stage-two gain produced output ranges within a few percent of the targets, confirming proper operation. These results demonstrate that the signal conditioner's offset and amplification functions work as intended and that the output is correctly conditioned for the downstream ADC. Using the signal conditioner subsystem as simulated here will ensure the signal is within the ADC's input voltage range of -2.048V to +2.048V. The added gain and high CMRR of the instrumentation amplifier will allow for the ADC to detect smaller changes in the input measurements, improving overall performance of the system.

2.4.4. pH Sensor Preamplifier Circuit Design <Cole Hadley>

The purpose of the preamplifier is to increase the pH sensor signal to a level that can be accepted by the signal conditioner and to ensure the signal is non-negative, as the analog multiplexer used for the signal conditioner subsystem is only compatible with positive

analog input voltages. The pH probe produces a differential signal that swings from approximately -500mV to $+500\text{mV}$, where -500mV corresponds to a pH of 14 and $+500\text{mV}$ corresponds to a pH of 0. The design goal of this subsystem is to provide a modest gain to the input signal, apply a DC offset to the input voltage so that it is entirely positive, and to compensate for the glass pH sensor's high output impedance by utilizing an operational amplifier with high input impedance and negligible input bias currents.

The pH sensor used is manufactured by DFRobot. When purchased, it is supplied with a helper interface board intended for direct connection to a microcontroller. To integrate this function into our sensor node PCB and maintain control over component choices, we reverse engineered the DFRobot circuit nearly one to one with targeted adjustments for our requirements. Figure 18 shows the resulting preamplifier schematic. The primary elements are two JFET input operational amplifier stages and a low-pass RC filter.

The signal from the probe first passes through a low-pass RC filter. This preserves the DC signal associated with pH while attenuating higher frequency noise. The next consideration is input loading. The pH probe behaves as a low current, high impedance source. A JFET input op-amp, specifically the TL082, provides buffering with minimal loading, and the non-inverting configuration preserves the ultra-high input impedance at the probe interface. Using the inverting input would present an input resistance that would load the sensor and degrade accuracy.

The first amplifier stage therefore uses a non-inverting configuration to provide gain. The second stage uses an op-amp configuration that inverts and applies a controlled DC offset. The offset is selected so that the combined effect of gain and level shifts the original -500mV to $+500\text{mV}$ sensor range into a strictly positive voltage window appropriate for the signal conditioner input.

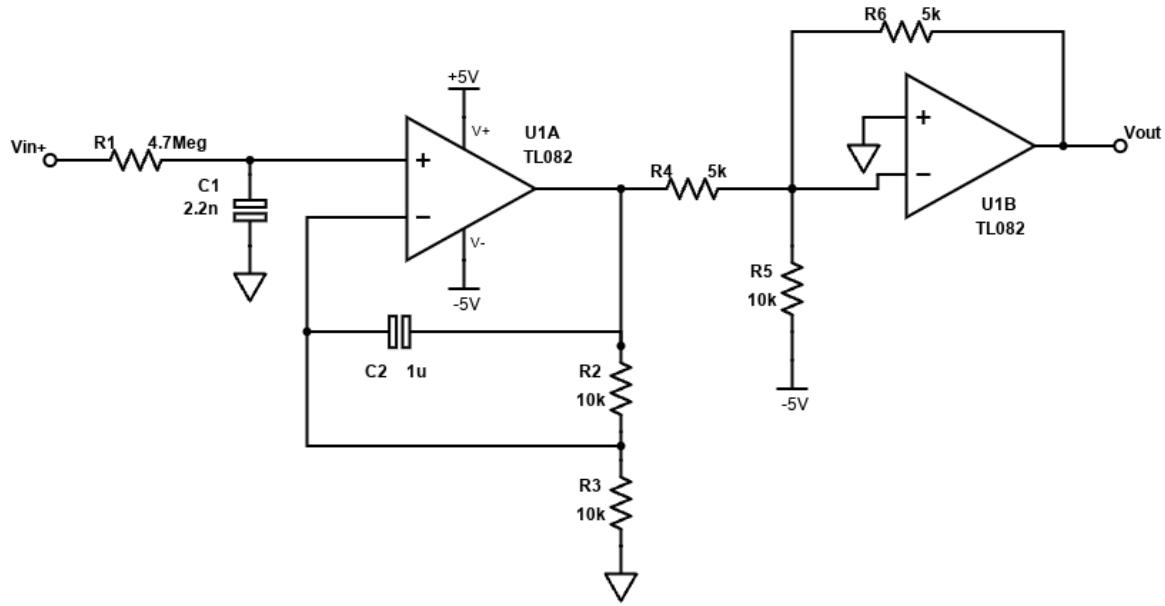


Figure 18: pH Preamplifier Circuit

2.4.5. pH Sensor Preamplifier Theory

The pH sensor produces a differential voltage that varies approximately from -500 mV to $+500$ mV over the full measurable pH range. Because this signal is both low in magnitude and bipolar, it requires amplification and level shifting before being processed by the signal conditioner or ADC. The preamplifier accomplishes this using two stages: a non-inverting amplifier to provide gain and high input impedance buffering of the incoming signal, and an inverting amplifier with a reference offset to shift the signal into a fully positive voltage range.

The pH signal first passes through a passive RC low-pass filter that removes unwanted noise while allowing the slow-varying pH signal to pass. Using Equation 4, the cutoff frequency of this filter is calculated to be 15.4 Hz, which is sufficient to capture the low-frequency nature of pH changes while attenuating high-frequency interference or switching noise.

After filtering, the signal enters the first op-amp stage configured as a non-inverting amplifier. Since the output impedance of the pH sensor used is very high, in the range of

$100M\Omega$ or more, it is essential that the input of the operational amplifier has a significantly larger impedance. The IC chosen is the Texas Instruments TL082H, which has a common-mode input impedance of $6T\Omega$ and input bias currents of $1pA$, according to its datasheet. This ensures that the input signal is not loaded down by the input stage of the pre-amplifier, and the sensor voltage is propagated to the signal conditioner. The closed-loop gain of this stage is given in Equation 7 below.

Equation 7: pH Sensor Pre-Amplifier Gain

$$Gain = 1 + \frac{R_2}{R_3}$$

With $R_2 = 10 k\Omega$ and $R_3 = 10 k\Omega$, the stage gain is 2. This doubles the $\pm 500mV$ input signal to $\pm 1 V$.

The output of the first stage is cascaded with another stage, which both inverts and applies a DC offset to the signal to eliminate its negative component. This configuration uses resistors R_4 , R_5 , and R_6 as shown in Figure 17. The general expression for the output of this stage is Equation 8 below.

Equation 8: First Stage Output Voltage of the pH Sensor Pre-Amplifier

$$V_{out1} = \frac{-R_6}{R_4} * V_{in} - \frac{R_6}{R_5} * V_{ref}$$

V_{in} is the input signal from the first stage, R_6 is the feedback resistor, R_4 is the input resistor, and V_{ref} is the reference voltage applied through R_5 . For this design, $V_{in} = 2V_{in+}$, $R_6 = 5 k\Omega$, $R_4 = 5 k\Omega$, $R_5 = 10 k\Omega$, and $V_{ref} = -5 V$, giving the overall relationship in Equation 9 below.

Equation 9: Second-Stage Output Voltage of the pH Sensor Pre-Amplifier

$$V_{out2} = 2.5 - (2 * V_{in+})$$

This preamplifier therefore delivers a filtered, level-shifted signal that occupies a strictly positive range of about $3.5 V$ to $1.5 V$ for a $\pm 500 mV$ probe input, while ensuring that the probe's large output impedance does not impact the integrity of the incoming data. The amplitude and baseline are now set to values that the downstream stages expect, so the

output feeds directly into the previously described signal conditioner. In that block, the signal can be offset and scaled under software control and then converted by the ADC, completing the measurement path from the pH probe to the microcontroller.

2.4.6. pH Sensor Preamplifier Simulation

The pH preamplifier was simulated in PSpice for TI. This tool provides a model for the TL082 JFET op-amp, which is the component used on the PCB, so the simulated behavior aligns well with the intended hardware. Figure 18 shows the schematic used for the simulation. The goal is to verify that the two stages deliver the required amplification and level shift for the sensor that swings from -500 mV to $+500$ mV.

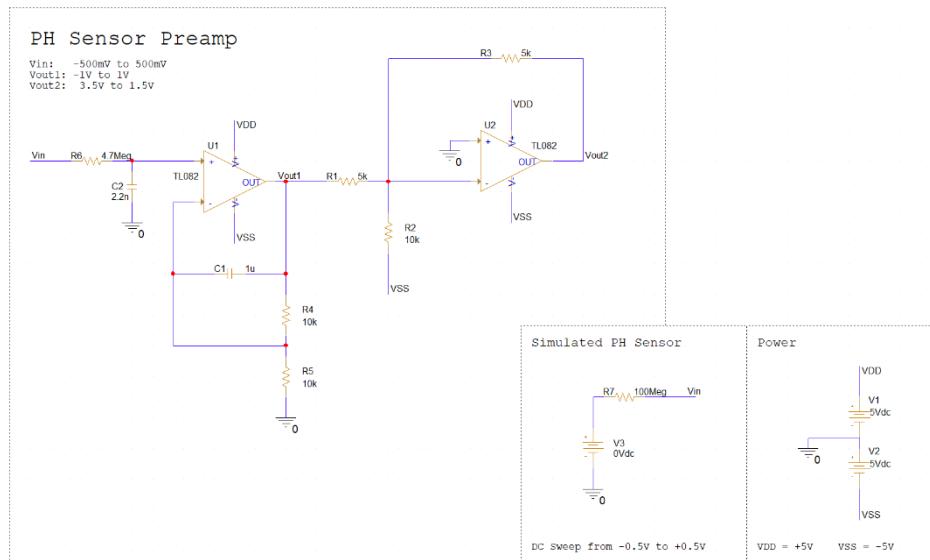


Figure 19: PSpice Simulation Schematic for pH preamplifier

A DC voltage source in series with a $100\text{M}\Omega$ resistor is used to emulate the high impedance pH sensor. All remaining component values match the design in Figure 17. The simulation is configured to do a DC sweep of the sensor source so the x-axis of the plot corresponds to the input voltage. Probes are placed at the sensor input, at the output of the first stage, and at the final output. Figure 19 presents the resulting traces. The green trace is the emulated sensor input, the pink trace is the output of stage one, and the blue trace is the output of stage two.

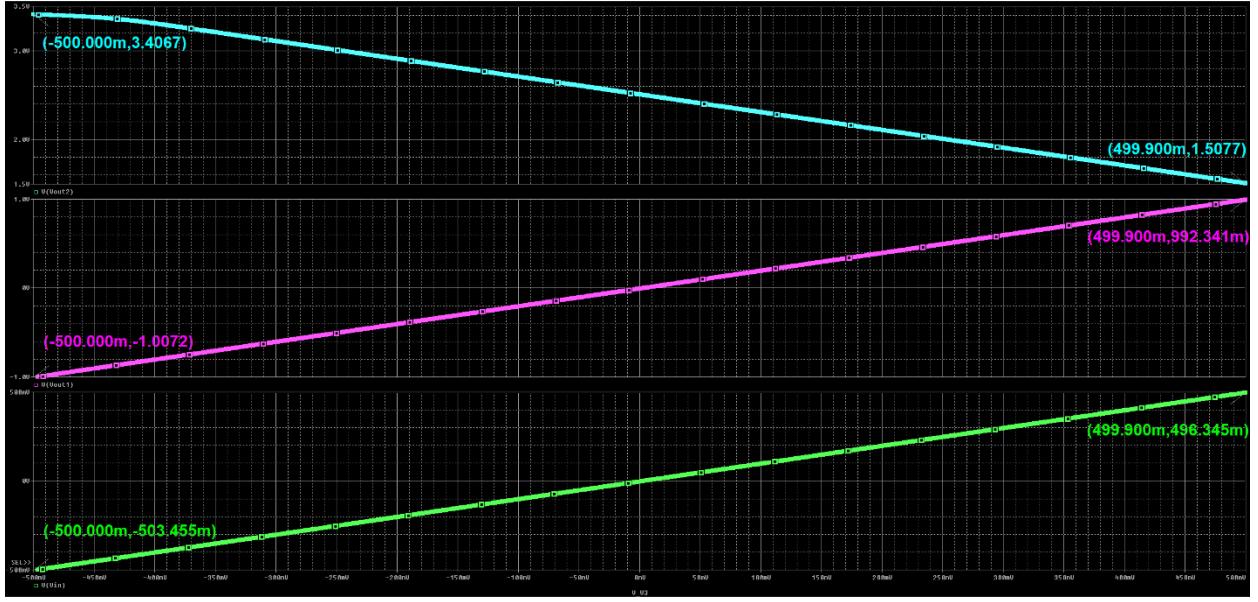


Figure 20: Simulated pH Preamplifier Plot

The plot shows that V_{in} spans -503mV to 496mV, slightly different from the intended -500mV to +500mV. This small deviation is likely due to the source resistance and the RC input network. The first stage output is -1.007V to +0.9923V, which is very close to the expected -1.0V to +1.0V for a gain of two. The second stage output is 3.407V down to 1.508V, which is about a 0.5% to 2.7% difference from the target 3.5V to 1.5V. These results indicate that the circuit performs as intended.

Overall, the simulation confirms that the preamplifier preserves the high-impedance input, applies the intended gain, and shifts the probe signal into the strictly positive range required by the downstream signal conditioner.

2.4.7. RP2350 Microcontroller

When selecting the microcontroller to be used for the development of the Sensor Node for the LoRa Water Quality Management System, three popular microcontroller platforms were evaluated for use: Atmel ATMEGA328P [38], commonly found on the Arduino Uno platform, the Espressif ESP32-WROOM-32 [39], and the Raspberry Pi RP2350 [40], used on the Raspberry Pi Pico 2. Table 20 below shows a comparison between these three platforms, based on the information listed in each device's manufacturer datasheet.

Table 20: Comparison of Microcontroller Platforms

Specification	ATMEGA328P	ESP32-WROOM-32	RP2350
Active Power Consumption (mW)	70	165	52.7
Low Power Consumption (μ W)	300	33	188
GPIO Pins	23	32	29
Serial Interfaces	UART, SPI, I2C	UART, SPI, I2C	UART, SPI, I2C
Clock Speed (MHz)	16	240	150
RAM (kB)	2	520	520
Program Memory Size (kB)	32	4,000	32,000 (external)

With these specifications in mind, the RP2350 microcontroller was ultimately selected for use with the LoRa Water Quality Management System sensor node because of its lower active power consumption and comparable general-purpose input/output (GPIO) and serial interface capabilities compared to an ESP32-based solution. Since the size of the compiled binary executable run by the microcontroller is in excess of 200KB and the microcontroller must manage over 13KB of amplifier calibration data in addition to all other data used within the system, the RAM and program memory sizes of the ATMEGA328P would be insufficient for the sensor node's firmware-based requirements. Beyond hardware capabilities, the RP2350 was chosen for this project because of its comprehensive reference design guides, robust and well-documented software development kit in the C programming language, and prior platform familiarity [41], [42].

Integrated into the larger sensor node module, the RP2350 microcontroller must manage the analog behavior of the software-defined instrumentation amplifier within the signal conditioner subsystem, conversion of analog sensor voltages to their real-world equivalent telemetry data, transmitting the incoming sensor data to the central gateway using the LoRa transceiver module, and the power consumption of the overall PCB.

The microcontroller manages these factors through its firmware, implemented entirely in the C programming language. For increased device control and universal compatibility with all hardware used on the sensor node PCB, the firmware was entirely implemented without the use of external libraries beyond the RP2350 C/C++ software development kit (SDK), which exposes an application programming interface (API) for hardware-level commands within the microcontroller, ensuring memory-safe writes to operation-critical registers and serial interfaces [41]. Though drivers were implemented entirely from scratch for all other ICs, such as the digital potentiometers and external flash, a low-level driver for the sx1262 LoRa modulator IC provided by the manufacturer was also incorporated into the firmware, again ensuring robust memory-safe access to low-level registers within the IC.

The sensor node's firmware is implemented at the high level using a finite state machine (FSM). As shown in Figure 20 below, the FSM is characterized by four distinct states: reset, sample, transmit, and hibernate, with an additional idle state if the sensor node fails to pass a power-on self test (POST). To ensure that any lock-ups or segmentation faults in the code do not prevent further operation of the sensor node when deployed in the field, the entire FSM is controlled by a watchdog timer. The watchdog timer will trigger a reset of the microcontroller after approximately 16 seconds if it is not updated. Thus, upon lock-up, the watchdog timer will not be updated within the code, and trigger a reset of the device. Upon state transition, the microcontroller updates the watchdog timer, resetting the timer to its original 16 second approximate countdown time. This ensures that the watchdog is only triggered if there is a lockup or fatal error in the code.

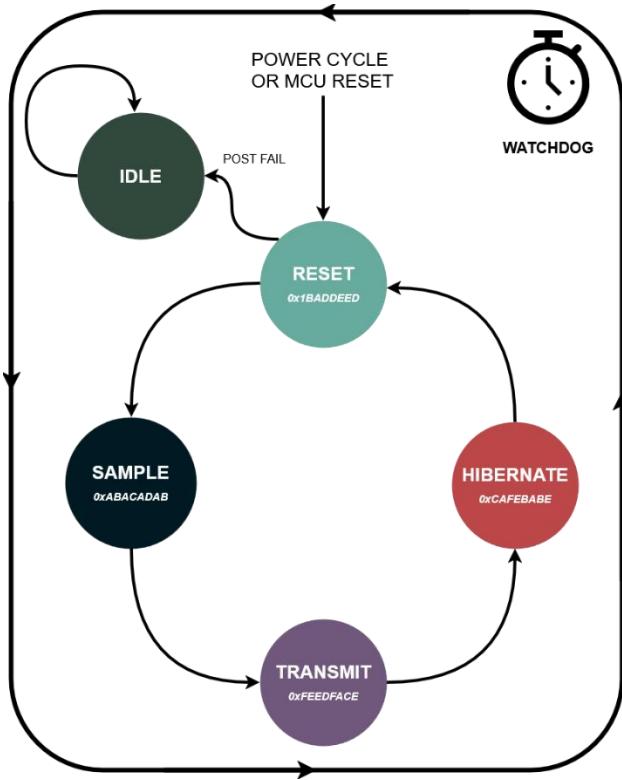


Figure 21: High-Level Sensor Node Finite State Machine

In the reset state, which the sensor node enters after a hard external power cycle or waking up from hibernation, the microcontroller initializes all GPIO pins, serial interfaces, and other internal control structures before performing a board-level POST. The POST ensures working board-level communications with the sx1262 radio over serial peripheral interface (SPI), the digital potentiometers and ADC over the inter-integrated circuit (I²C) communications protocol, and reads in sensor node configuration data and analog calibration data for the software-defined instrumentation amplifier from the external flash IC.

Once the POST is completed and passes, the sensor node enters the sample state, in which it measures the sensor voltage using the signal conditioner. For each sensor, the microcontroller configures the software-defined instrumentation amplifier according to a preset analog characteristic. Once the sensor input voltage is calculated based on the ADC's input voltage and the known characteristics of the amplifier from the calibration data, the voltage is converted to its equivalent turbidity, temperature, or pH value.

With the sensor telemetry obtained, the sensor data is then transmitted to the central gateway module using the LoRa transceiver module. As outlined in section 2.4.9, the raw telemetry data is formatted into a 19-byte packet, and data reception is validated on both ends of the LoRa channel using the Reliable Data Transfer 3.0 (RDT 3.0) communications protocol.

Following transmission of the input data, the sensor node enters the hibernation state, in which the 5V power rail is disabled, minimizing power draw from the sensors, the sx1262 LoRa modulator is put into its deep sleep state, and the RP2350 enters its lowest power state, in which the main logic core, RAM, and execute-in-place (XIP) logic are disconnected from power, leaving only the always-on logic block connected to the main power input. The sensor node maintains this state for 15 minutes, after which it boots back into the reset state and the FSM repeats. Since waking from the low power state is indistinguishable from a software perspective to a cold boot due to a system power cycle, three 32-bit unsigned integers are written to an on-chip power management scratch register before going into low power mode. These include a “magic number” to signal a boot from the dormant state, 0xBEEFCAFE, the last used packet ID for logging purposes, and the next state in the FSM, which is written for future expansion of the sensor node design. To signal an uncorrupted boot from the hibernation state, the states are similarly represented as “magic numbers,” with 0x1BADDEED, 0xABACADAB, 0xFEEDFACE, and 0xCAFEBABE representing the reset, sample, transmit, and hibernate states, respectively.

2.4.8. Buck-Boost Regulator

To extend the battery life of the sensor node using standard alkaline batteries, a buck-boost switching regulator was used to generate both the 3.3V rail used by the microcontroller and LoRa modulator, and a 5V rail used by the sensors and analog frontend. Though alkaline batteries have a nominal output voltage of 1.5V, manufacturers of these batteries generally consider an output voltage of 0.8V as being depleted [43]. Thus, the power input to the sensor node must maintain both rails at a constant value with a variable input voltage of 4.5V to 2.4V as the 3 D-Cell alkaline batteries cascaded in series deplete over the course of the sensor node’s operation. Since the 5V rail requires a constant boost mode and

the 3.3V regulator must operate in both the buck and boost regions, two buck-boost regulators were used in the node's development to reduce design complexity.

The Texas Instruments TPS63000 high-efficiency buck-boost converter IC was selected to generate the 3.3V and 5V power rails required for our node. The device is straightforward to implement and is well-suited for battery-powered applications, as recommended by its manufacturer. In addition, the IC includes an enable pin that can be controlled by a GPIO, allowing the system to disable unused rails and conserve additional battery power.

Though a reference design is provided by the manufacturer for the TPS63000, shown in Figure 21 below, calculations needed to be performed for the inductor value, input and output capacitors, feedback voltage divider resistors, and feedforward capacitor. First, the inductor selection was made using the maximum duty cycle of the device in boost mode, and the target peak current through the inductor according to the component datasheet, shown in Equation 10 and Equation 11 below.

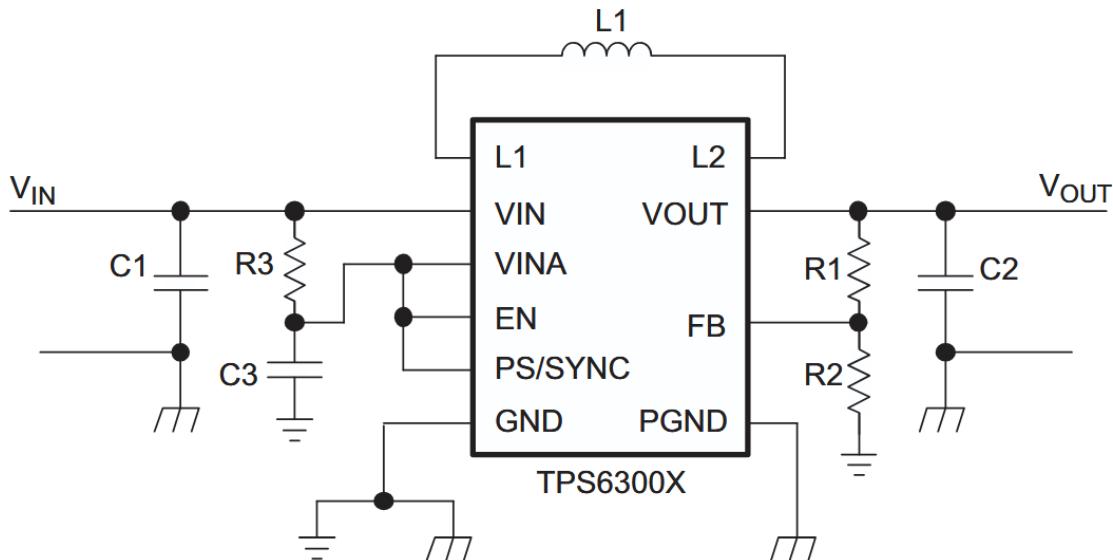


Figure 22: Typical Application Circuit for the TPS63000 Buck-Boost Regulator

Equation 10: Maximum Duty Cycle of TPS63000 in Boost Mode

$$D_{BOOST-MAX} = \frac{V_{OUT} - V_{IN-MIN}}{V_{OUT}}$$

Equation 11: Peak Inductor Current of TPS63000 In Boost Mode

$$I_{PEAK} = \frac{I_{OUT}}{\eta \times (1 - D_{BOOST-MAX})} + \frac{V_{IN-MIN} \times D_{BOOST-MAX}}{2 \times f_{SW} \times L}$$

Where η is the efficiency of the converter, typically 0.9, and f_{SW} is the switching frequency of the buck-boost regulator, typically 2.5MHz. Assuming a worst case input voltage of 2V input and 5V output, a duty cycle of 60% is expected in the worst case for the regulator. Thus, using a $2.2\mu H$ inductor limits the peak inductor current to approximately 1.5A assuming a worst-case output current of 500mA, well within the selected inductor's rated current of 2.1A [44].

With the inductor selection complete, the feedback path for the output voltage could be designed according to Equation 12 and Equation 13 from the component datasheet below [45]. The feedback path consists of a voltage divider, which produces 0.5V at the feedback node of the regulator when the output voltage is at the design specification, and a feedforward capacitor, which, placed in parallel with the top resistor in the feedback voltage divider (R_1 in Figure 22), adds a phase lead to the control loop, improving loop stability and transient response.

Equation 12: Feedback Resistor Selection for TPS63000

$$R_1 = R_2 \left(\frac{V_{out}}{V_{FB}} - 1 \right)$$

Equation 13: Feedforward Capacitor Selection for TPS63000

$$C_{ff} = \frac{2.2\mu s}{R_1}$$

Using these relationships, the feedback path components were selected as $R_1 = 560k\Omega$, $R_2 = 100k\Omega$, $C_{ff} = 4pF$ and $R_1 = 910k\Omega$, $R_2 = 100k\Omega$, $C_{ff} = 2.4pF$, for the 3.3V and 5V outputs, respectively.

Finally, the input and output capacitors for both regulators were selected as $22\mu F$ and $47\mu F$, as both values exceeded manufacturer design recommendations and using these values minimized the overall BOM complexity of the design.

2.4.9. Switched Capacitor Voltage Converter

Though the +3.3V and +5V power rails produced by the buck-boost regulator on the sensor node provide power to most ICs and other components on the board, the pH Sensor Pre-Amplifier and Software-Defined Instrumentation Amplifier components of the Signal Conditioner subsystem both require an additional -5V rail to power their respective operational amplifiers, as seen in Figure 18 and Figure 10, respectively. Since the power supply for the sensor node is entirely positive, the -5V rail could not be generated directly using a traditional switching DC-DC converter. To overcome this limitation, the Texas Instruments LM2660 Switched Capacitor Voltage Converter was integrated into the design to generate the -5V rail.

The operating principle of a switched capacitor voltage regulator centers around a charge pump topology. When power is first connected to the IC, a capacitor charges to the supply voltage, in the case of our node, +5V. Then, the IC uses four internal switches, as shown in Figure 23 below to disconnect the charged capacitor from the supply voltage and connect the positive terminal of the capacitor to ground, with the negative terminal of the capacitor connected to the output negative rail. Since the voltage across the capacitor can not change instantaneously, and the positive terminal of the capacitor is driven to ground by the internal circuitry of the IC, the output voltage of the IC is $-V_{in}$, or -5V in application. The switches are then driven by opposite phases of a reference oscillator, and capacitors are placed on both rails to smooth ripple voltages produced by the oscillation [46]. The typical application circuit for such an IC is shown in Figure 24 below; this application circuit was integrated into the final design for the sensor node PCB.

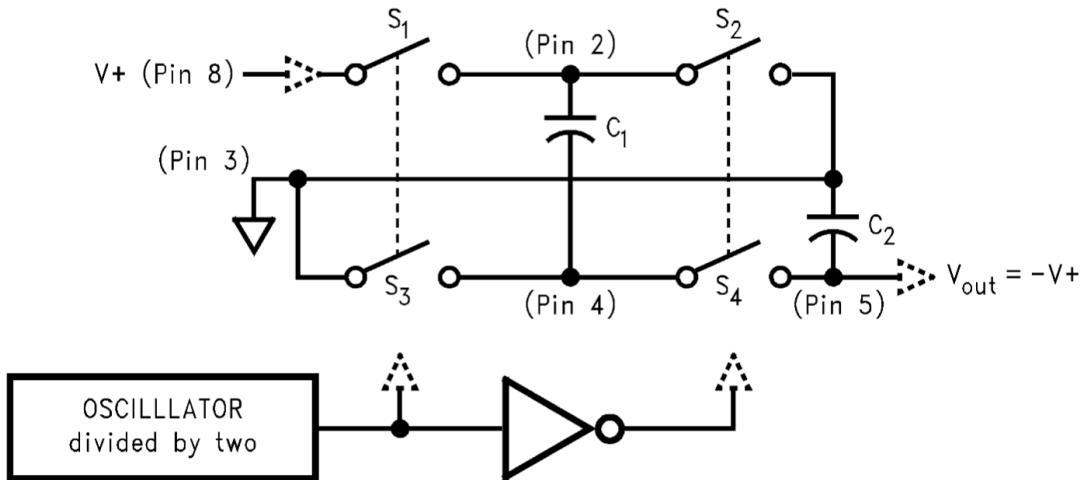


Figure 23: LM2660 Simplified Internal Design [46]

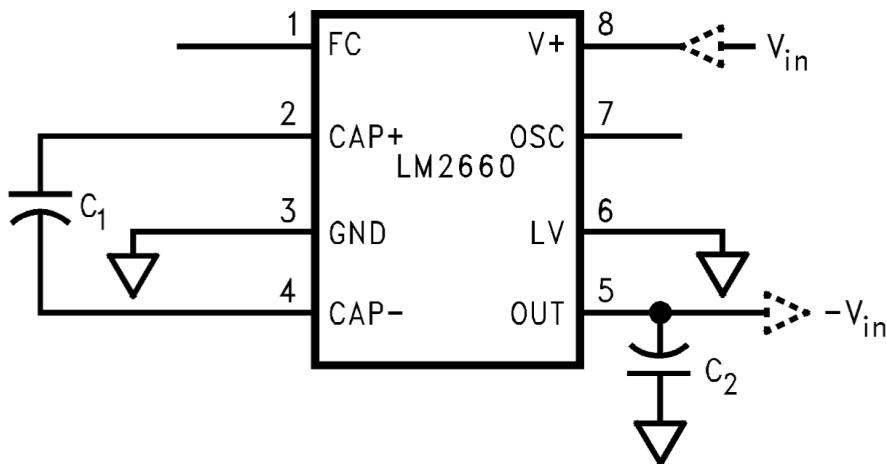


Figure 24: LM2660 Typical Application Circuit [46]

2.4.10. LoRa Modulator

The Semtech sx1262 RF Transceiver IC was selected for our design, as it is the industry-standard device for implementing communications over LoRa. Outside of its direct support of the LoRa communications protocol in hardware, the sx1262 provides comprehensive documentation, reference designs, and low power modes enabling further power savings when RF communications are not required. Furthermore, the sx1262 implements board-level serial communications using SPI, making it easily integrated with our chosen microcontroller platform. Our initial plan for the design phase for the sensor node was to

implement the sx1262 design at the board level according to reference schematics and PCB layout, however the Waveshare Core1262 HF module was ultimately selected for the sensor node [47]. Though this design approach reduces direct design control over the RF matching path and component selection, integrating a drop-in module with the design significantly reduces design complexity and RF integration risk.

The sx1262 RF Transceiver IC supports both the LoRa Communications Protocol, which uses Chirp Spread Spectrum (CSS), as well as communications using Gaussian Frequency Shift Keying (GFSK). For this reason, initializing transmit and receive operations using the device over these two protocols require different initialization sequences using commands that must be issued over the IC's SPI bus [48].

A timing diagram of an example SPI transaction is shown below in Figure 25, however these communications are implemented on the microcontroller at the hardware level through the RP2350 C/C++ SDK. SPI communications follow a Master-Slave topology and are performed using four of the microcontroller's GPIO pins, which are assigned as Master-Out Slave In (MOSI), Master-In Slave-Out (MISO), Serial Clock (SCK), and Slave-Select (NSS). When the NSS pin transitions to a logical low level, the SPI interface on the slave is activated, and data can be written to and read from the device using the MOSI and MISO pins, respectively. Both serial lines are synchronized using SCK, which is controlled by the master [48].

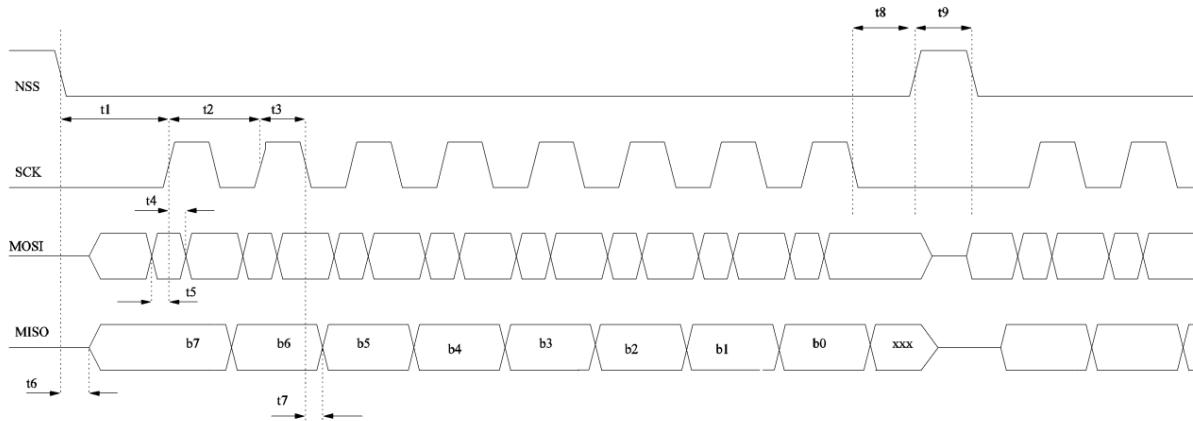


Figure 25: Example SPI Transaction for the sx1262 IC [48]

A flowchart for initialization of the transmit and receive operations on the sx1262 is shown in Figure 26 below, as implemented in the sensor node's firmware. Each command is structured as an operation code (opcode) followed by a sequence of data bytes which parameterize the command if required. The opcode and command sequence for each of the initialization steps is also shown for each block in the corresponding flowchart.

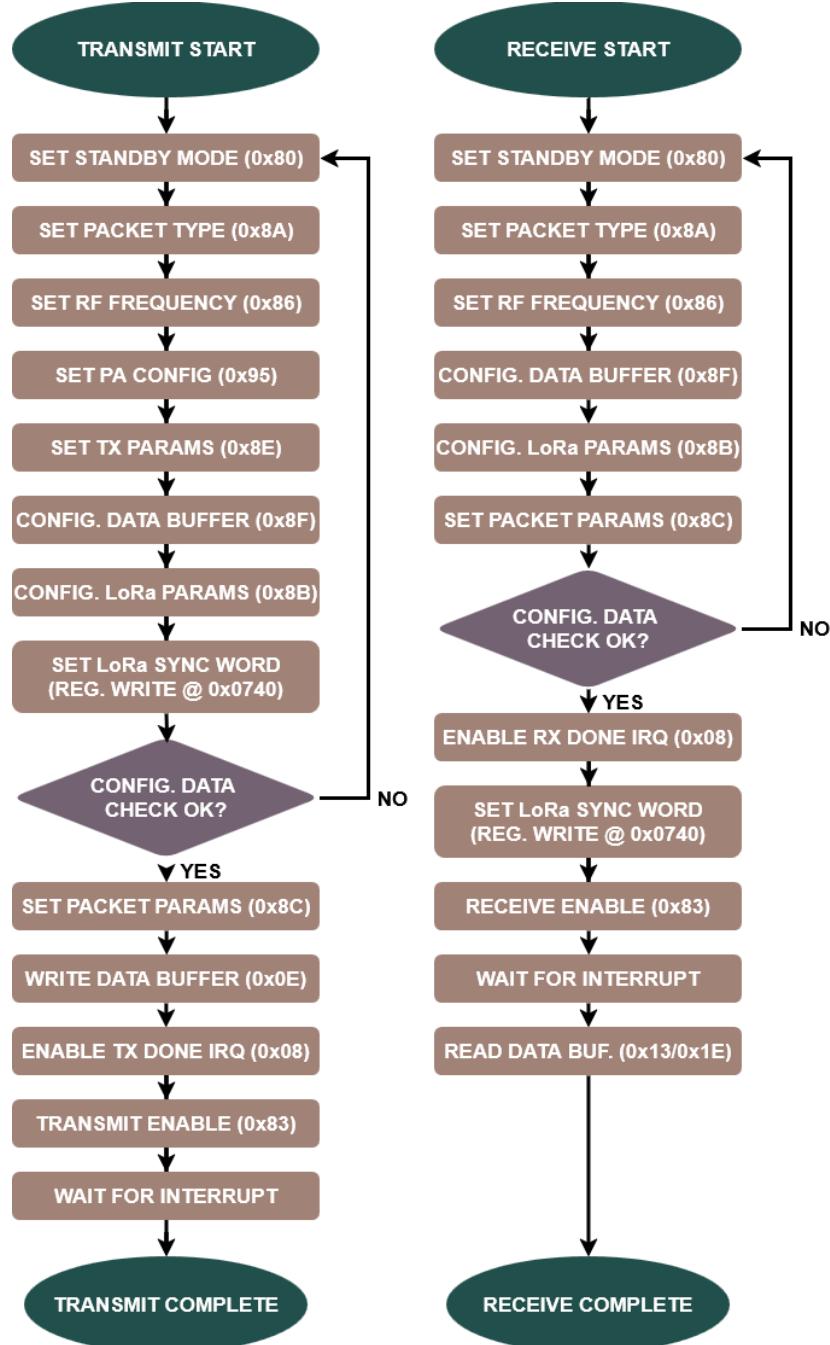


Figure 26: SX1262 Transmit and Receive Operation Initialization Process

With the LoRa Physical Layer (PHY) implemented using the SPI interface and initialization sequences shown above, reliable high-level communications are implemented between the sensor node and central gateway using the Reliable Data Transfer 3.0 (RDT 3.0) protocol. The transmitter and receiver for RDT3.0 are both implemented using finite state machines, as shown in Figure 27 and Figure 28 below. To implement both the transmitter and receiver for this protocol, both the transmit and receive PHYs must be implemented on the RF Transceiver, making streamlined initialization of both states critical to efficient operation. Expanding upon a simple open-loop communications topology in which the sender does not know if the destination device has received the data, RDT3.0 implements an acknowledgement scheme, allowing for efficient and reliable transfer of data throughout the LoRa Water Quality Management System [49].

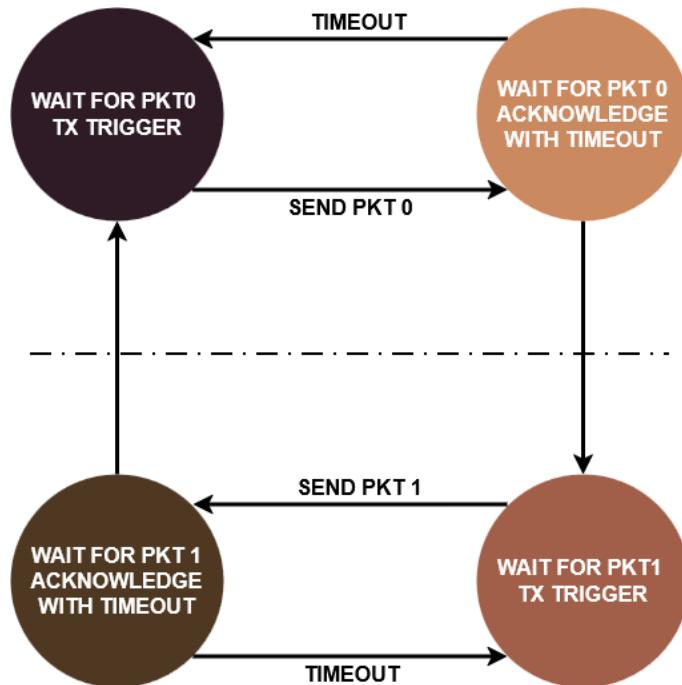


Figure 27: RDT3.0 Transmitter FSM Diagram

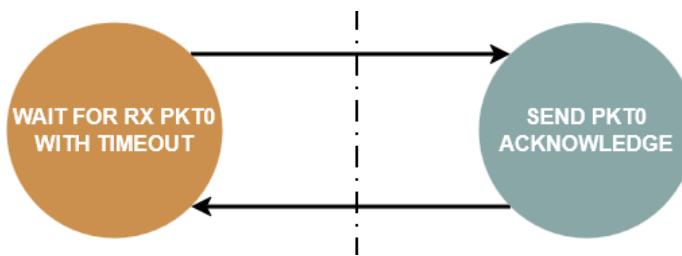


Figure 28: RDT3.0 Receiver FSM Diagram

With the hardware-level communications and setup of the device for use with LoRa communications established, the final concern for implementation of LoRa communication was compliance with the standards outlined in Section 2.1.6. These include CFR Title 47, Part 15, Subsection 247, and IEEE 1528.7. To meet both requirements, LoRa communications on the sensor node must not dwell on a single frequency for more than 400ms in a 20 second interval, and the time-averaged output power of the transmitter must not exceed 20mW. To simplify the firmware on the sensor node, all data on the sensor node is transmitted using a single 400ms packet with 915MHz center frequency with a +22dBm transmitter output power. This timing and frequency arrangement ensures compliance with former standard, as the transmitter will only be occupying the band for 400ms within a 15 minute interval. The latter standard is followed in this scheme as well, as a +22dBm power output.

The LoRa packet structure, shown in Figure 29 below, incorporates the payload data with a preamble, packet header, and Cyclic Redundancy Check (CRC). The preamble signals the start of a packet, and includes a synchronization word, which must be agreed upon by both the sender and receiver. Thus, the sync word acts as an indicator that the packet is intended for devices on a certain LoRa network. The header is used to provide information about the packet content, while the CRC is used to check for errors in the packet content.

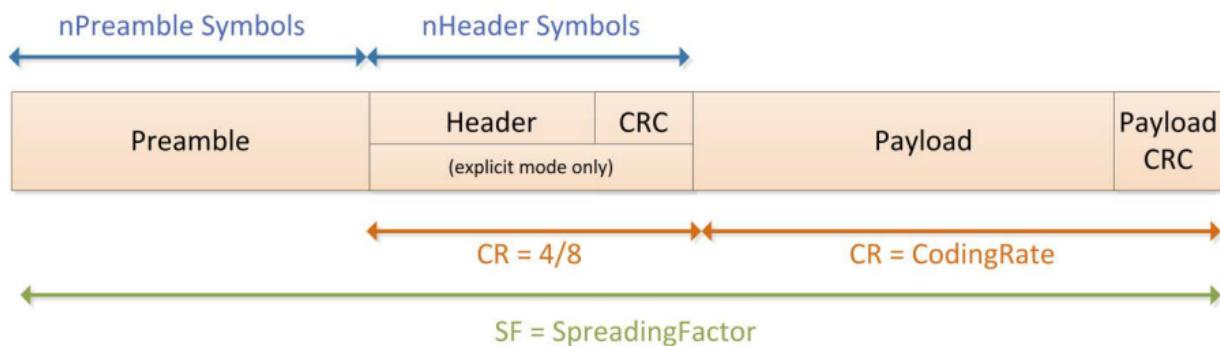


Figure 29: LoRa Packet Structure [48]

Using the LoRa time-on-air calculator from [50], the preamble length, spreading factor, output power, coding rate, bandwidth were selected as 8 symbols, SF10, +22dBm, 4/5, and 125kHz, respectively, allowing for a theoretical range of 4.73km and 370.68ms dwell time with a 19-byte packet length. Since FCC regulations require a dwell time of less than 400ms

within a given 20 second interval, these parameters maximize the potential range of the packet while also maintaining legal compliance.

With the packet parameters established to maximize range while maintaining legal compliance, the packet structure could be defined within software to ensure that the 19-byte length requirement was upheld. As shown in Figure 30 below, both the sensor node and gateway encode the packet data using a structure type in C, which allows for easy access and indexing of the original telemetry data while also enabling the data to be easily collapsed into an array of bytes to be transmitted over the air.

```

typedef unsigned char uint8_t;
typedef unsigned short uint16_t;

/**
 * @brief Different types of packets that the sensor node can send: telemetry or a message
 */
typedef enum lwqms_packet_types_e {
    LWQMS_PACKET_TYPE_TELEMETRY = 0,
    LWQMS_PACKET_TYPE_MESSAGE = 1
} lwqms_packet_types_t;

/**
 * @brief Structure to encode the three measured sensor values as floating point numbers.
 * Enables easy collapse to a uint8_t array for transmission.
 */
typedef struct lwqms_telemetry_s {
    float turbidity_measurement;
    float temperature_measurement;
    float pH_measurement;
} lwqms_telemetry_t;

/**
 * @brief Defines the ultimate payload contained in the packet. By using a union type, we can
 * represent both payload types using the same C type and same memory locations.
 */
typedef union lwqms_pkt_payload_u {
    lwqms_telemetry_t telemetry;
    char message[sizeof(lwqms_telemetry_t)]; // 12 bytes
} lwqms_pkt_payload_t;

/**
 * Packet Structure:
 *
 * 2 byte packet ID
 * 2 byte destination ID
 * 2 byte source ID
 * 1 Byte Packet Type (Telemetry/Message)
 * 4 byte raw float x 3 (turb, temp, pH) OR 12-byte char array (message string)
 * = 19 bytes total
 */
typedef struct lwqms_pkt_s {
    uint16_t pkt_id;
    uint16_t dest_id;
    uint16_t src_id;
    lwqms_pkt_payload_t payload;
    uint8_t packet_type; // Enum type will be cast to uint8_t so we have only 1 byte used
} lwqms_pkt_t;

```

Figure 30: Packet Data Structure Used to Encode Sensor Node Telemetry Data

2.4.11. Turbidity Sensor

The DFRobot Gravity Analog Turbidity Sensor was selected for our node based on cost, availability, and ease of integration. It provides an analog output that can be read directly by our signal conditioner. In contrast, most industrial turbidity probes cost on the order of \$1,500 or more and often require proprietary interface hardware, making them difficult to integrate into custom systems within our budget and schedule. Given these constraints, the DFRobot sensor was the most practical choice for our application.

The sensor consists of an IR LED transmitter and an IR phototransistor with a water sample between them. Suspended particles scatter and absorb the IR light, reducing the amount that reaches the phototransistor as turbidity increases. The phototransistor is wired in series with a $4.7\text{k}\Omega$ resistor, forming a simple current-to-voltage stage. When fewer particles (lower turbidity) are present, more IR reaches the phototransistor, its collector current increases, and the voltage across the resistor rises. This node voltage is measured, passed through the signal conditioner, and sent to the ADC, from which turbidity is computed in firmware [51].

2.4.12. Temperature Sensor

The PT100 RTD (Resistance Temperature Detector) was selected for its accuracy, stability, and repeatability over a wide operating range. The PT100 has a nominal resistance of $R_0 = 100 \Omega$ at 0°C and follows the Callendar-Van Dusen (CVD) resistance-temperature relationship, shown in Equation 14 and Equation 15 below for temperatures greater than and less than 0°C , respectively [52].

Equation 14: CVD Temperature-Resistance Relationship for $T > 0^\circ\text{C}$ [52]

$$R(T) = R_0 [1 + AT + BT^2].$$

Equation 15: CVD Temperature-Resistance Relationship for $T < 0^\circ\text{C}$ [52]

$$R(T) = R_0 [1 + AT + BT^2 + C (T - 100) T^3].$$

The coefficients for the chosen PT100 RTD are as follows [52].

$$A = 3.9083 \times 10^{-3}, B = -5.775 \times 10^{-7}, \text{ and } C = -4.183 \times 10^{-12}$$

These expressions define the expected sensor resistance $R(T)$ as a function of temperature T [°C]. For acquisition, the PT100 is placed in a voltage divider with a known series resistor $R_s = 4.7 \text{ k}\Omega$ and a 5V source. The node voltage V is measured and, over the 0 to 100°C operating range, this node voltage is approximately 0.10V to 0.15V. Combining the divider relationship with the Callendar–Van Dusen model and solving for temperature gives a direct voltage to temperature conversion as shown in Equation 16 below.

Equation 16: Voltage to Temperature Relationship of the PT100 RTD

$$T(V) = \frac{-A + \sqrt{A^2 - 4B * \left(1 - \frac{47 * V}{5 - V}\right)}}{2B}$$

In practice, the node voltage is first passed through the signal conditioner and then read by the ADC. The resulting voltage value is used to compute the PT100 temperature in real time.

2.4.13. Multiplexer

The TMUX1309 is the analog multiplexer used in our node circuit. It is a 4:1, dual-channel differential multiplexer. This means it can route one of four differential sensor inputs to the output. This lets us support our three current sensors, while keeping one spare input for a possible future sensor channel. Each channel is selected using address lines and an enable pin that can be driven directly from our microcontroller's GPIO. We placed this device between the sensors and the signal conditioner so that only one sensor is connected at a time. By multiplexing the sensor inputs, we can reduce the complexity of the analog frontend and BOM cost of the overall design.

2.4.14. Analog to Digital Converter (ADC)

The MCP3425 Analog Digital Convertor was chosen for our node circuit. It is a 16-bit, single-channel, low-noise, high-accuracy delta-sigma ADC with a true differential input. This makes it suitable for sensors that require differential measurement and precision readings. Power consumption was another key factor for choosing this ADC. In continuous conversion mode, the device typically draws about 145 μA. It also supports one-shot conversion modes to further reduce current. These features made the MCP3425 an

appropriate choice for our node. One limitation of this ADC is that its measurement accuracy relies upon a reference voltage of +2.048V, limiting its measurement range for single supply operation to just 0 through 2.048V. As such, the analog parameters of the software-defined instrumentation amplifier must be set such that this range is never exceeded [37].

Though the RP2350 microcontroller platform selected for use with the node includes Analog-to-Digital conversion on-chip, the performance of these ADCs are insufficient for precision use with the sensors used in the design. The RP2350's on-board ADCs have a listed 12-bit resolution, however have a listed effective number of bits (ENOB) of 9.2, greatly reducing the resolution of the device [40]. By comparison, the MCP3425 has a 16-bit base ADC resolution, with a listed ENOB of 16, giving it nearly twice the effective resolution of the RP2350's on-board ADCs [53].

Interfacing with the MCP3425 ADC is performed using the Inter-Integrated Circuit (I²C) communications protocol, which is essential for reliable and efficient communications between the microcontroller and the device, as it requires no additional hardware to facilitate serial data transfer. Like SPI, I²C utilizes a Master-Slave organizational structure, however it uses a bus topology, meaning that the serial data (SDA) and serial clock (SCL) lines are shared between the master and all slave peripherals. I²C devices use open-drain communication pins, meaning that each line functions as a “wired-and” connection when combined with pull-up resistors to the bus voltage. Each slave on an I²C bus has a unique address, which must be invoked by the master before communication with the slave can occur. Thus, I²C enables several peripheral devices to be connected simultaneously to the microcontroller without greatly increasing the routing complexity of the design. Typical bus topologies and timing diagrams used to communicate with the MCP3425 and other I²C peripherals are shown in Figure 31 and Figure 32 below, respectively [54].

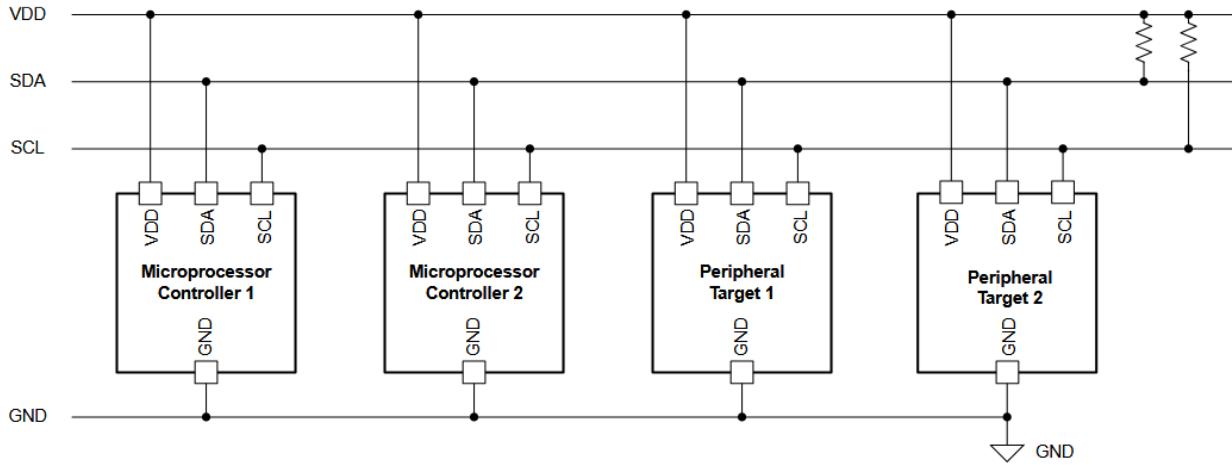


Figure 31: Typical I²C Bus Topology [54]

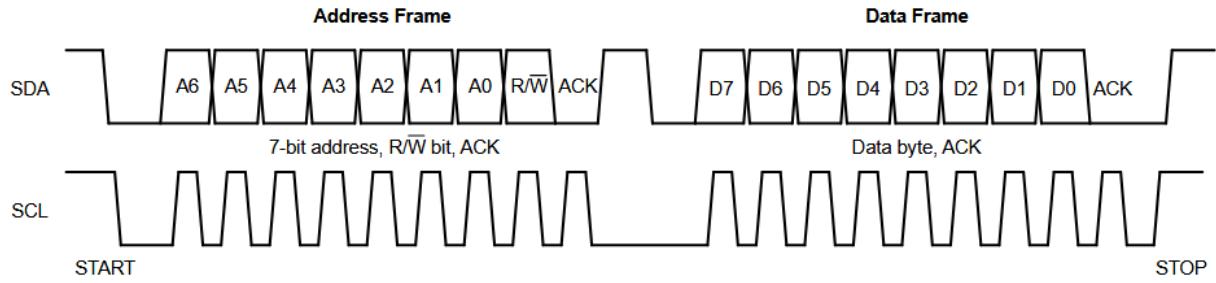


Figure 32: Typical Timing Diagram of an I²C Transaction [54]

2.4.15. Digital Potentiometers

To control DC offset levels and gain values within the Software-Defined Instrumentation Amplifier component of the Signal Conditioner Subsystem, digitally controlled potentiometers were used to autonomously set these values within the microcontroller's firmware. To implement this functionality, the MCP4651-503 digital potentiometer was selected, as it provides a $50k\Omega$ potentiometer resistance to minimize power consumption for the DC level generation and provide a wide gain range with the instrumentation amplifier. Additionally, the IC features two potentiometers with 257 taps each. Assuming a nominal $50k\Omega$ end-to-end resistance and 5V supply voltage for the input offsets, this allows for the amplifier to achieve a resolution of approximately 195Ω and 20mV for the gain resistance and DC offsets, respectively. The MCP4651 additionally features an I²C serial interface with programmable address, allowing for multiple instances to be cascaded with

the ADC to form a single serial bus for the software-defined instrumentation amplifier, as outlined in Figure 31 and Figure 32 above [54], [36].

2.4.16. SPI Flash Memory

The Macronix MXL25L3233F NOR Flash (SPI Interface, 32Mbit) is used to store calibration data for the sensor node and other configuration data used by the microcontroller for efficient operation. During boot-up of the node, the microcontroller reads the data from memory and uses it to set digital potentiometers to the desired gain and DC offset values. All the calibration information is retained in the flash ensuring consistent performance during power cycles and will allow for accurate calculations of sensor input voltages. The MXL25L3233F provides sufficient non-volatile storage and uses the same SPI interface used by the sx1262 RF Transceiver IC, as shown in Figure 25. It also offers read and write speeds of 133Mbps, and $33\mu W$ of power consumption in its idle state, making it a great choice for our sensor node [55].

In addition to the dedicated nonvolatile storage IC integrated onto the PCB, the Winbond W25Q128JVSIQ NOR Flash with SPI interface was used to store the compiled binary program executed by the microcontroller during sensor node operation. As mentioned in section 2.4.7, one of the primary advantages of the RP2350 is that it supports a program memory space of up to 32 MB (256 Mbit). This IC has an available 16 MB of storage, meaning that it leverages this advantage of the selected platform, and provides sufficient space for future firmware development. Furthermore, this IC has a Quad-SPI (interface), a command set which is compatible with the RP2350's bootloader, and maximum read and write speeds of 266Mbps using both serial output pins, making it sufficiently fast to facilitate operation at the RP2350's clock speed of 150MHz [56].

2.4.17. Logic Level Translator

As outlined in Section 2.4.7, the GPIO pins of the RP2350 microcontroller operate exclusively within the 3.3V logic level domain. Applying voltages above this level to the GPIO pins of the microcontroller would likely result in them burning out and becoming nonfunctional. Thus, interfacing the 3.3V I/O logic of the RP2350 with the analog frontend of the signal conditioner subsystem, which operates on a 5V supply to capture the full

output range of the sensors, can not be done directly. As such, the Texas Instruments LSF0102 Logic Level Translator IC was integrated into the design, facilitating I²C communications between the microcontroller and peripheral devices on the analog frontend. As shown in Figure 33 below, the LSF0102 is implemented as a single MOSFET per channel, facilitating communications between otherwise incompatible logic level domains. The example shown in the figure below is for translation between 3.3V and 1.8V logic level domains, however the IC is compatible with many other voltage combinations. Since the output is driven only by the single MOSFET, the output side of any channel must be implemented using a pull-up resistor to the target supply voltage level if the device is being used for up-translation. If both sides of the translator are implemented with pull-up resistors instead of being driven by a push-pull output, the logic level translator may facilitate bidirectional communications between both logic level domains, key for implementing the I²C level translation used in our design [57].

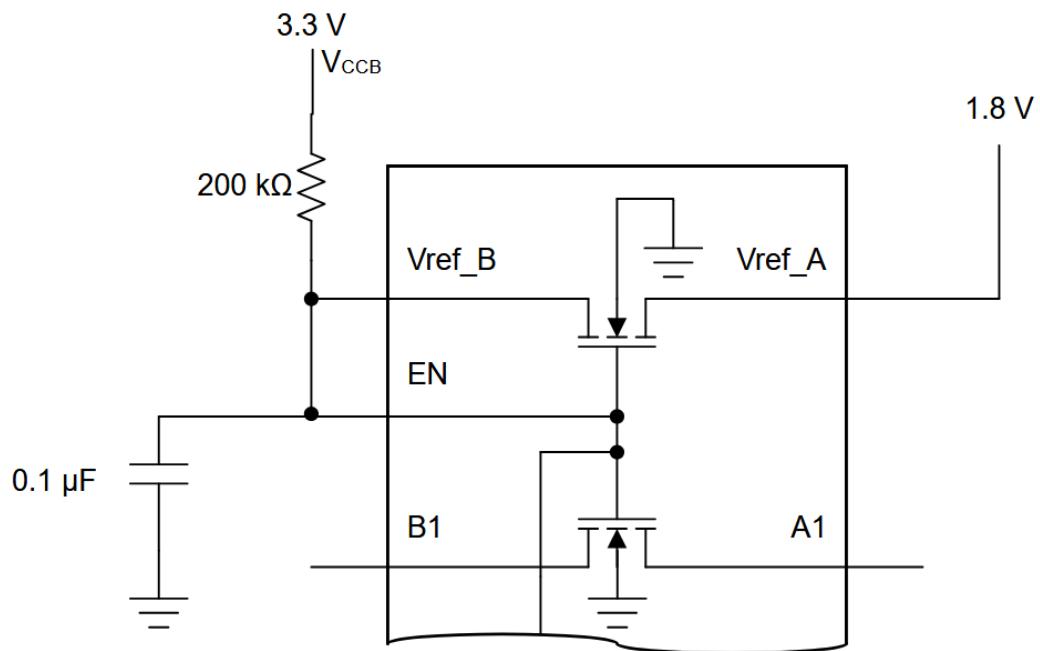


Figure 33: Internal Driving Circuitry of the LSF0102 Logic Level Translator [57]

2.4.18. Neural Net Theory

The neural network (NN) was designed using the deep neural network designer toolbox in MATLAB. The NN designed for this project used a long-short term memory (LSTM) network, which is a type a recurring neural network (RNN). RNN are especially suited for time series forecasting, since it backpropagates past information to improve the performance of the neural network [58].

2.4.19. Neural Net Design

The architecture of the NN included an input layer, an LSTM layer, a dropout layer, and an output layer. The input layer has an input size of three, and the output layers has an output size of three, since there are three water quality metrics being collected—temperature, pH, and turbidity. On the LSTM layer, the output mode chosen was sequence and the number of hidden units chosen were 128. This output mode returns the output of every time step, which makes it the network was designed for time series forecasting. The number of hidden units was chosen through an iterative process by seeing which design resulted in a lower root-mean-square error (RMSE). Lastly, the dropout was added to avoid overfitting the model to the training data.

The data set used for training the neural network was obtained from the Queensland Government [59]. The dataset contains water quality metrics taken at 15-minute intervals measured from the Brisbane river from August 2023 to June 2024, with a substantial gap of missing data around May 2024 and other sporadic missing data, shown in Figure 34. Various data cleaning techniques were used to clean the missing data, remove outliers in the data, and smooth the data. For the substantial gap, the skewness and kurtosis of the datapoints before the long gap were calculated, and these statistics were used to fill in 1 in every 10 missing data point of long gap to more accurately reflect behavior of data. Then, the rest of the missing data points were filled using the modified Akima cubic Hermite interpolation. Outliers were removed using different methods for each water quality metric, and the data was smoothed using a moving average. Finally, normalized the data using z-score method for a better fit and to prevent the training from diverging. The cleaned dataset used before normalization is shown in Figure 35.

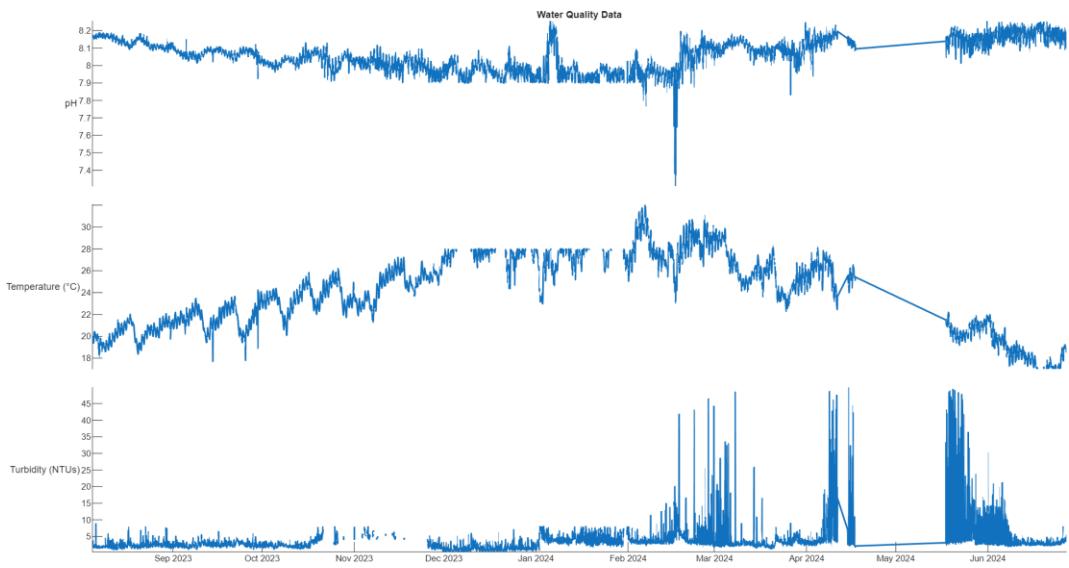


Figure 34: Raw Water Quality Data Set for use with Neural Network Training

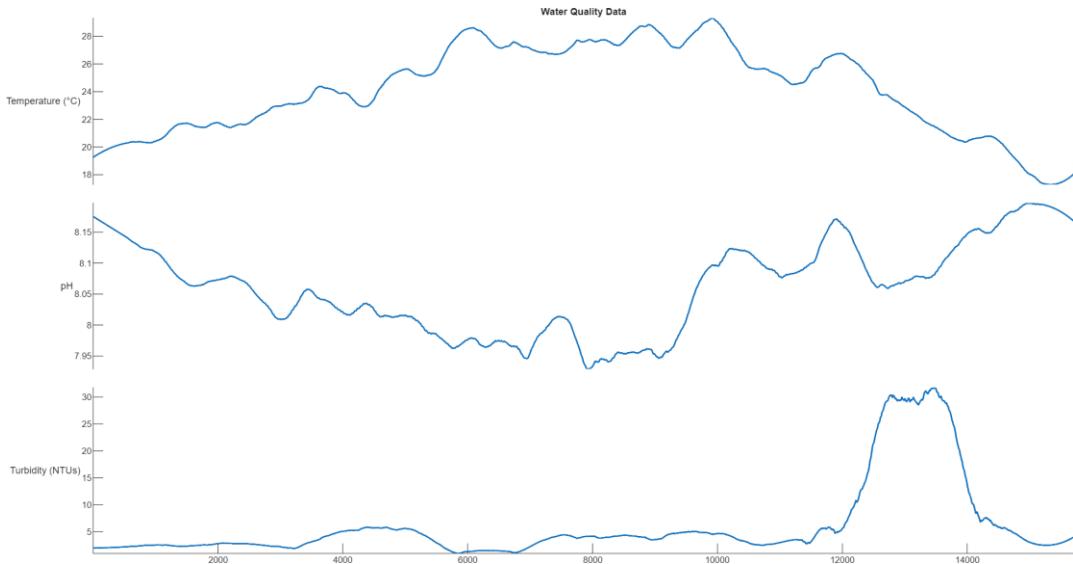


Figure 35: Cleaned Data for use with Neural Network Training

2.4.20. Gateway

The gateway is implemented using a Raspberry Pi 4 single-board computer (SBC) that receives all transmissions from the sensor nodes. The Raspberry Pi 4 was selected because

it is a budget-friendly SBC that provides sufficient processing power, memory, and I/O capabilities to support data logging, neural-network inference, and real-time data visualization. The Pi 4 interfaces with a separate PCB called the Gateway Receiver, which handles the LoRa wireless link to the sensor nodes. The complete gateway assembly, including the Raspberry Pi 4 and Gateway Receiver PCB housed in an enclosure, is shown in Figure 36 below.

The gateway design consists of three main elements: the LoRa receiver, the gateway dashboard, and a neural-network module. During operation, the Gateway Receiver forwards incoming LoRa packets to the Pi 4 over USB serial communication. These packets contain pH, temperature, turbidity, a timestamp, and a node ID. On the Raspberry Pi 4, a Python listener script continuously monitors the USB serial connection to the Gateway Receiver. Whenever a valid packet arrives, it parses the timestamp, node ID, and parameter values and appends them to a CSV log file. A separate Python dashboard script periodically reads this CSV file, uses the data to update the gateway dashboard graphical user interface (GUI), and performs alert detection. When it detects that a parameter is out of range, or that no update has been received from a sensor node for more than 15 minutes, it generates an alert that is displayed on the dashboard and sent by email. Integration of the neural network onto the Raspberry Pi is planned as future work and was not completed within the project timeline

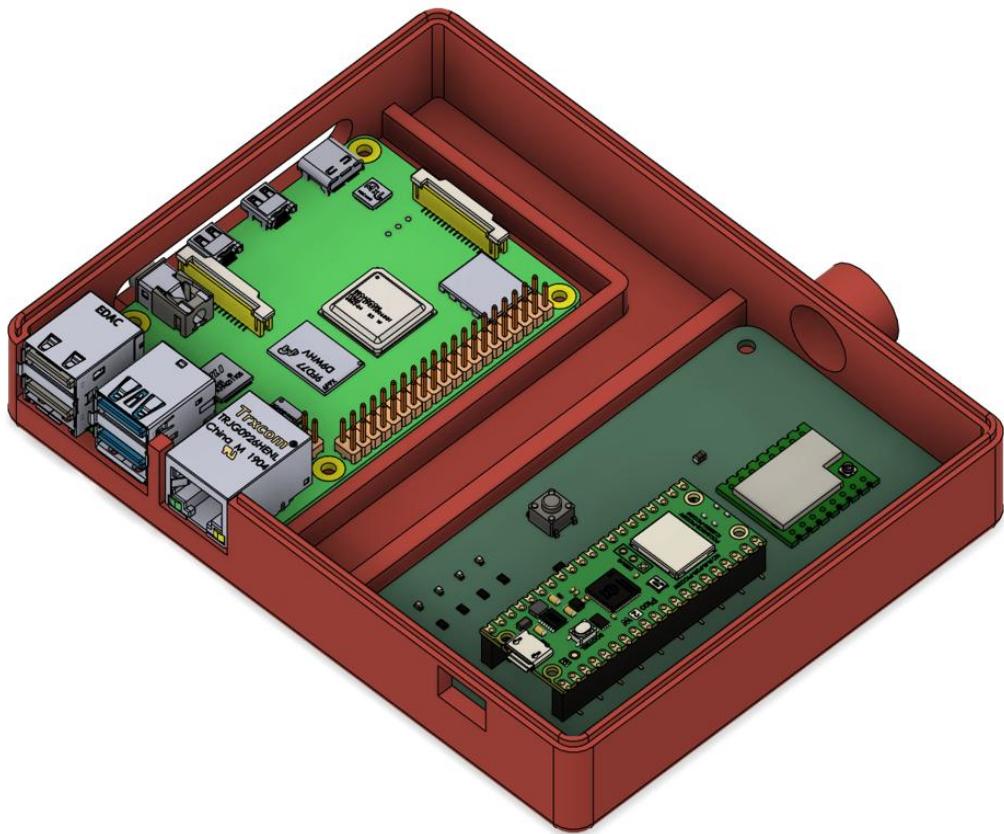


Figure 36: Gateway Design with Receiver PCB

2.4.21. Gateway Receiver

The Receiver Portion of the LoRa Water Quality Management System integrates the RP2350 Microcontroller and Waveshare Core1262 LoRa Modulator into a simple PCB. To reduce design complexity, the RP2350 is implemented using the Raspberry Pi Pico 2 Microcontroller development board. A single SPI bus is used on the design to facilitate communication between the microcontroller and LoRa Transceiver Module, along with indicator LEDs to communicate the current operational status of the board. The schematic and layout for the gateway receiver PCB are shown in Figure 37 and Figure 38 below, respectively. A final populated version of the PCB is shown in Figure 39 below.

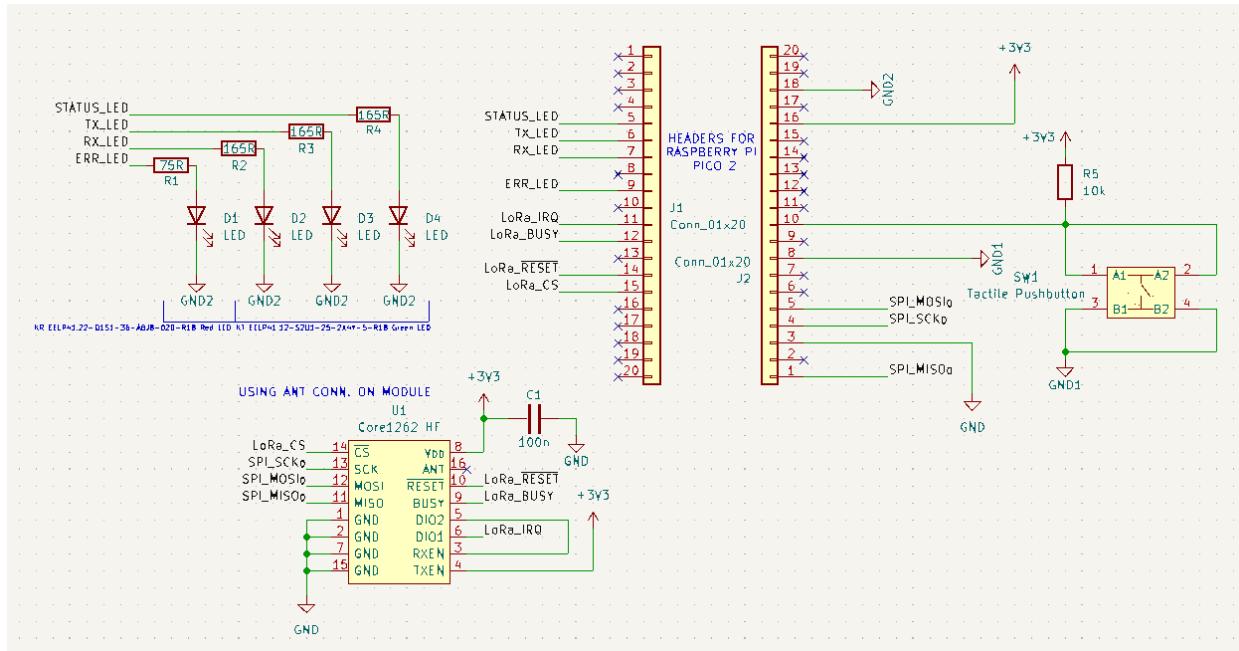


Figure 37: Gateway Receiver Board PCB Schematic

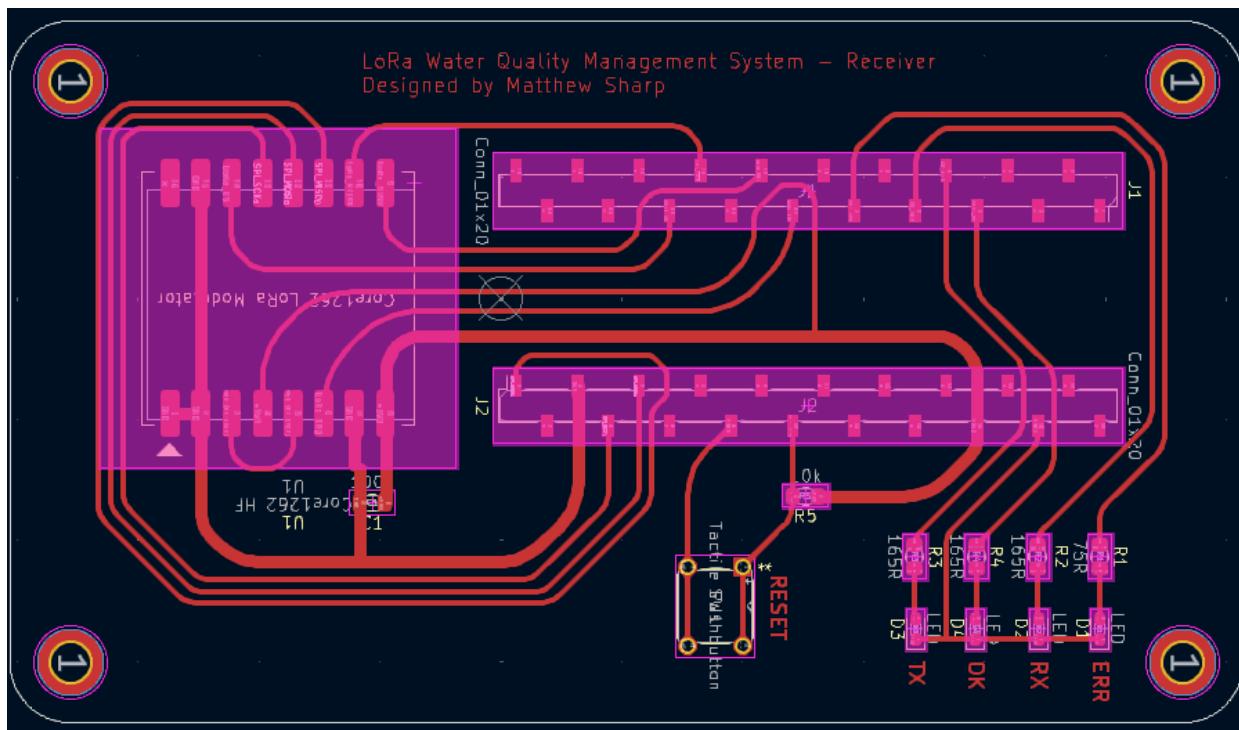


Figure 38: Gateway Receiver PCB Layout

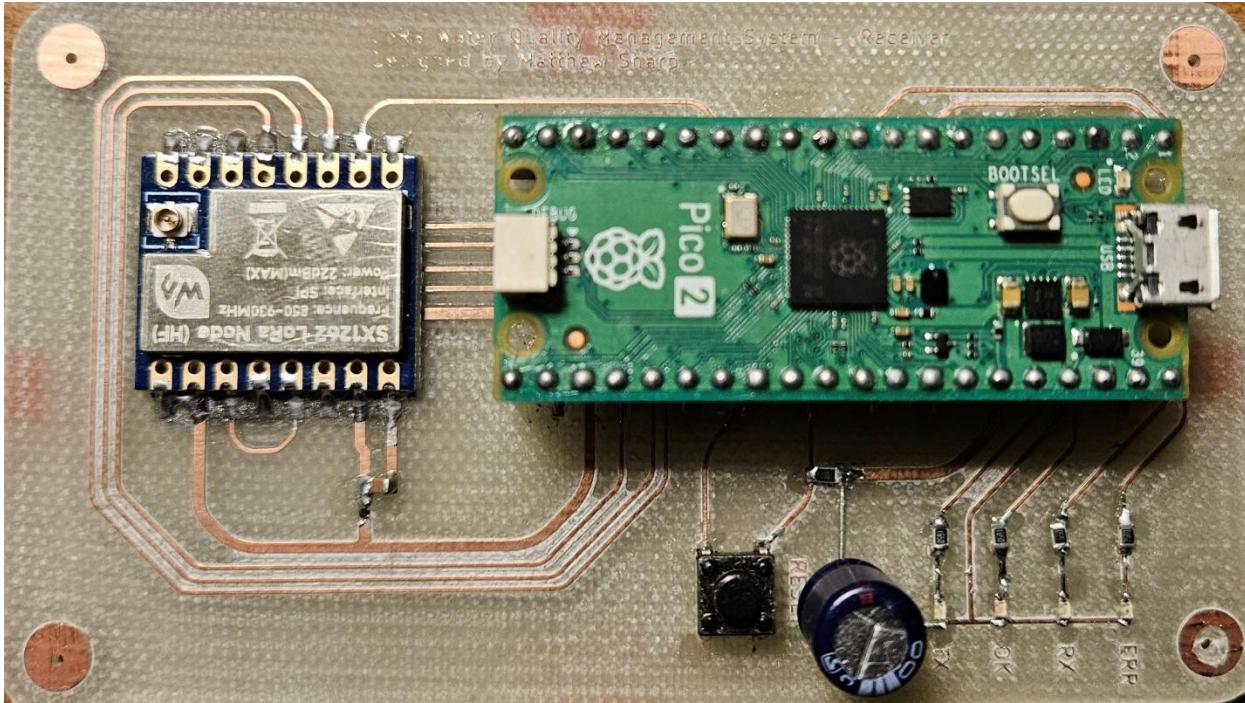


Figure 39: Gateway Receiver PCB, Populated

The RP2350 microcontroller implements the RDT3.0 Receiver FSM as outlined in Figure 28 above. As incoming packets are received, the microcontroller reports the incoming data using the console port, which is interpreted and saved to CSV files using a python script running on the Raspberry Pi 4, included in the GitHub Repository linked in Appendix C.

2.4.22. Gateway Dashboard

The gateway dashboard GUI was designed to provide a clear, easy-to-read display for water quality technicians. In addition to presenting the current measurements, the dashboard performs problem detection when the gateway has not received a transmission from a node for more than 15 minutes, or when any sensor parameter for a node falls outside its specified threshold. When an alert is triggered, it is both displayed on the dashboard and sent to technicians by email.

The dashboard was implemented in Python using the Tkinter library to generate the GUI. The complete dashboard can be viewed in Figure 40. The dashboard script reads the CSV file generated by the listener script on the Raspberry Pi, which logs the timestamp, node ID, and sensor parameters. The current implementation displays two sensor nodes, but the

layout can be extended to accommodate additional nodes in the future. Each row shows a node ID, the timestamp of the most recent packet received from that node, and the corresponding pH, turbidity, and temperature values. The current date and time are also shown in the lower corner of the main window. Technicians can view node history by pressing the “Node # History” button, which opens a separate window that displays past timestamps and parameter values in a scrollable table. This history window is shown in Figure 41 below.

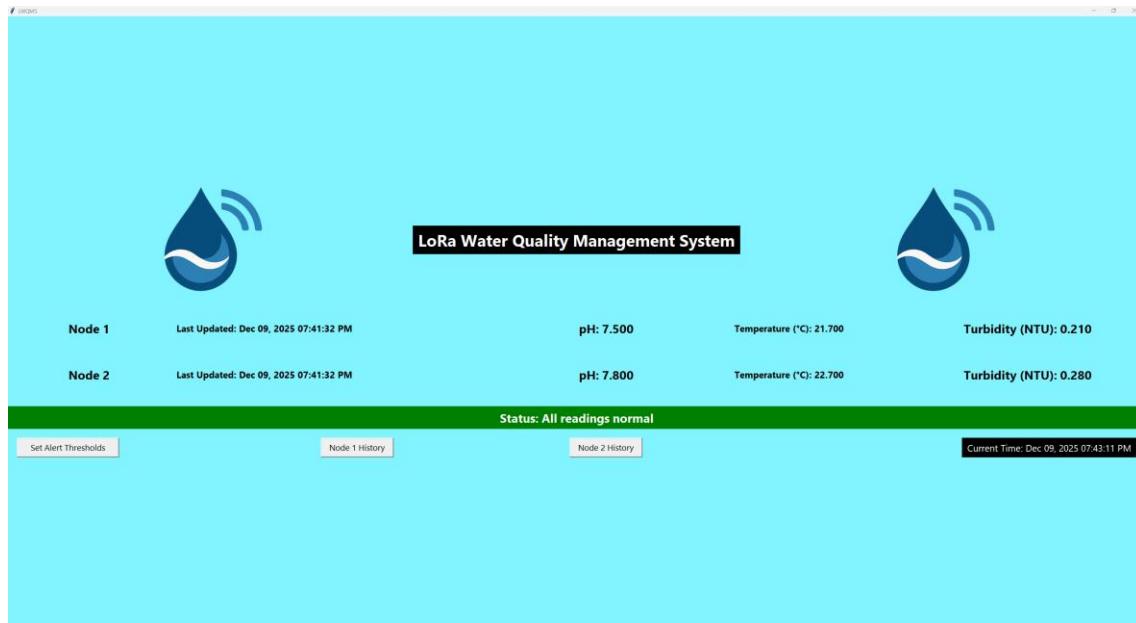


Figure 40: LoRa Water Quality Management System Dashboard Design

Figure 41: Dashboard Node Data History Window

The dashboard updates every 5 seconds. At each update, it checks whether each node has reported within the last 15 minutes and whether each parameter is within its configured thresholds. If all conditions are satisfied, a green status banner appears at the top of the dashboard with the message “Status: All readings normal.” If any condition is violated, the banner turns red and displays a description of the current alert. An example of the banner when pH falls out of range is shown in Figure 42.

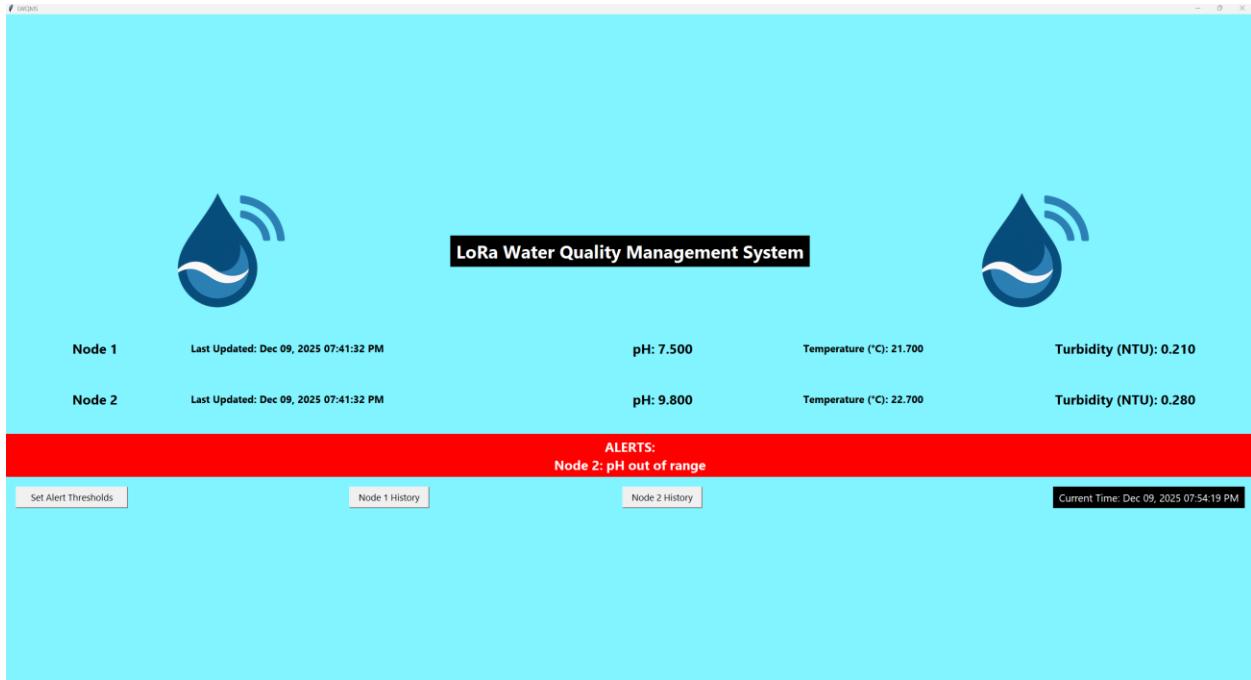


Figure 42: Dashboard with Alert Banner On-Screen

In addition to the status banner, the dashboard sends an email notification whenever a new alert is detected. Using a Python email library, a dedicated Gmail account was connected to the dashboard script. When an alert occurs, the script generates an email subject and message body that describe the issue and sends it to the configured recipient. Because the system uses the Gmail Application Programming Interface (API), technicians can receive notifications on a desktop client or directly on a mobile device with email access. An example email for a pH out-of-range condition is shown in Figure 43.

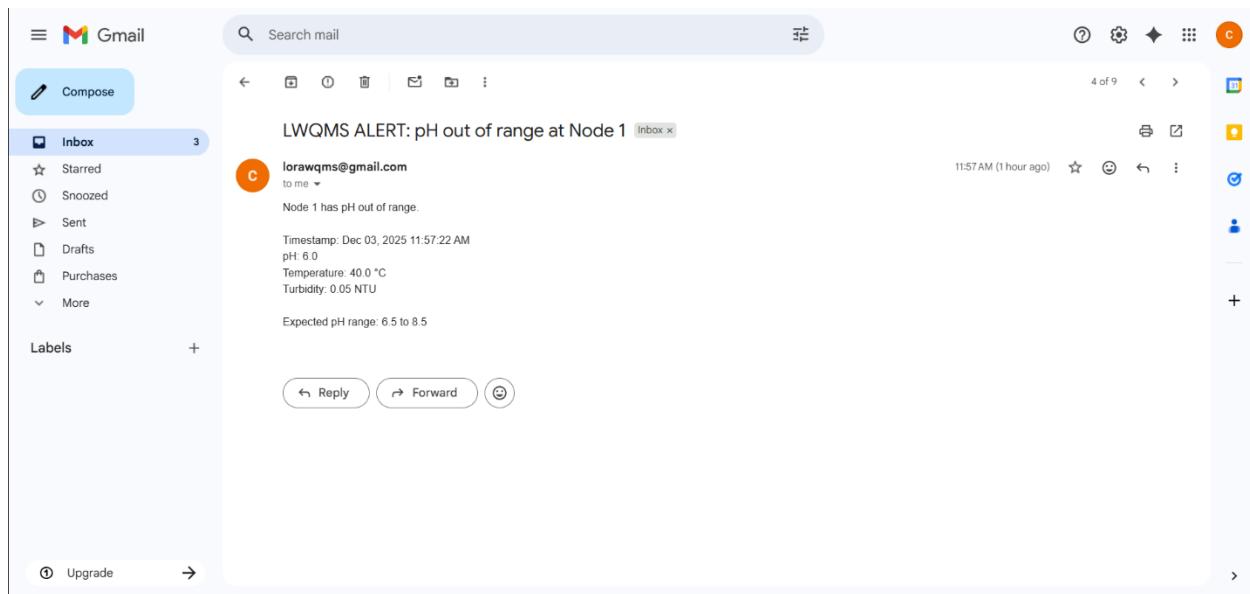


Figure 43: Mobile Water Quality Alert via Email

Alert thresholds are configured through a dedicated settings window, opened by pressing the “Set Alert Thresholds” button. By default, the pH thresholds are set to 6.5–8.5 and the turbidity threshold is set to 0–1.0 NTU, based on MCL guidelines, while the temperature range can be adjusted by the local water authority as needed. The threshold configuration window is shown in Figure 44 below.

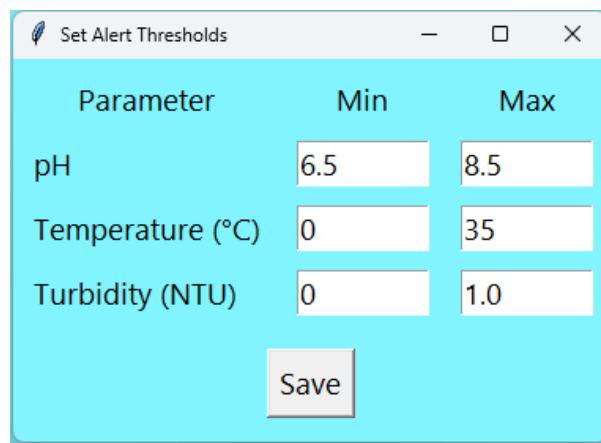


Figure 44: Customizable Parameter Settings in Dashboard

2.4.23. Sensor Node PCB

The sensor node PCB integrates sensor interface circuitry, the signal conditioner subsystem, a microcontroller, the LoRa transceiver module, USB, UART, and SWD interfaces, along with indicator LEDs and a RF signal path. This part of the system is responsible for interfacing directly with the turbidity, temperature, and pH sensors which will inform the decision-making logic at the central gateway.

The PCB was entirely designed using KiCAD, a freely available PCB design software. In addition to the features available within KiCAD the RF-Tools plugin was used to manage critical high-speed signal paths, as well as Saturn PCB Toolkit to determine PCB trace geometries for impedance-controlled traces.

Though most of the traces carry low-speed analog and digital signals, there were a few special considerations that needed to be made when developing the final design. First, the design involved a RF signal trace connecting the LoRa transceiver module and the external antenna, as well as USB 2.0 data traces. To maintain signal integrity within the PCB and prevent potentially damaging signal reflections, the RF signal trace had to be impedance controlled to 50Ω , while the USB 2.0 traces needed to maintain a differential impedance of $90\Omega, \pm 15\%$.

Next, the power supply into the sensor node needed to accept a wide input voltage range, manage heat dissipation effectively, and accommodate large switching currents through its inductor. Another key area of special concern is power distribution, as the PCB must accommodate four distinct voltage rails throughout the design, while maintaining low impedance and efficient routing.

The microcontroller layout was another area of concern within the design, as the chosen RP2350 microcontroller requires a substantial decoupling network, introduces a new power rail, and provides all the external interfaces on the board.

Finally, the PCB must minimize the propagation of high-frequency noise throughout the rest of the traces and outside of the PCB to improve signal integrity and minimize unwanted external noise.

The final PCB design can be divided into four main sections: power input and regulation, microcontroller layout, analog frontend, and RF interface. To minimize the effects of high-frequency noise on critical traces, the RF interface and switching regulators were isolated from the microcontroller and analog frontend.

As mentioned previously, the primary focus of this PCB design is to implement the signal conditioner subsystem, which provides a unified analog frontend which will be shared among all the sensors used with the system. It applies DC offsets and gain to the incoming signal, making them compatible with the MCP3425 ADC. The value read by the ADC is read by the microcontroller over an I²C bus, which formats the incoming data as real-world measurements, and transmits the telemetry to the central gateway using LoRa. The gain and DC offsets of the signal conditioner are controlled using digital potentiometers, making the signal acquisition process fully software-controlled and autonomous. Outside of managing the parameters of the signal conditioner and the LoRa interface, the microcontroller also manages the power consumption of the PCB, controlling the 5V rail by enabling and disabling the power regulator IC, disconnecting the sensors, and minimizing power draw while the system is idle.

Before the PCB routing was initiated, the schematic design was first finalized using KiCAD for schematic capture and analog simulation was performed using LTSpice and PSPICE for TI, as outlined in prior sections of this report. The complete schematic of the Sensor Node PCB is attached in Appendix F of this report.

With the component selections made as detailed in prior sections of this report, the component footprints were downloaded from the respective component manufacturers where available, and standard IC package footprints were used when models were not directly available. From there, the components were segregated based on to which PCB category they belonged – the RF interface and switching power supply were located towards the corners of the board to minimize noise, while the microcontroller was located centrally to maximize access to all digital signals. The analog frontend was placed at the bottom of the board for proximity to the sensor connectors. All other connectors were located such that they were close to the nets and board sections that they would directly

interface with. Before traces were routed, the board stackup and net classes were defined, producing a 4-layer board with impedance control on the USB traces and RF signal path. Since the board would be ordered through JLCPCB, critical impedance-controlled sections of the board were designed according to the JLC04161H-7628 Stackup, shown in Figure 45 below.

2) JLC04161H-7628 Stackup

Layer	Material Type	Thickness	
Top Layer	Copper	0.035mm	
Prepreg	7628*1	0.21040mm	
Inner Layer L2	Copper	0.0152mm	
Core>	Core	1.065mm	1.1mm H/H0Z with copper
Inner Layer L3	Copper	0.0152mm	
Prepreg	7628*1	0.21040mm	
Bottom Layer	Copper	0.035mm	

Figure 45: JLC04161H-7628 PCB Stackup [60]

With the board stackup defined, impedance-controlled sections of the board could then be designed using Saturn PCB Toolkit. The USB geometry was defined as a differential pair with 0.2mm track width and 0.14mm separation, giving a 90.049Ω differential impedance. The RF signal path was defined as a coplanar waveguide with 0.35mm track width and 0.15mm clearance, giving a 50.13Ω characteristic impedance. To maintain the topology needed for the differential traces and coplanar waveguide, the stackup of the board was defined to be signal traces, ground, power rails, signal traces, and ground, from top to bottom.

With the impedance-controlled trace geometries defined in software, PCB routing could begin according to these constraints. The first section that was routed was the power regulator, shown in Figure 46 below. This section's key concerns included thermal management and managing potentially high currents. Stitching vias were used on the power input to increase surface area for heat dissipation, and on power outputs to provide a low impedance path to the power planes. Traces to the inductor were maximized in

available width, supporting up to 1A of current through the inductor, well above the calculated switching current.

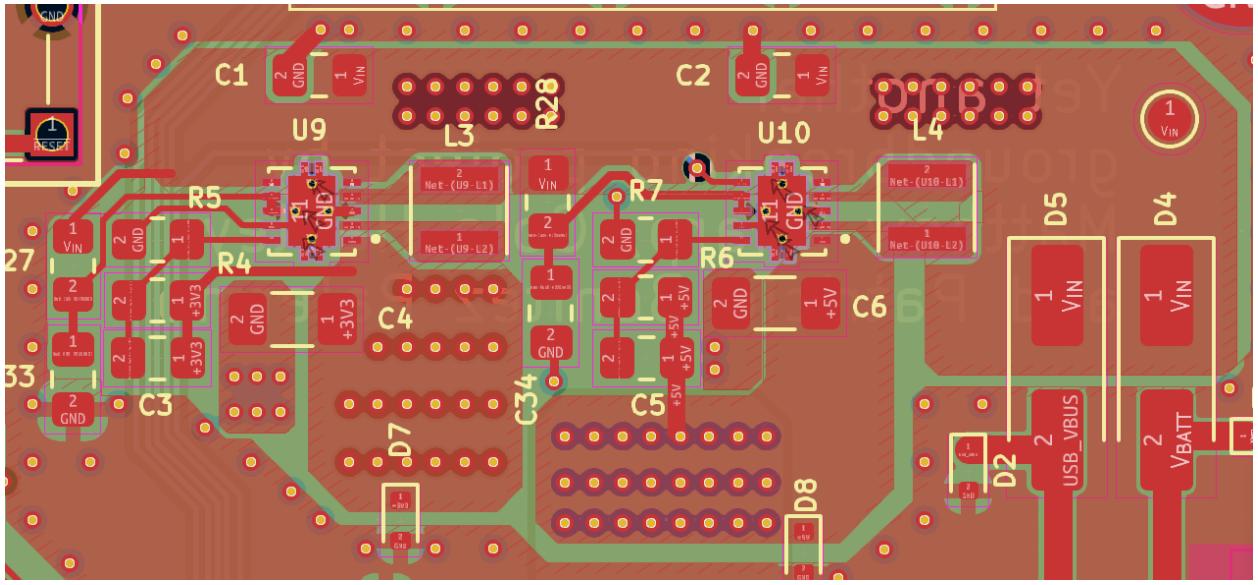


Figure 46: Buck-Boost Power Regulation PCB Layout

In addition to the two buck-boost regulators integrated onto the PCB, power management design also needed to be performed surrounding the LM2660 Switched Capacitor Voltage Converter IC, as shown in Figure 47 below. To provide the -5V rail to the operational amplifiers used throughout the signal conditioner subsystem, the LM2660 was placed on the rear side of the board to keep this power rail's source as closed to its destination as possible. The trace width to the capacitor undergoing switching, C37, was maximized, and direct connections to copper zones were used for the input and output filtering capacitors. The -5V rail is transferred to the front side of the board through an internal copper pour, shown as the yellow region directly below C38 in the figure. Using a copper pour in this enables a direct connection to the rail through a via.

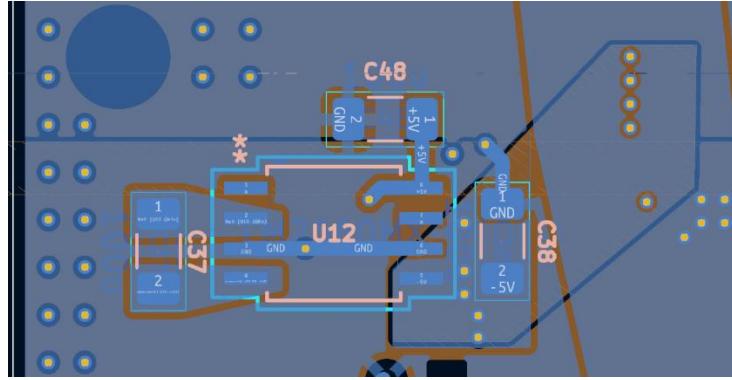


Figure 47: Switched Capacitor Voltage Converter Layout

The next section to be routed was the microcontroller, which relied heavily on the manufacturer's reference design [42], however it needed to be adapted for the 4-layer board stackup and larger 0805 surface-mounted components. The RP2350 is shown as U8 in Figure 49 below, with USB traces run directly to the USB Type-B port through 27Ω termination resistors. The internal power and ground planes have been hidden in the viewport shown to show the 1.1V copper pour used to supply the internal logic core of the microcontroller. This rail is produced by an internally-controlled buck regulator inside the RP2350 MCU, with external inductor and capacitors required for operation, shown as C20, C21, and L2 below. Key attention needed to be paid to keep these components as close to the footprint of the chip as possible while maximizing ampacity.

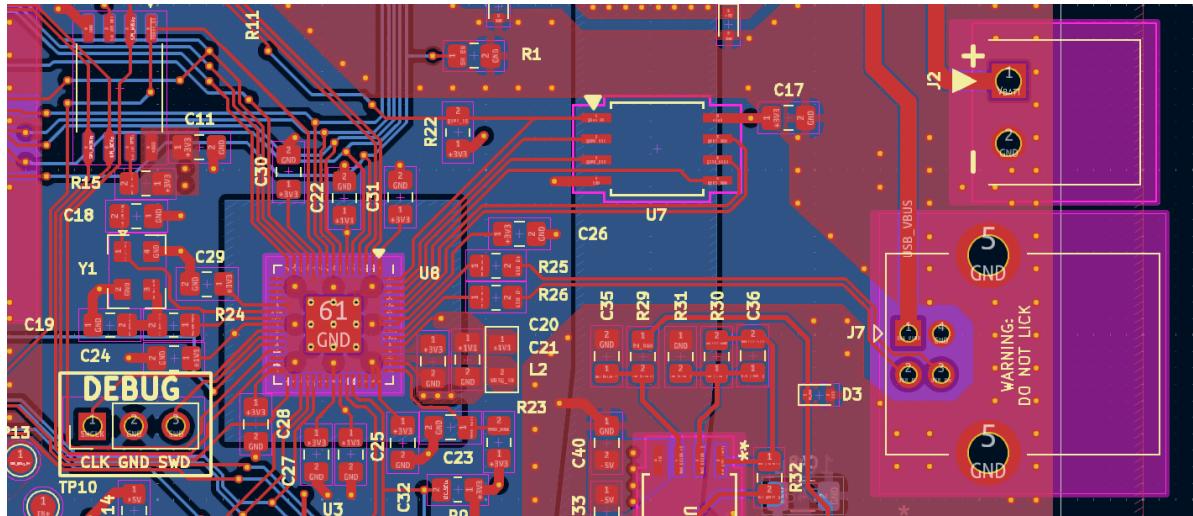


Figure 48: RP2350 and Peripherals PCB Layout

Next, the signal conditioner subsystem was routed, as shown in Figure 49 below. Important considerations for this section included prioritizing the I²C traces in the routing order for reliable communication with the digital potentiometers and ADC. Analog traces needed to be kept short, while also maintaining access to test points for hardware debugging.

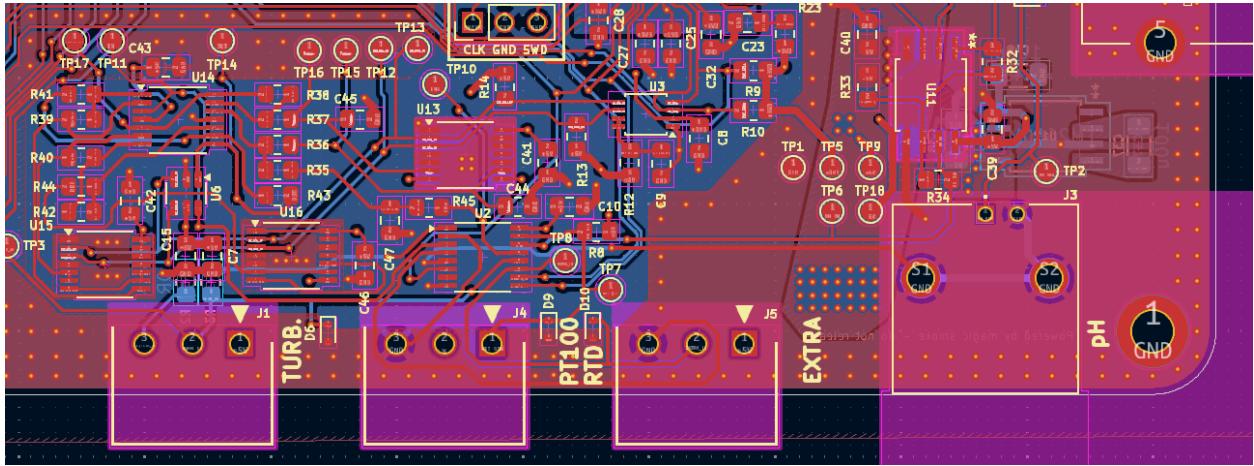


Figure 49: Signal Conditioner Subsystem PCB Layout

The final routing section is the RF interface, for which most of the components were managed by the integration of the Waveshare Core1262 LoRa Modulator Module. Key considerations in this section included maintaining 50Ω characteristic impedance along the coplanar waveguide, integrating pi filter footprints such that they have minimal impact on trace geometry, and utilizing fencing and stitching vias in this area as well as along the board edge to minimize interference caused by this section, shown in Figure 50 below.

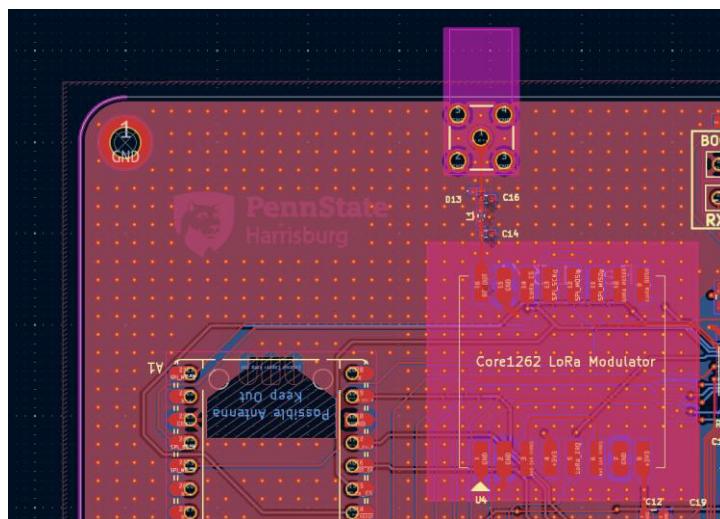


Figure 50: RF Frontend PCB Layout

Below are 2D plots and 3D renders of the final PCB design, as produced in KiCAD. In addition to the layout described in the previous section, a full Raspberry Pi Pico 2 Footprint was added to the design as a failsafe mechanism if any layout errors in the microcontroller system of the board would prevent the functioning of the design.

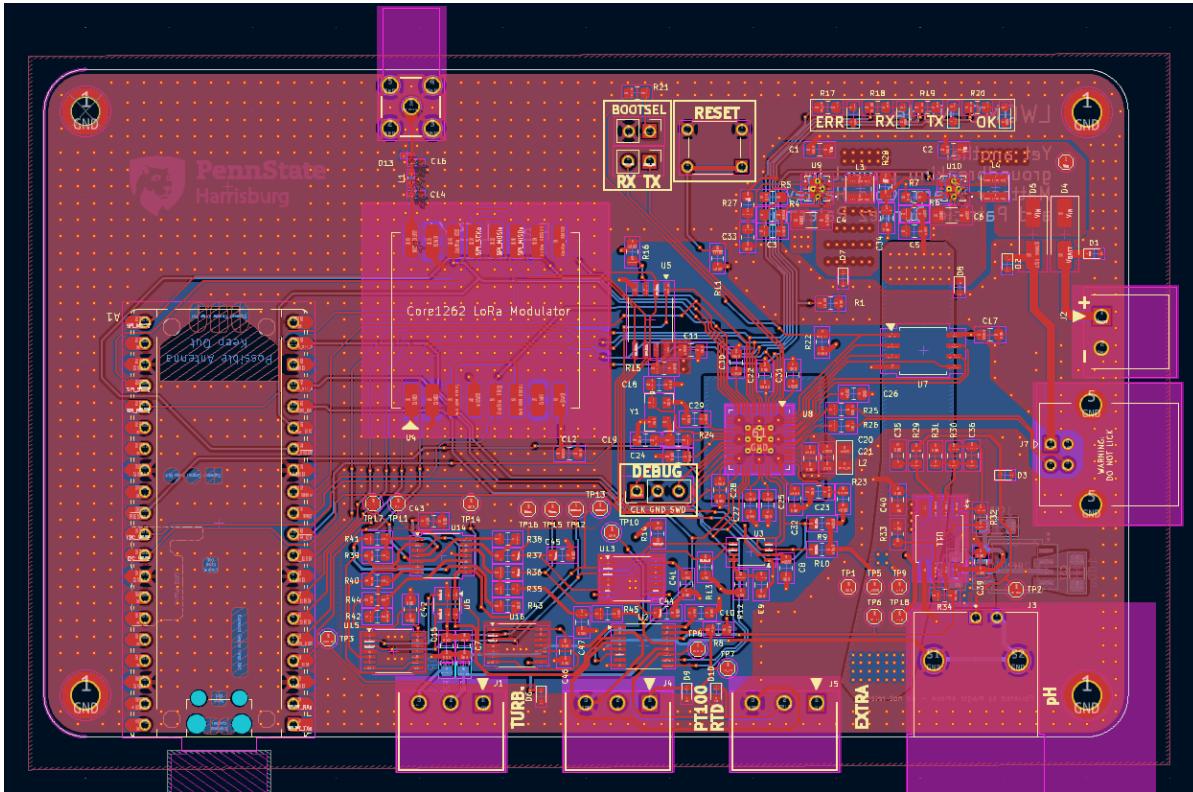


Figure 51: Full 2D Render of Design as produced in KiCAD.

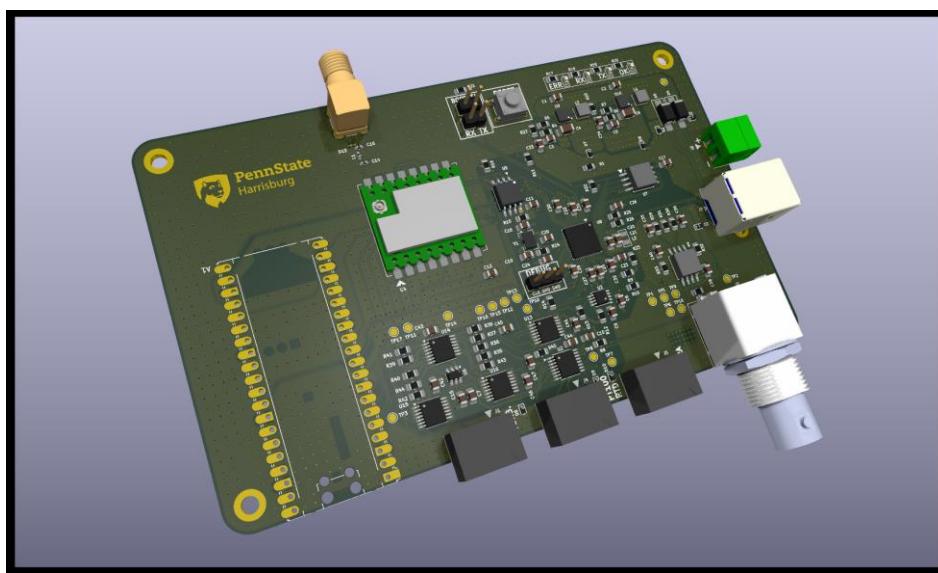


Figure 52: Full 3D Render of PCB, Populated

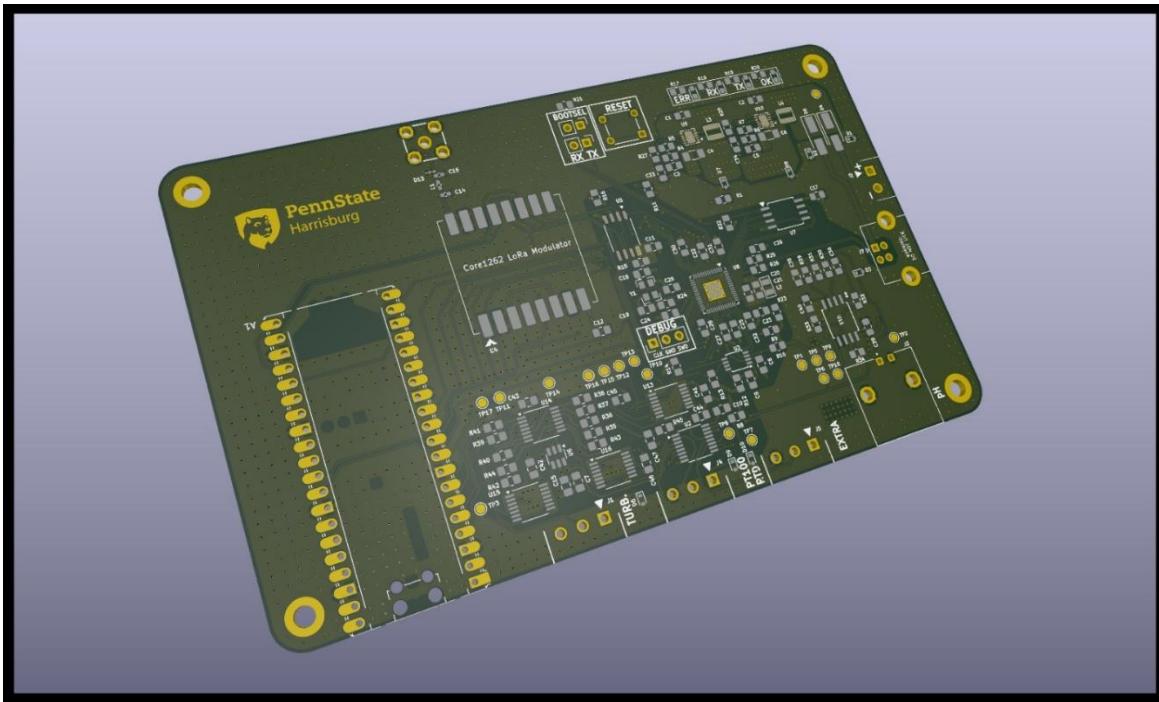


Figure 53: Full 3D Render of PCB, Unpopulated

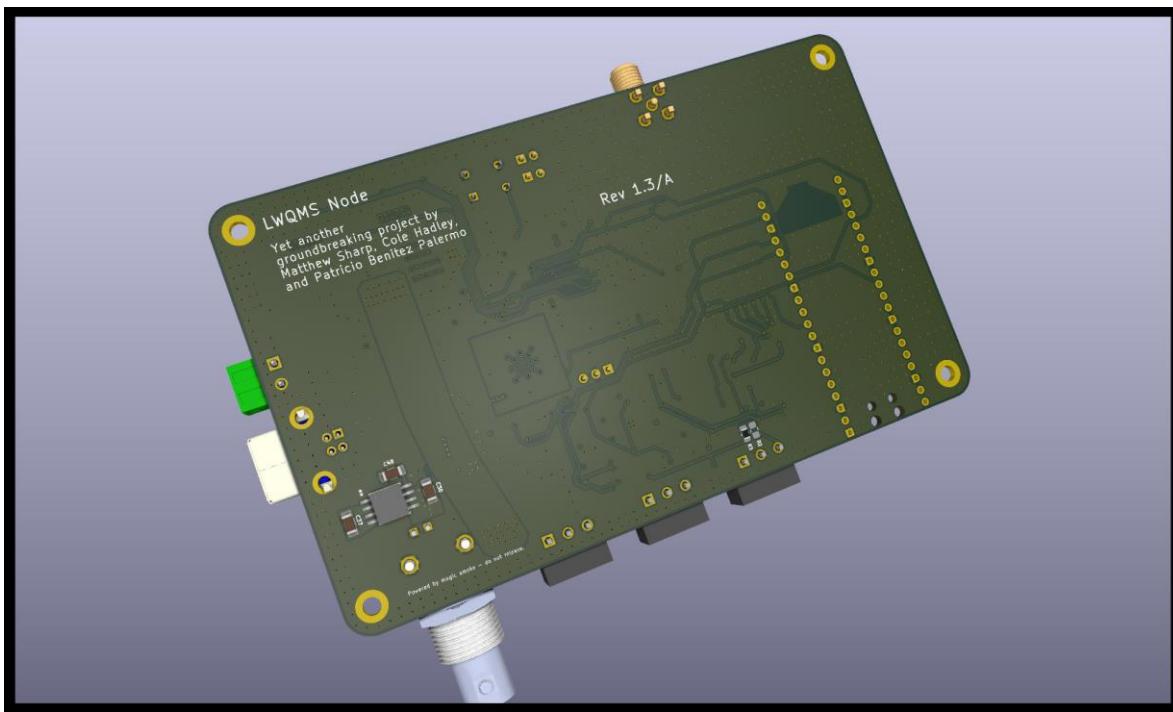


Figure 54: 3D Render of Reverse side of PCB, Populated

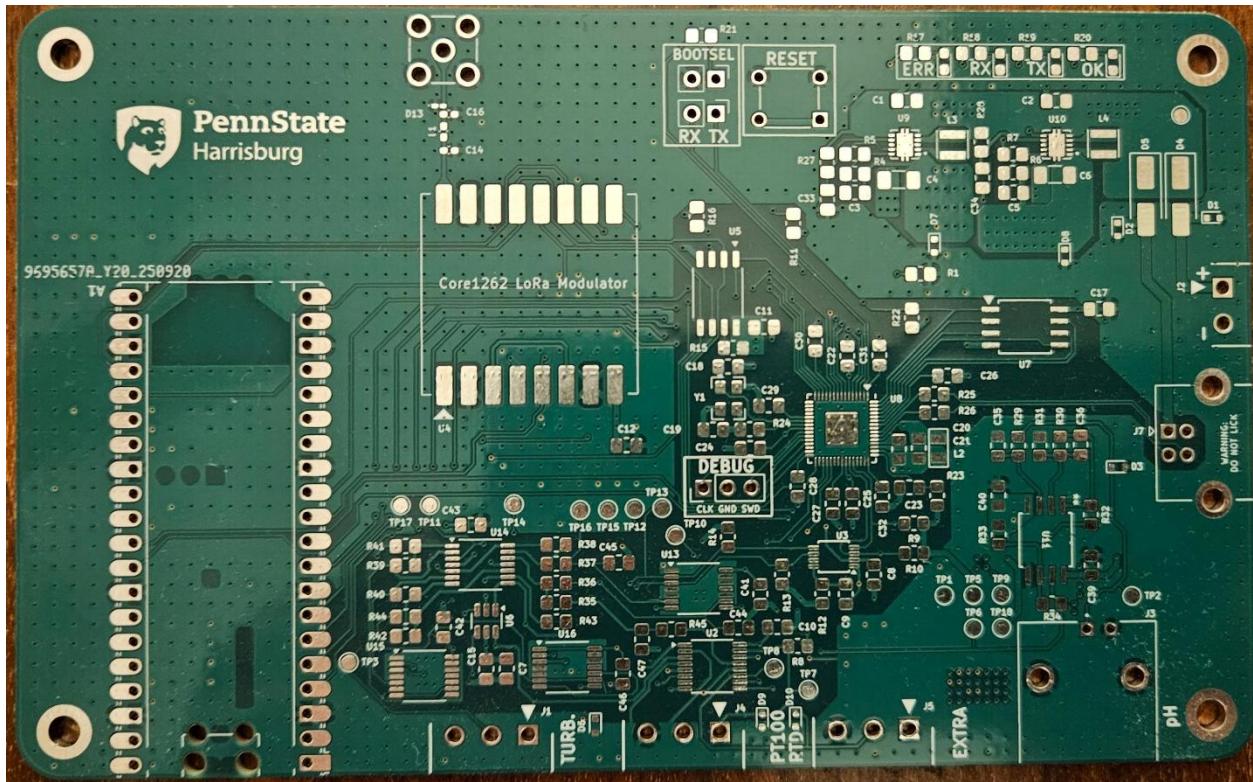


Figure 55: Final Ordered PCB, Unpopulated

With the PCB design received from the manufacturer, and all components ordered, the final populated PCB is shown in Figure 56 below. As seen in the figure, two routing errors were discovered during assembly and initial testing of the PCB. First, the IRQ and TX enable pins of the Core1262 LoRa modulator were reversed, resulting in the two jumpers seen in the figure being applied to the PCB. Second, the original PCB layout used the Texas Instruments LMV324 Operational Amplifier in single-supply operation. Further testing of the board revealed that this design was insufficient for analog stability, so the LMV324 was replaced with the TLV9144 operational amplifier in dual supply mode. To accommodate the dual-supply mode needed by the operational amplifier, another jumper wire was placed between the -5V test point on the PCB and the V_{SS} pin of the IC.

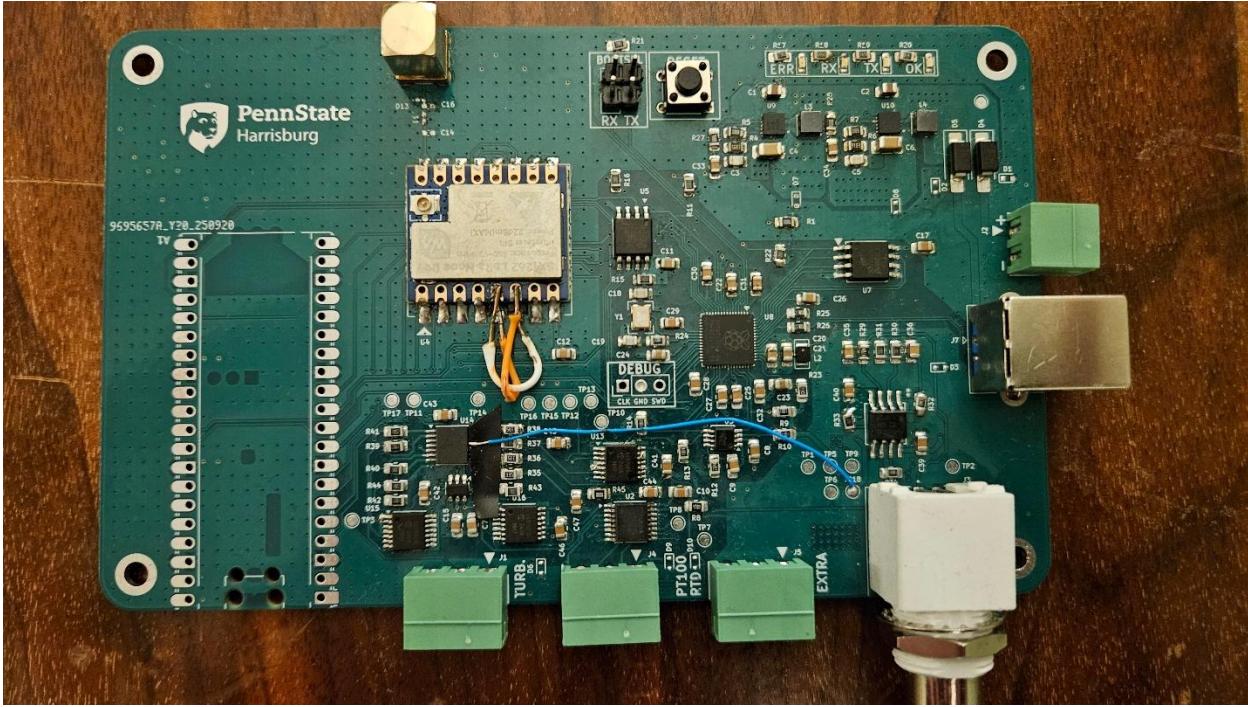


Figure 56: Final Ordered PCB, Populated

2.4.24. Sensor Node Power Management

To achieve the battery life goals established by our customer survey and formalized in our engineering requirements for the LoRa Water Quality Management System, careful attention needed to be paid to the power consumption of the sensor node. Since three D-Cell batteries placed in series were chosen to meet the engineering requirement that the device must be powered by readily commercially available batteries to produce the power rails for the device, the average current consumption of the sensor node must not exceed $608\mu A$ and $365\mu A$ to achieve a 3-year and 5-year battery life, respectively, assuming 16,000mAh of capacity [61]. A final design with a battery life in this range would cover more than half of survey responses. Table 21 and Table 22 below show the estimated current consumptions of the device in both the active and dormant states, respectively. Since two distinct rail voltages are used on the node to power all devices, only the total power consumption figure can be used to estimate the total battery life of the device.

Table 21: Sensor Node Power Consumption, Active State

Powered Devices in Active State	Est. Current Draw	Power Consumption
RP2350 Microcontroller [40]	30mA	99mW
SX1262 LoRa Modulator [48]	47.8mA	157.74mW
PE4259 RF Switch IC [62]	9µA	29.7µW
MX25L3233F Flash Memory [55]	10µA	33µW
W25Q128JVS1Q Flash Memory [56]	10µA	33µW
LSF0102 Level Translator [57]	6µA	19.8µW
TPS63000 Regulator (Quiescent, x2) [45]	90µA	405µW
TMUX1309 Analog Multiplexer [34]	1.5µA	7.5µW
TLV9144 Operational Amplifier [35]	28µA	140µW
MCP4651 Digital Potentiometer (x3) [36]	2.2mA	11mW
MCP3425 ADC [37]	155µA	775µW
TL082H Operational Amplifier [63]	2.3mA	11.5mW
Turbidity Sensor [51]	40mA	200mW
PT 100 RTD [52]	1mA	5mW
TOTAL		474.15mW

Table 22: Sensor Node Power Consumption, Dormant State

Powered Devices in Dormant State	Current Consumption	Power Consumption
RP2350 Microcontroller [40]	57µA	188µW
SX1262 LoRa Modulator [48]	0.16µA	0.528µW
PE4259 RF Switch IC [62]	9µA	29.7µW
MX25L3233F Flash Memory [55]	10µA	33µW
W25Q128JVS1Q Flash Memory [56]	10µA	33µW
LSF0102 Level Translator [57]	6µA	19.8µW
TPS63000 Regulator (Quiescent) [45]	90µA	405µW
TOTAL		709.028µW

Energizer shows a discharge curve for its D-Cell battery as shown in Figure 57 below for a battery discharging under a 10Ω load. For the purpose of estimation, we will assume an average D-Cell battery voltage of $1.15V$ over its entire lifetime.



Figure 57: Alkaline D-Cell Battery Discharge Curve under a 10Ω Load [61]

Since the Sensor Node is in active mode for approximately 2 seconds, with four active-mode wakes per hour, we can estimate the average current consumption of the node as shown in Equation 17 below.

Equation 17: Sensor Node Average Current Consumption

$$I_{avg} = \frac{t_{active}}{t_{interval}} \times I_{active} + \frac{t_{dormant}}{t_{interval}} \times I_{dormant}$$

Where $t_{active} = 12$ seconds, $t_{dormant} = 3,588$ seconds, and $t_{interval} = 3,600$ seconds. Since both power rails are produced by buck-boost regulators, we can estimate the current draw into each regulator using Equation 18 below.

Equation 18: Buck-Boost Regulator Input Current

$$I_{in} = \frac{P_{load}}{\eta V_{in}}$$

Assuming an efficiency (η) of 90% from the datasheet of the regulator [45], and input voltage of 3.45V for three D-Cell batteries in series, the active and dormant mode input currents can be estimated as $152.7mA$ and $228.4\mu A$, respectively. Using Equation 17 above, the average current draw of the node can be estimated as $567\mu A$, placing it within the 3–5-year range, meeting this customer requirement.

3. System Integration and Test Results

To thoroughly characterize the performance of the Sensor Node, Central Gateway, and Neural Network components of the LoRa Water Quality Management System, numerous tests were carried out, covering the individual integrated circuits used in the design, software components and handwritten libraries, subsystem behavior, and system-level performance.

A description of each test performed is provided over the following sections. In addition to the qualitative test setup and results provided here, comprehensive quantitative test result tables are provided in Appendix D.

Since this project is centered around a microcontroller, many of the tests performed to validate the performance of the system at the component level required a firmware component to setup ICs or take measurements. For each of these tests, the highest-level firmware file, which contains the entry point for the program, main(), is shown for the test. However, this code relies extensively on the software framework and drivers developed for the project, utilizing custom-built device drivers and hardware abstraction layer (HAL) calls, which integrate the hardware-level RP2350 C/C++ SDK with additional logging and device management functionality. The testing code and full software library used for the testing code can be found in the GitHub repository linked in Appendix C.

3.1. Digital Potentiometer Operation

To ensure functionality of the software-defined instrumentation amplifier component of the signal conditioner subsystem, one of the key component-level tests was validating the performance of the digital potentiometers which would produce the DC levels and gain resistance values to autonomously process incoming analog signals. To perform this test, the digital potentiometer was connected to a RP2350 microcontroller, integrated on a Raspberry Pi Pico 2, over I²C through the LSF0102 logic level translator, enabling bidirectional communication between the microcontroller's 3.3V logic domain and the digital potentiometer's 5V logic domain. The code shown in Figure 58 below was then used to validate the performance of both wipers on the MCP4651, and wiper voltage

measurements taken on a Keithley 2000 DMM were then compared against theoretical expectations.

```
#include "main.h"

static uint16_t wiper_positions[6] = {0, 50, 100, 150, 200, 256};

int main() {
    init_usb_console_hal();

    // Wait for the USB console to be opened on the host PC
    wait_for_usb_console_connection_hal();

    printf("Initializing hardware [HAL]...");
    i2c_init_hal(&context_i2c_1);    //

    printf("DONE\n");

    int wiper_position_index = 0;
    uint16_t wiper_position;

    while (true) {
        wiper_position = wiper_positions[wiper_position_index];

        printf("To advance the wiper position, press 't'. Next position: %d\n", wiper_position);

        // Wait for input
        while ((usb_console_getchar_hal() | 0x20) != 't') {} // t, case insensitive

        usb_console_write_hal("Setting...");

        // Set the wiper position
        mcp4651_set_wiper(&context_digipot_offset, MCP4651_WIPER_A, wiper_position);

        printf("DONE\n\n");

        wiper_position_index++;

        // Wrap around the index if it is out of bounds.
        if (wiper_position_index > 5) wiper_position_index = 0;
    }
}
```

Figure 58: Digital Potentiometer Testing Code

The full quantitative test results for the MCP4651 digital potentiometer are shown in Appendix D, Test #01. Potentiometer wiper voltage levels observed were highly consistent with expectations, as all wiper voltage levels were within 0.5% of theory. Despite the consistency of the wiper voltage levels with theory, this was later found to be a result of well-matched resistors implemented on-chip, rather than the potentiometer's resistance being highly consistent with the nominal datasheet value of $50\text{k}\Omega$. Further testing of the MCP4651 would find potentiometer resistances in the $35\text{k}\Omega - 50\text{k}\Omega$ range, which was not

captured by this test, as wiper voltages are representative of the matching of the resistive network, not the total resistance.

3.2. Level Translator Operation

As outlined in prior sections of this report, the RP2350 microcontroller operates within a 3.3V logic domain, which is incompatible with the analog frontend's supply voltage of 5V. As a result, direct connection between the I²C pins of the microcontroller and the digital potentiometers or ADC would not be possible. As such, the LSF0102 logic level translator was incorporated into the design. Since I²C uses open-drain output pins with pull-up resistors to create a wired-and connection on the serial data and clock lines, pull-up resistance value selection is critical to reliable operation of the I²C bus.

The LSF0102 datasheet provides a relationship for determining the minimum pull-up resistor value for use with the device, given in Equation 19 below.

Equation 19: Minimum Pull-Up Resistor Value for Logic Level Translation [57]

$$R_{PU} = \frac{V_{PU} - 0.35V}{15mA}$$

For the 5V side of the translator, this gives a minimum resistance value of 310Ω. However, such a low pull-up resistance would draw over 15mA of current from the power supply, when the bus is pulled low, greatly increasing power consumption of the node. As such, the pull-up resistance must be sized as large as possible to reduce power consumption of the node while also facilitating reliable data transmission through the level translator IC. Due to capacitances internal to the ICs on the I²C bus, as well as parasitic capacitances from the PCB layout along the I²C bus, rise time degradation is commonly observed when the pull-up resistance is oversized. Thus, testing was required to identify the largest allowable pull up resistance while also minimizing the rise time to within 10% of the signal's period at both the 100kHz and 400kHz standard-mode and fast-mode I²C clock frequencies.

As shown in Figure 59 below, oversizing the pull-up resistance greatly degrades signal rise time due to parasitic capacitances on the I²C bus, compared to Figure 60, which saw minimal rise-time degradation with appropriately-sized pull-up resistors.

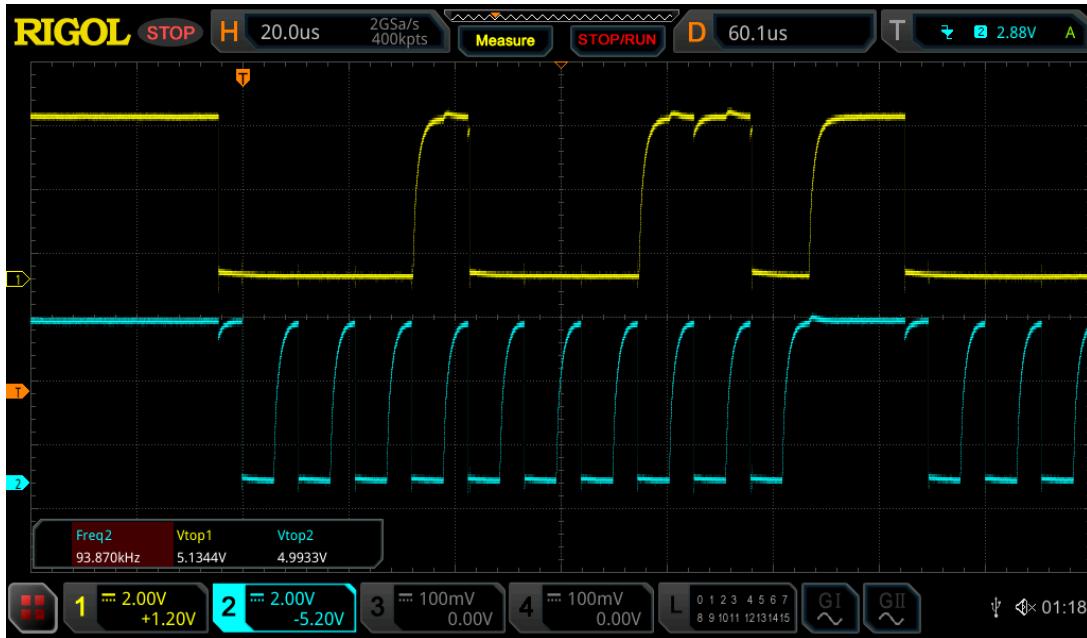


Figure 59: Serial Data Transaction over I²C with Oversized Pull-Up Resistors

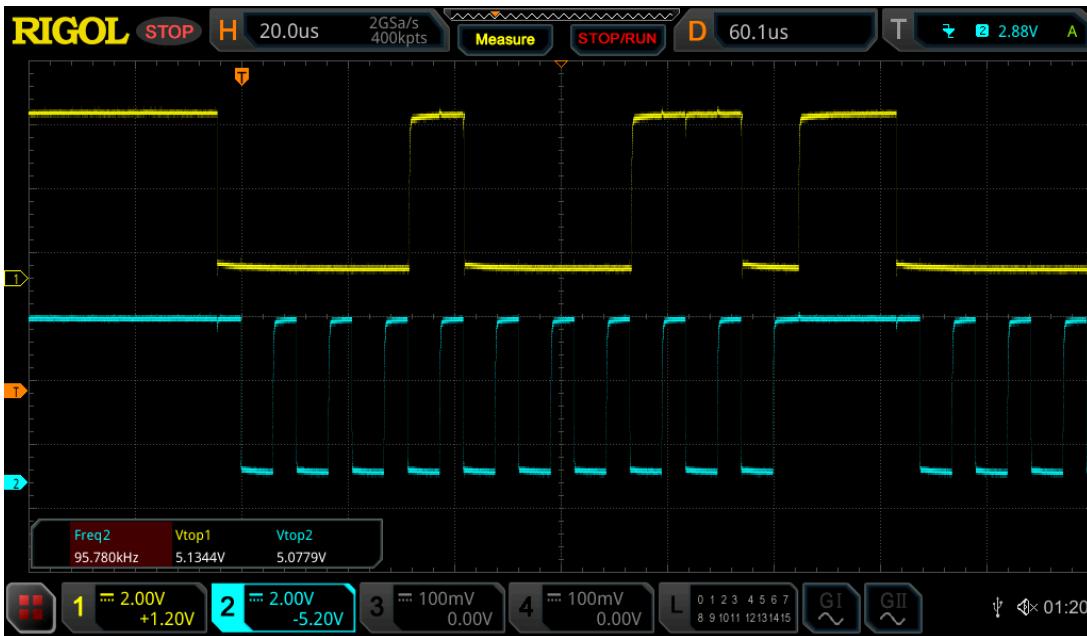


Figure 60: Serial Data Transaction over I²C with Well-Sized Pull-Up Resistors

Though the 30%-70% rise-time of the I²C data transaction is constrained to 1000ns and 300ns representing 10% and 12% of the period for the standard (100kHz) and fast (400kHz) modes, respectively [64], the test criterion was set to exceed the minimum requirement for data communication to account for unforeseen issues in the channel that may cause further rise time degradation beyond what was observed in the basic test setup.

Using a 10%-90% rise time criterion of 10% of the period, the $1k\Omega$ pull-up resistor was the only one which facilitated both data transmission through the channel while also meeting the rise time requirements for both logic levels for both the standard and fast mode. A full test report can be found for Test #02 in Appendix D.

3.3. Dual 4:1 Analog Multiplexer Operation

Another key component within the signal conditioner subsystem is the dual 4:1 analog multiplexer, which switches between differential input signals for use with the software-defined instrumentation amplifier. By multiplexing the sensor input signals, the analog frontend complexity is significantly reduced, as only one amplifier design needs to be implemented for all sensors used with the node.

To validate the functionality of the selected TMUX1309 multiplexer, four single-ended input signals were applied on the positive ends of the four input channels, while the negative ends were tied to ground (Test #03 in Appendix D). The input signals tested were sinusoidal, square, sawtooth, and DC, and the four inputs were selected using a Raspberry Pi Pico driving the output select pins of the device. The noise floor of the multiplexer and Rigol MSO5104 Oscilloscope were then captured, and to validate analog performance of the device, the attenuation and SNR of the pass-through signals were compared against a tolerance band of less than approximately 5% attenuation and >60 dB SNR. As shown in Figure 61, the SNR of the multiplexer and oscilloscope were measured as $179\mu V$ RMS. Figure 62 below shows one example of a test measurement through the multiplexer, with the yellow signal showing the output of the function generator and the blue signal showing the output through the multiplexer. The sinusoidal signal shown below did encounter a 5.2% attenuation through the multiplexer, the most seen of any of the tests performed.

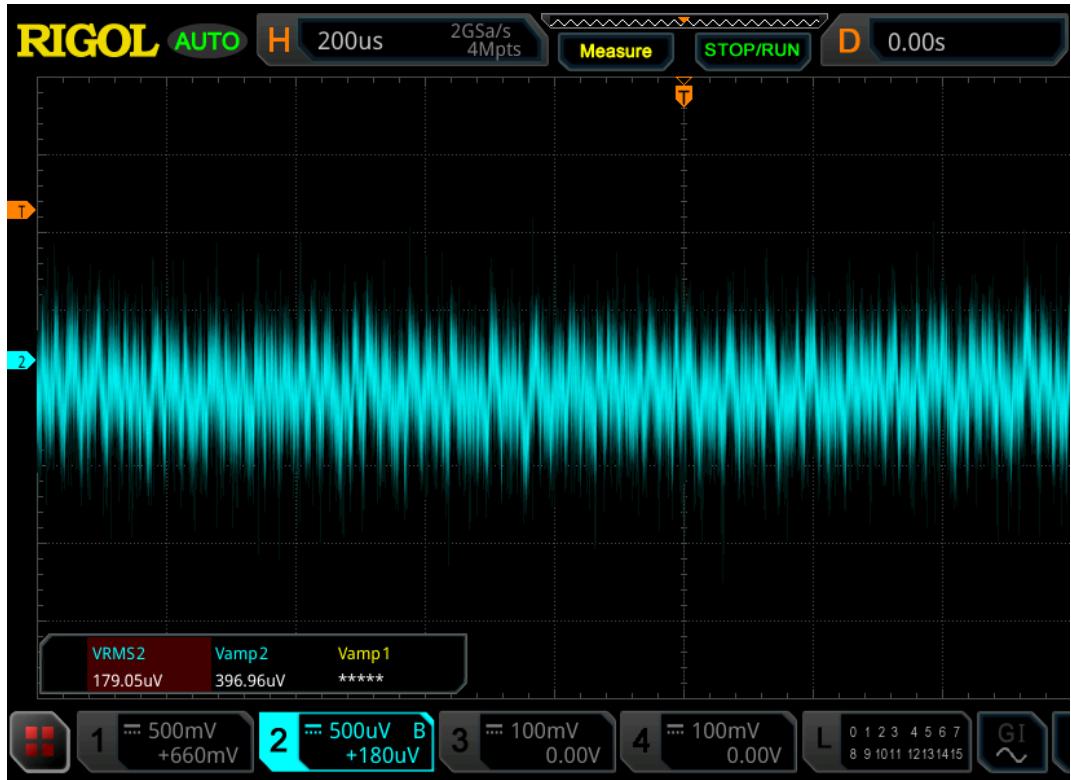


Figure 61: Noise Floor of the TMUX1309 and Oscilloscope

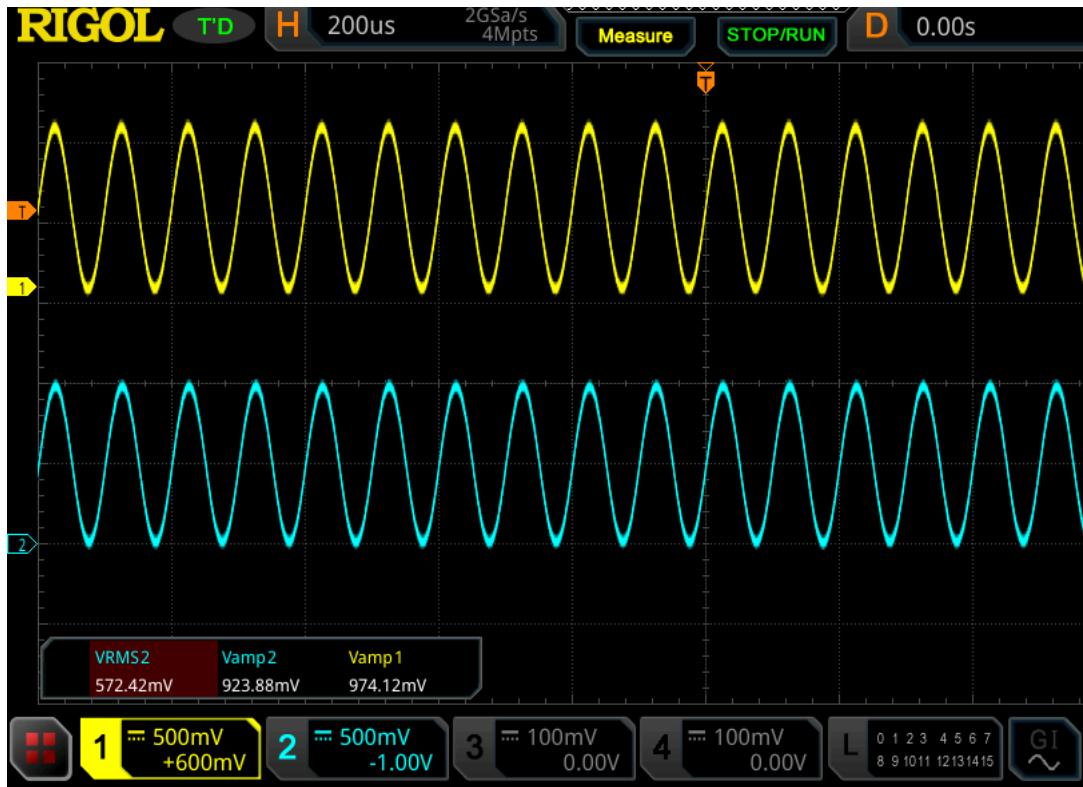


Figure 62: Attenuation of Sinusoidal Signal through TMUX1309

3.4. Switched Capacitor Operation

To validate the generation of the -5V rail for use with the analog frontend of the sensor node design, a full functional test of the LM2660 Switched Capacitor Voltage Converter was performed in the lab. The LM2660 IC was placed on a breadboard though an SMT-to-THT breakout PCB. Three $100\mu F$ capacitors were used in the implementation of the IC, in an identical arrangement to the reference design shown in Figure 24. A varying load resistance was placed between the negative output and ground to observe the operation of the regulator through approximately 10mA, well exceeding the combined current draw of the TLV9144 and TL082 operational amplifiers [35], [63]. As seen in Test #04 in Appendix D, the LM2660 was able to maintain the magnitude of its negative output within 95% of the input voltage across this entire range.

3.5. Buck-Boost Regulator Operation

For the functional test of the buck-boost regulator, a simple test of the IC would be insufficient, as the TPS63000 requires a specific layout to be fully functional. Since the TPS63000 uses a Very Thin Small Outline No-Lead (VSON) package as shown in Figure 63, and relies upon careful routing of inductor and feedback path traces for full operation, the test for this device needed to be performed using a custom PCB layout produced using KiCAD [45]. The schematic implementation of the custom evaluation PCB largely mirrors the implementation used on the sensor node PCB. Though the PCB layout for the design follows the recommended layout from the manufacturer's and the layout used in the final PCB design for the critical inductor traces, fabrication limitations introduced by University's PCB mill required some modifications to the feedback path of the test PCB. As seen in the schematic and PCB for the custom evaluation module, shown in Figure 64 and Figure 65, respectively, surface-mount test points were used along the feedback path between R_4 and R_5 . Since manufacturing complexity would be greatly increased by the introduction of vias and a bottom layer, the trade-off was made to instead use a jumper wire between the two test points to complete the feedback path to the IC. This ultimately did introduce instability that was not observed in the final PCB, which followed

manufacturer layout recommendations. A final 3D render of the evaluation PCB, produced in KiCAD, is shown in Figure 66 below.

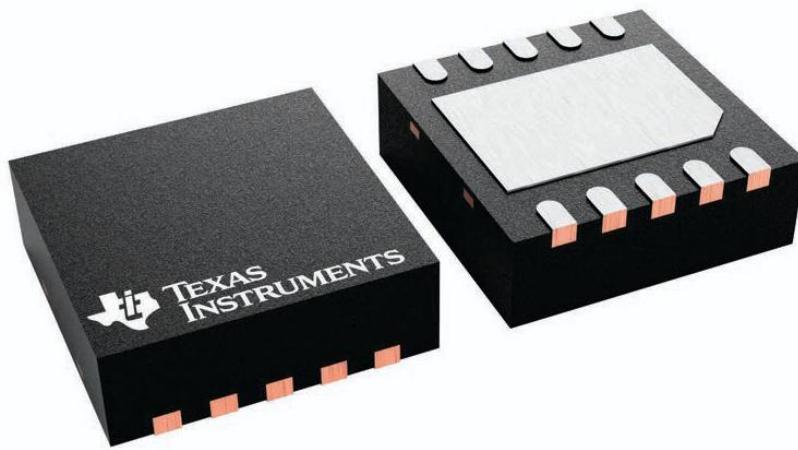


Figure 63: TPS63000 Very Thin Small Outline No-Lead (VSON) Package [45]

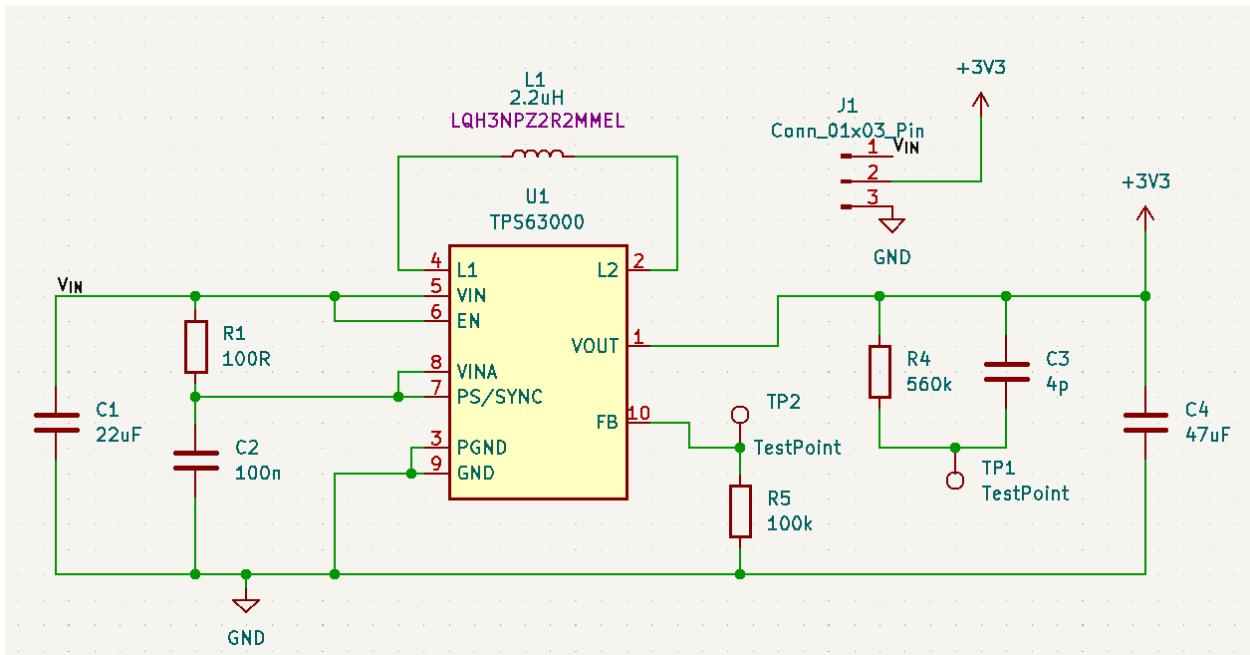


Figure 64: Schematic for the TPS63000 Custom Evaluation Module

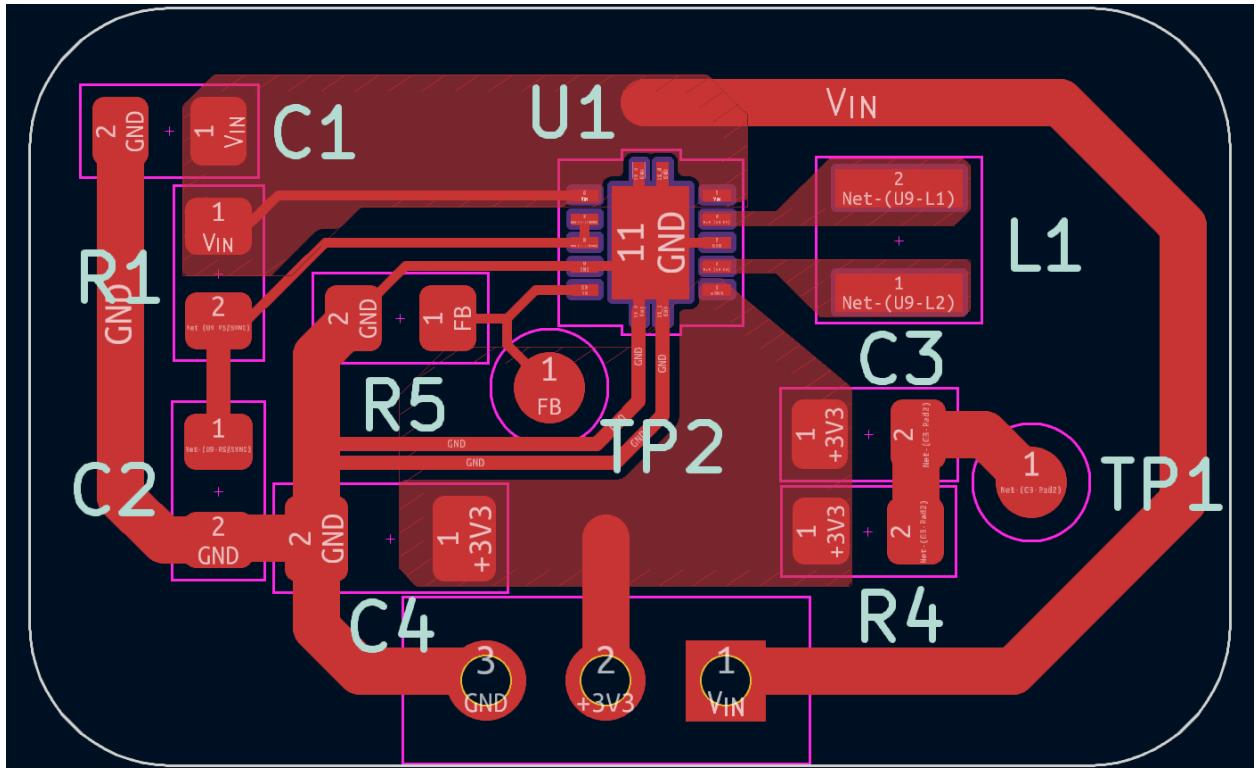


Figure 65: PCB Layout for the TPS63000 Custom Evaluation Module

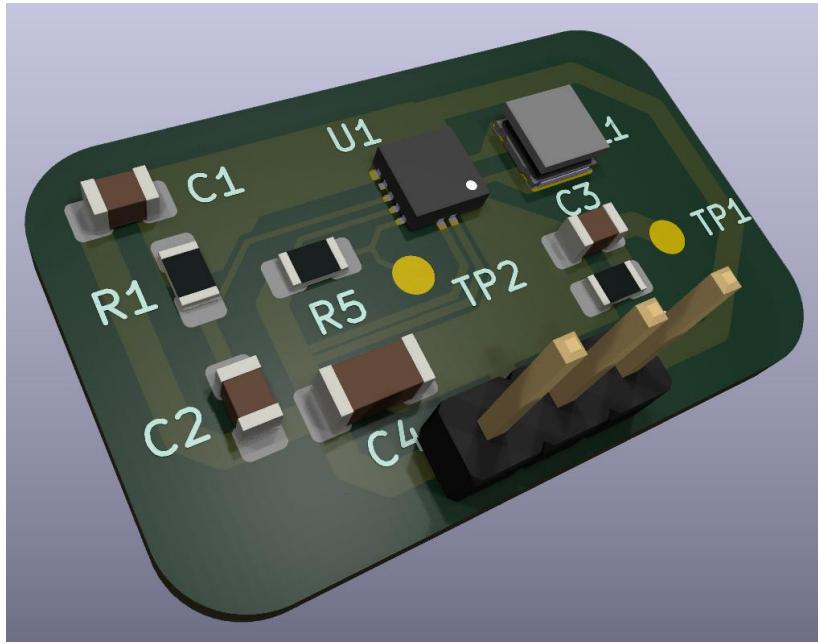


Figure 66: 3D Render of the TPS63000 Custom Evaluation Module

The PCB was then fabricated using the University's PCB milling machine and assembled with all required components. Testing was first performed for the +3.3V output using the

feedback path component value selections determined in Section 2.4.8, as shown in Table 23. With all testing complete, the feedback path components were replaced with those needed to produce a +5V output. In both cases, the input voltage was varied from 1.8V to 5.5V to verify the regulator would operate as expected in both the buck and boost modes of operation. The input voltage was varied using a Rigol DP832 3-Output Variable DC power supply, and the regulator output voltage was monitored using a Keithley 2000 DMM. A picture of the test setup and final assembled custom evaluation module is shown in Figure 67 below.

Table 23: Component Value Selections for TPS63000 Evaluation Module Feedback Path

Output Voltage	R_1	R_2	C_{ff}
+3.3V	$560k\Omega$	$100k\Omega$	$4pF$
+5.0V	$910k\Omega$	$100k\Omega$	$2.4pF$

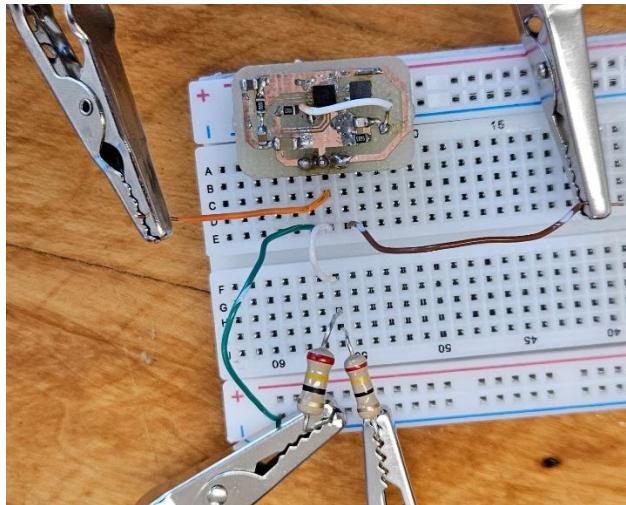


Figure 67: Final Assembled TPS63000 Evaluation PCB In Test Setup

As outlined in Test #05 in Appendix D, output voltages in both cases remained within the acceptable error tolerance range of 5%. Figure 68 below shows the observed input-output voltage relationship observed for the TPS63000 buck-boost regulator, demonstrating the ability of the IC to produce stable voltage rails along the entire discharge curve of the alkaline power supply batteries.

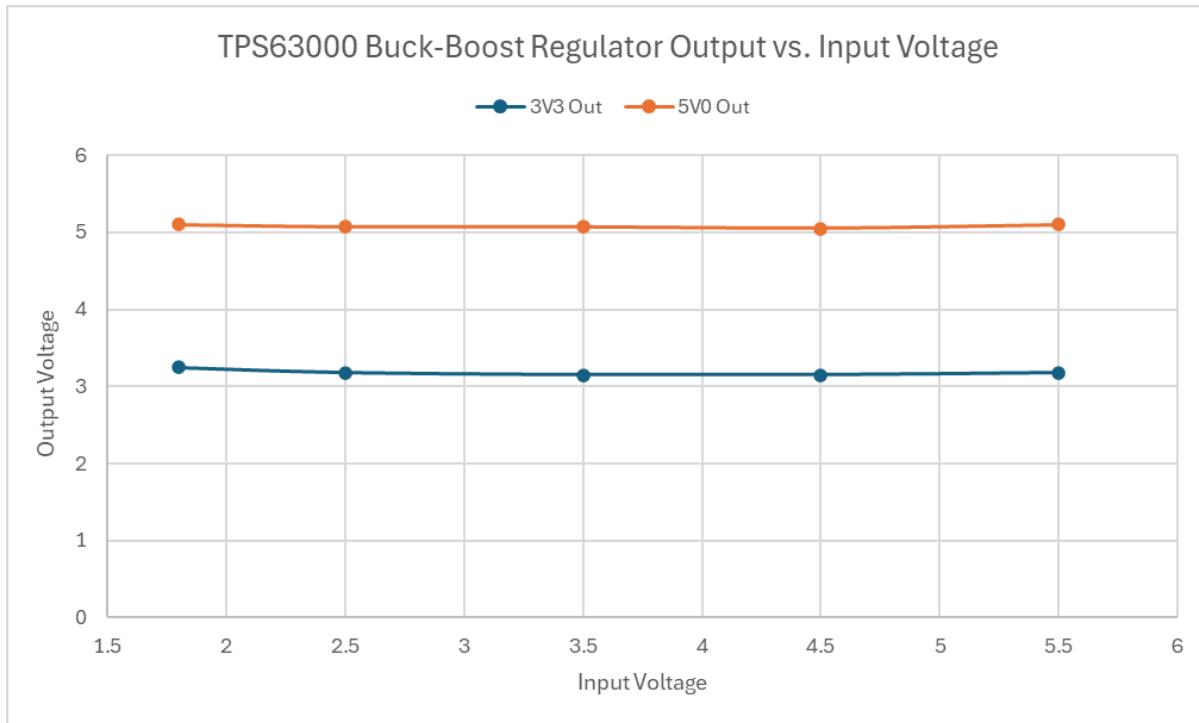


Figure 68: TPS63000 Regulator Output vs. Input Voltage

3.6. 16-bit Analog-to-Digital Converter Test

Though the theoretical accuracy of the MCP3425 ADC IC exceeds that of the RP2350 by nearly two times, as outlined in section 2.4.14, the accuracy of the IC still needed to be verified against professional measurement equipment. For this test, the input of the MCP3425 was driven by one of the channels of the Rigol DP832 3-Output Variable DC power supply, and the voltage into the device was varied across the ADC's entire input range. The actual voltage seen at the ADC input, as measured using a Keithley 2000 DMM, was compared against the reported voltage of the ADC. As shown in Test #06 in the appendix, the actual voltage and ADC reported voltage were nearly identical, with about 0.2% error in every case. As shown in Figure 69, the error between the ADC's reported voltage measurement and the DMM reading was nearly constant for all test cases, within 0.25% for every voltage tested, demonstrating this IC's value in producing an accurate measurement for use with the sensors on the node.

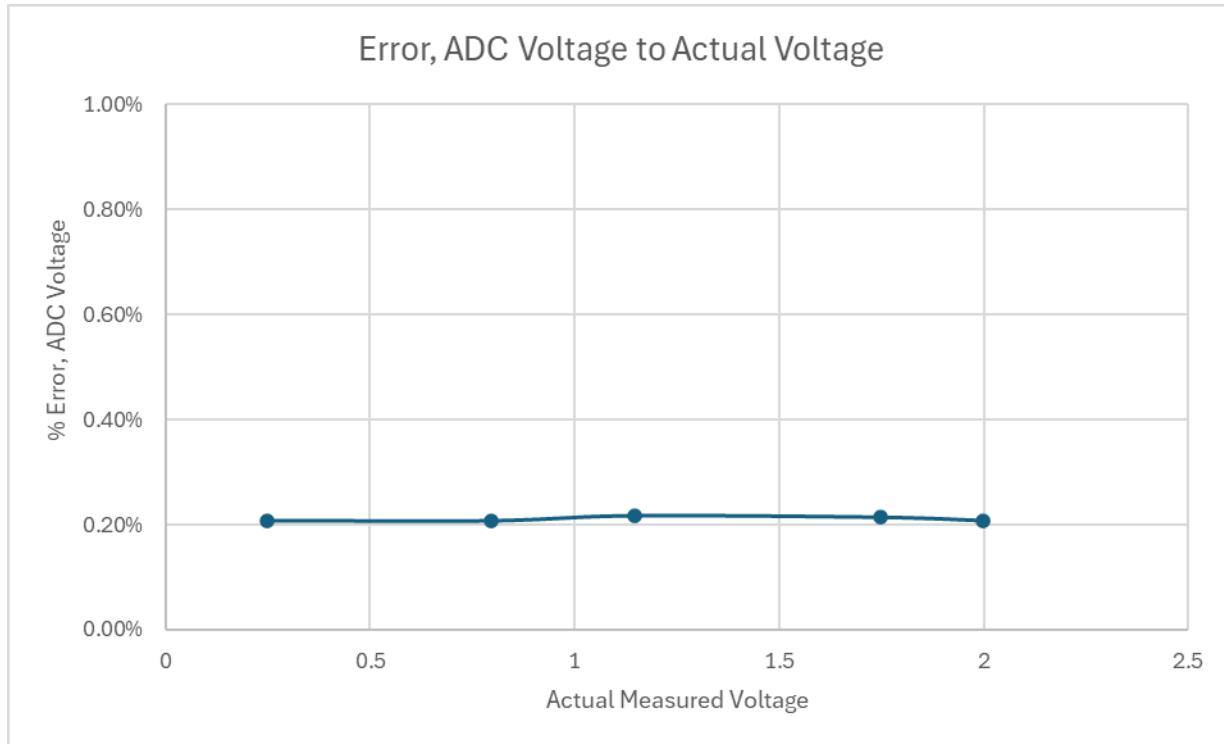


Figure 69: MCP3425 ADC Measurement Error

In addition to a test of the functionality of the IC's hardware, this test also served as a validation for the custom-written driver for the MCP3425 for this project. Like prior tests relying upon project firmware code, the high-level driver code used for this test is shown in Figure 70 below, however the calls to the HAL for initialization and the custom-built MCP3425 library rely upon code provided in the GitHub repository linked in Appendix C.

```

#include "main.h"

/**
 * @brief Full functional test of the MCP3425 Analog-To-Digital Converter (ADC).
 *
 * @remark Code for the MCP3425 library is in mcp3425.c in the lib/ directory.
 *
 * @author Matthew Sharp
 */

int main()
{
    init_usb_console_hal();

    // Wait for the USB console to be connected
    wait_for_usb_console_connection_hal();

    sleep_ms(500);

    print_banner();

    // Initialize the gpio pins
    printf("Initializing hardware...");

    i2c_init_hal(&context_i2c_1);

    // Set to one-shot conversion mode, 16 bit resolution, and a PGA gain of 1.
    mcp3425_init(&context_adc_0, MCP3425_SPS_15_16BITS, MCP3425_PGA_1, false);

    bool configured_correctly = (!context_adc_0.continuous_conversion_mode_enabled) &&
    (context_adc_0.sampling_rate == MCP3425_SPS_15_16BITS) && (context_adc_0.gain == MCP3425_PGA_1);

    if (!configured_correctly) {
        printf(
            "Failed to configure ADC. Received one-shot, sampling, and gain values of: %d, %d, %d\n",
            context_adc_0.continuous_conversion_mode_enabled,
            context_adc_0.sampling_rate,
            context_adc_0.gain
        );
        while (1) {} // Idle
    }

    printf("DONE\n");

    while (1) {

        printf("To take a reading, press 't'.\n");

        // Wait for input
        while ((usb_console_getchar_hal() | 0x20) != 't') {} // t, case insensitive

        double voltage = mcp3425_get_measurement(&context_adc_0);

        printf("Reading: %f\n", voltage);
    }
}

```

Figure 70: MCP3425 ADC Test Code

3.7. SPI Flash Operation

Another IC used on the sensor node is the Macronix MXL25L3233F 32Mbit NOR Flash IC with SPI interface. This test, listed under Test #07 in Appendix D, served both as a test for the IC and the custom firmware driver written for it for use with this project. This test validated the functionality of the IC's read, write, and erase modes, as well as the IC's deep power down mode. Since this test's code is more intensive than other hardware-software integration tests and thus too long to include directly in this report, the full driver code for this test can be found in `spi_flash.c` under the `tests/` directory in the GitHub repository linked in Appendix C.

Since the MXL25L3233F is a flash memory device, data can not be written to it in the same way as an MCU might interface with RAM; byte-by-byte writes are not possible with the device. Rather, an entire 4kB sector must be erased at minimum and then re-written with new data page-by-page in 256-byte blocks. For this reason, the test was conducted first with a full-chip erasure, then a write of dummy data, readback of the dummy data, and then the test of the low power mode of the device. As shown in Figure 71, chip erasure was successful, as all memory bytes on the IC were set to `0xff`. From there, test data was written to the IC and read back to validate the read and write operations, as shown in Figure 72. As seen in this figure, Abraham Lincoln's 1863 Gettysburg Address was used as it provided sufficiently long plain text data for large data transfers between the microcontroller and flash IC [65]. Finally, the chip's deep power-down state was validated, as the chip entered a lower power consumption state, and would only respond to the wake command before resuming full regular operation.

Figure 71: Flash Memory Dump of First 256-Byte Page after Chip Erasure

	Memory	Address	Offset	Data	ASCII	Equivalent											
00000000	46	6f	75	72	20	73	63	6f	72	65	20	61	6e	64	20	73	Four score and s
00000010	65	76	65	6e	20	79	65	61	72	73	20	61	67	6f	20	6f	even years ago o
00000020	75	72	20	66	61	74	68	65	72	73	20	62	72	6f	75	67	ur fathers broug
00000030	68	74	20	66	6f	72	74	68	20	6f	6e	20	74	68	69	73	ht forth on this
00000040	20	63	6f	6e	74	69	6e	65	6e	74	2c	20	61	20	6e	65	continent, a ne
00000050	77	20	6e	61	74	69	6f	6e	2c	20	63	6f	6e	63	65	69	w nation, concei
00000060	76	65	64	20	69	6e	20	4c	69	62	65	72	74	79	2c	20	ved in Liberty,
00000070	61	6e	64	20	64	65	64	69	63	61	74	65	64	20	74	6f	and dedicated to
00000080	20	74	68	65	20	70	72	6f	70	6f	73	69	74	69	6f	6e	the proposition
00000090	20	74	68	61	74	20	61	6c	6c	20	6d	65	6e	20	61	72	that all men ar
000000a0	65	20	63	72	65	61	74	65	64	20	65	71	75	61	6c	2e	e created equal.
000000b0	20	4e	6f	77	20	77	65	20	61	72	65	20	65	6e	67	61	Now we are enga
000000c0	67	65	64	20	69	6e	20	61	20	67	72	65	61	74	20	63	ged in a great c
000000d0	69	76	69	6c	20	77	61	72	2c	20	74	65	73	74	69	6e	civil war, testin
000000e0	67	20	77	68	65	74	68	65	72	20	74	68	61	74	20	6e	g whether that n
000000f0	61	74	69	6f	6e	2c	20	6f	72	20	61	6e	79	20	6e	ation, or any na	

Figure 72: Flash Memory Dump of First 256-Byte Page after Write Operation

3.8. LoRa Modulator Test

This test, listed under Test #08 in the appendix, is jointly an evaluation of the physical hardware component of the Waveshare Core1262 LoRa Modulator as well as the software integration of the custom-built device drivers. Since the driver provided by Semtech only provides low-level hardware-level commands, the implementation of data transfer using LoRa still requires an additional software layer above the underlying hardware-level driver. Like the SPI Flash test, the testing code used for the LoRa modulator is too long to incorporate directly into this report. The full testing code for the LoRa modulator can

instead be found in `txrx.c` under the `tests/` directory within the GitHub repository linked in Appendix C.

In addition to hardware-level testing of the devices, the 915MHz band was simultaneously monitored using a Software-Defined Radio visualized within the SDR++ software package to validate that packet transmission was taking place according to the specified center frequency and bandwidth. As seen in Figure 73 below, the packet occupies frequencies from approximately 914.94MHz through 915.06MHz , giving it an estimated bandwidth of 120kHz , consistent with expectations.

Using the custom-built LoRa-level software layer for this project, transmit and receive operations were successfully implemented on the sx1262, and the packet data was successfully received without errors. Low-power mode testing was also performed on the module, however a sleep current of $31.6\mu\text{A}$ was observed on the module, nearly 200 times the $160n\text{A}$ listed current for sleep mode in the manufacturer's datasheet [48]. However, the Core1262 module integrates an RF switch into the design to support both transmit and receive operations through the same antenna; one such RF switch identical in operation to the one used lists a $20\mu\text{A}$ maximum quiescent current, likely accounting for this additional observed current draw into the module [62].

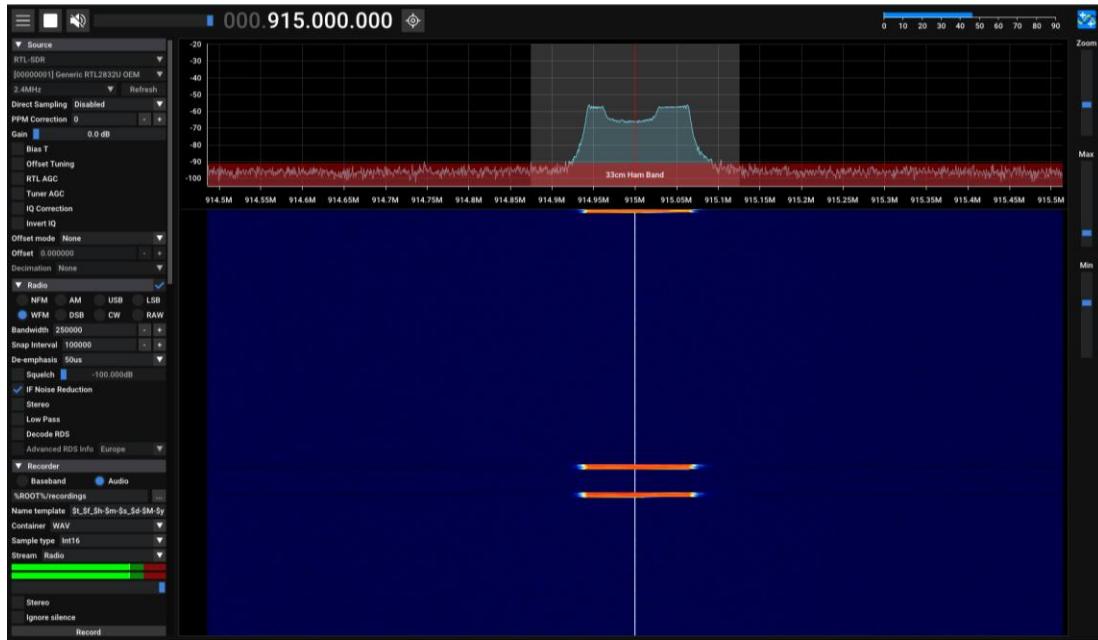


Figure 73: LoRa Packet Capture on SDR, as Produced in SDR++

3.9. LoRa Range Test

With the transmit and receive operations validated working in both hardware and software as detailed in Section 3.8, the range of the device was tested and compared against the theoretical transmission distance detailed in Section 2.4.10. Using the LoRa time-on-air calculator from [50], a preamble length, spreading factor, output power, coding rate, bandwidth of 8 symbols, SF10, +22dBm, 4/5, and 125kHz, respectively, produced a theoretical range of 4.73km. To validate this theoretical range, three range tests were performed in the Harrisburg, Pennsylvania area. Since LoRa requires line-of-sight between transmitter and receiver, the relatively hilly terrain made test site selection challenging. For this reason, the longest test performed is approximately 2.2km, about half of the theoretical range of the LoRa modulator with the above settings.

The three test sites are shown on the map in Figure 74 below, with each blue marker representing the location of the transmitter, with the receiver at the other end of each test's respective colored line.

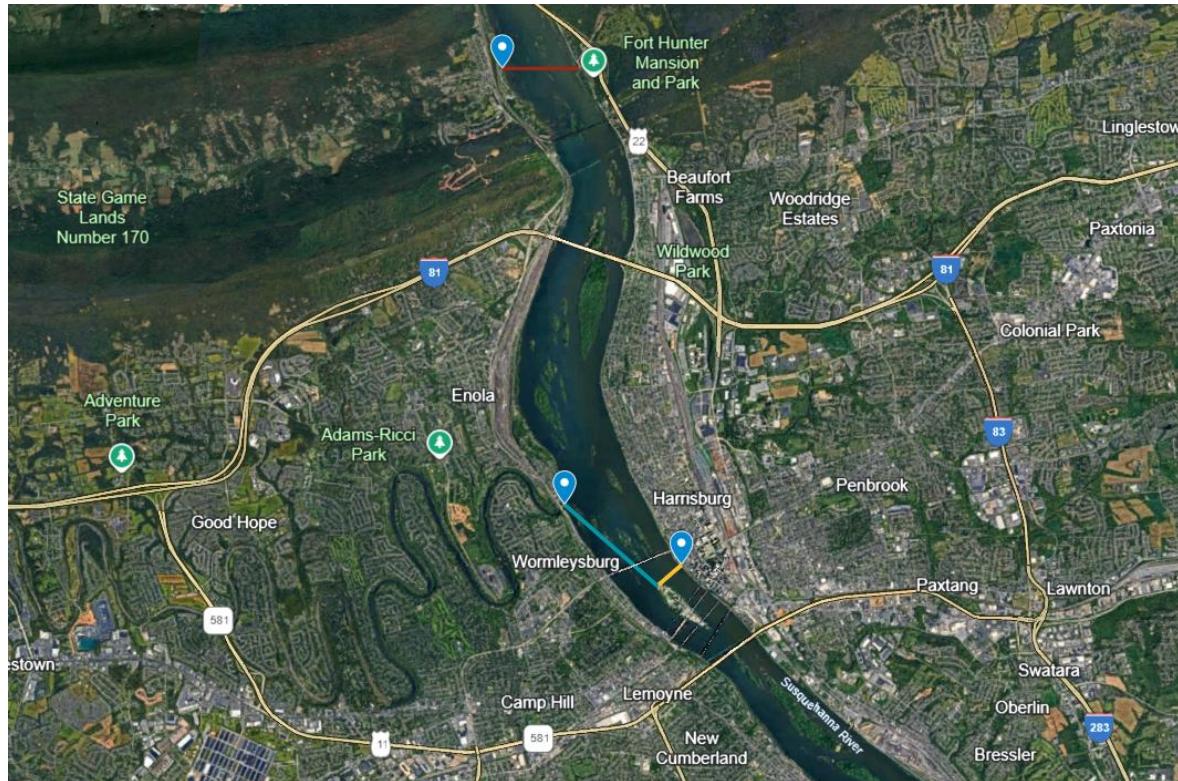


Figure 74: LoRa Range Testing Map, as Produced in Google Earth

In each test case, ranging from approximately 500 meters through 2.2km, (full results tabulated in Test #014 in Appendix D), successful packet transmission was achieved between the sender and receiver. However, when line-of-sight between the LoRa modules was not present due to a physical obstacle such as a building, transmission could not be achieved. This demonstrates a need for a more robust mesh networking architecture for a fully implemented system, as line-of-sight will not always be available between sensor nodes and the central gateway.

3.10. pH Sensor Pre-Amplifier Operation

A subsystem level operational test (Test ID #009) was performed for the pH sensor pre-amplifier circuit to verify the functionality of the TL082 JFET input op-amp for use with the DFRobot pH sensor. The pre-amplifier circuit shown earlier in Figure 17 was implemented on a breadboard. On the PCB design, a TL082 dual op-amp IC is specified; however, for this test, two TL081 single-op-amp ICs were used as a functionally equivalent substitute. The pH sensor output was connected to the input of the pre-amplifier, and the output of the second stage was measured with a Keithley 2000 digital multimeter.

The pH sensor was immersed in solutions with pH values of 4, 7, and 11, producing sensor output voltages of 170mV, 2mV, and -290V, respectively. After amplification, the measured outputs were 1.76V, 2.11V, and 2.56V, compared to the theoretical outputs of 1.66V, 2.00V, and 2.58V. All errors were below 6%, with a maximum difference of 5.8%, so the pre-amplifier was deemed to have passed the operational test. When the pre-amplifier is integrated on the sensor node PCB, software calibration and the downstream signal conditioner will further refine the pH measurement accuracy.

3.11. Turbidity Sensor Operation

A component level operational test (Test ID #012) was performed for the DFRobot analog turbidity sensor to verify its functionality and prepare it for integration with the sensor node. The circuit used on the sensor node PCB was implemented on a breadboard. The internal IR phototransistor of the sensor is connected in series to an external $4.7\text{k}\Omega$ resistor. With a 5V supply, the voltage across the resistor was measured using a Keithly 2000 digital multimeter and used as the sensor output for determining turbidity.

Different turbidity solutions were prepared using kaolin clay and water. The actual turbidity of each solution was measured using a Hach 2100p turbidimeter. The resulting solutions created were: 0.2, 7, 27, 60, 140, and 330 NTU. Attempts to prepare kaolin-water solutions greater than 300NTU were unstable NTU readings on the Hach turbidimeter. The solutions' turbidity reading would drop quickly after mixing, so each solution was measured with the turbidimeter and then immediately transferred into the DFRobot sensor so that the sensor output could be recorded.

For each turbidity level, the initial output voltage was taken as the primary voltage-to-NTU relationship, which showed the voltage decreases as NTU increases. Over 0.2-140 NTU, the initial output dropped from 4.118V to 3.996V (a 122mV swing). In addition, the output voltage for each solution was measured over a 2-minute window. The measured voltage did not remain constant; instead, it fluctuated and showed an overall gradual decline, likely due to particle settling and the corresponding reduction in NTU. This behaviour also highlights the limited stability and precision of the low-cost sensor. At 330 NTU, the voltage changed the most over the 2-minute window and the readings were deemed unstable. Therefore, the practical operating range for this sensor in the current design was taken to be approximately 0-100 NTU.

3.12. RTD Test

A component level operational test (Test ID #015) of the PT100 RTD temperature sensor was performed to verify its functionality and prepare it for integration with the sensor node. The circuit used on the sensor node PCB was implemented on a breadboard. As described in the subsystem section, a 5V source is connected to a $4.7\text{k}\Omega$ resistor in series with the RTD, which is then connected to ground, forming a voltage divider.

A beaker was filled with water of ideal temperatures of 0°C , 25°C , and 100°C to cover the desired measurement range. Using a digital thermometer, the actual water temperatures were measured to be 3°C , 22°C , and 96°C , respectively. For each temperature, the RTD was placed into the beaker, and the voltage at the node between the resistor and RTD was measured with a Keithly 2000 digital multimeter.

The measured voltages were then compared to the theoretical outputs calculated using the CVD resistance–temperature relationship. At 3°C, 22°C, and 96°C the theoretical outputs were 105.36 mV, 112.89 mV, and 141.61 mV, while the measured outputs were 107.68 mV, 114.61 mV, and 143.72 mV, respectively. The test was deemed successful, resulting in less than a 3% difference between the measured and theoretical output. Therefore, the RTD and associated circuit are considered suitable for use in the sensor node.

3.13. Software-Defined Instrumentation Amplifier Test

To account for nonlinearities and resistance deviations between digital potentiometers discovered during initial testing, an automatic calibration routine was incorporated into the sensor node’s firmware. The calibration measures the DC offsets and Gain Resistor values at each potentiometer wiper step location (0-256). This data is then stored on the external flash IC, which the RP2350 microcontroller then reads into memory, using it to accurately set the digital potentiometers according to desired gain and DC offset values. Plots of the DC offset and gain resistance values, as well as corresponding error are shown in the figures below.

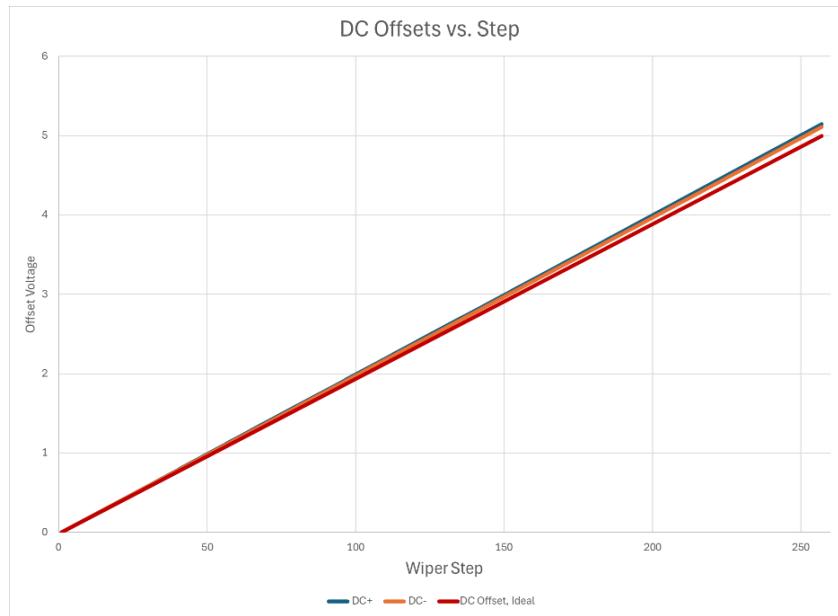


Figure 75: DC Offset Voltages versus Potentiometer Wiper Step Location

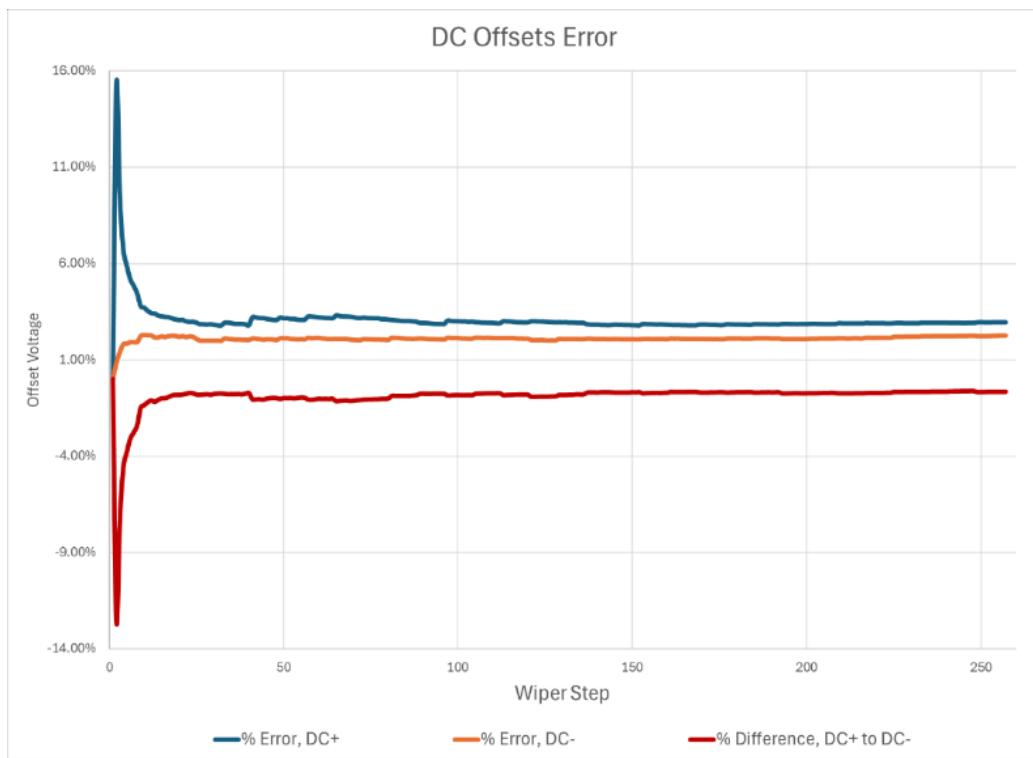


Figure 76: DC Offset Voltage Error with Respect to Ideal, and Each Other

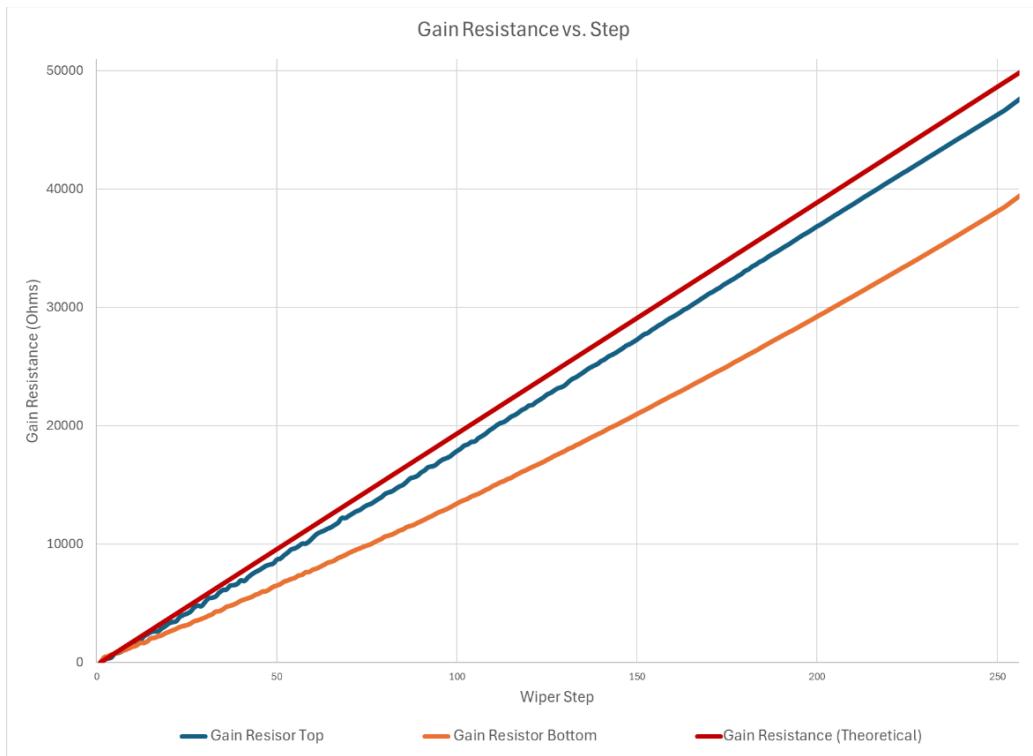


Figure 77: Gain Resistance Measurements Versus Wiper Step Location

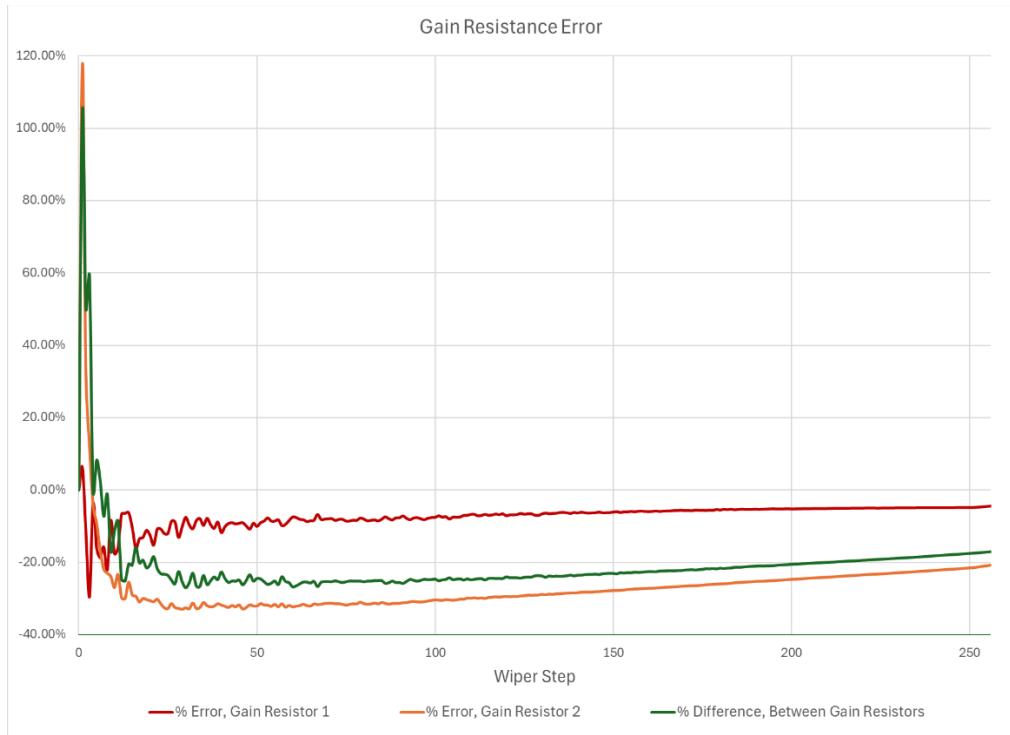


Figure 78: Software-Defined Instrumentation Amplifier Gain Resistance Error

With the calibration of the software-defined instrumentation amplifier portion of the signal conditioner subsystem complete, a preliminary test of the calibration software component was performed. Following from the engineering requirements of the signal conditioner subsystem:

- Interprets the analog voltages produced by the sensors as digital data
- Converts the digitized analog voltage readings to relevant measurement units for pH, turbidity, and temperature
- Formats the data to be sent to the central gateway module

Accurate and reliable operation of the software-defined instrumentation amplifier, which pre-processes incoming analog sensor voltages to be compatible with the ADC and provide increased resolution in the measurement region of interest, are necessary for full operation of this subsystem.

Table 24 and Table 25 below show the performance of the software-defined instrumentation amplifier in meeting desired analog characteristics for both gain and input DC offset. The values selected by the firmware using the calibration data were then compared against the resulting values if an idealized potentiometer was considered in the code to compute both the gain and the DC offsets. For example, if a gain of 3 is requested by the user or within an autonomous call to the function in the code, the firmware will select the combination of gain resistors on both the top and bottom wipers that are as close to $10k\Omega$ as possible ($Gain = \frac{2 \times 10k\Omega}{R_G} + 1 = 3 \rightarrow R_G = 10k\Omega$). If this calibration data was not used, the firmware would instead need to select the wiper positions assuming an ideally linear potentiometer with end-to-end resistance of $50k\Omega$. For a gain of 3, the selected wiper position would be calculated as: $(1 - \frac{10k\Omega}{50k\Omega}) * 256 = 205$, producing a gain resistance of $9.8132k\Omega$ given the measured resistance of the potentiometer at that location. This deviation carries through to a resulting gain of 3.03, introducing non-negligible error into the system. The resulting performance difference in the amplifier output voltage and computed input voltage by the firmware for both the calibration-controlled system and the non-calibrated system are shown in Table 26 and Table 27 below.

As seen in the tables below, the integration of calibration data into the software-defined instrumentation amplifier greatly improves system performance, driving gain to 0.26% in the worst case compared to 11.63% in the worst case for the uncalibrated system, and ultimately resulting in significant reductions in error between the calculated input voltage of the system and the actual input voltage, demonstrating the software component of the design's ability to help the system meet its functional requirements.

Table 24: Gain Selection of the Software-Defined Instrumentation Amplifier

Input Voltage	Ideal Gain	Actual Gain, Assuming Ideal Pots	% Error	Software-Selected Gain	% Error
0.10727	12	10.64	11.33%	12.030886	0.26%
0.40825	4	3.95	1.25%	4.000105	0.00%
0.7996	2	2.01	0.50%	2.000037	0.00%
1.20286	3.5	3.469	0.89%	3.49919	0.02%
1.5077	3	3.0022	0.07%	2.99962	0.01%
1.8005	5	4.936	1.28%	4.998462	0.03%
2.00948	2.5	2.513	0.52%	2.500217	0.01%

Table 25: DC Offset Selection of the Software-Defined Instrumentation Amplifier

Input Voltage	Ideal DC Offset	Actual DC Input Offset, Assuming Ideal Pots	% Error	Software-Selected DC Offset	% Error
0.10727	0	0.00002512	N/A	0.002512	N/A
0.40825	0	0.00002512	N/A	0.002512	N/A
0.7996	0	0.00002512	N/A	0.002512	N/A
1.20286	-1	-0.977	2.30%	-0.994864	0.51%
1.5077	-1	-0.977	2.30%	-0.994864	0.51%
1.8005	-1.5	-1.534	2.27%	-1.492605	0.49%
2.00948	-1.5	-1.534	2.27%	-1.492605	0.49%

Table 26: Output Voltage of the Software-Defined Instrumentation Amplifier

Input Voltage	Theoretical ADC Voltage	Measured ADC Voltage	% Error, w.r.t. Theoretical	ADC Voltage, Assuming Ideal Pot	% Error, w.r.t. Theoretical	% Difference, Measured vs. Idealized
0.10727	1.2	1.425812	18.82%	1.141620077	4.86%	19.93%
0.40825	1.6	1.703875	6.49%	1.612686724	0.79%	5.35%
0.7996	1.6	1.602	0.13%	1.607246491	0.45%	0.33%
1.20286	0.7	0.7775	11.07%	0.78350834	11.93%	0.77%
1.5077	1.5	1.598125	6.54%	1.59326754	6.22%	0.30%
1.8005	1.5	1.639438	9.30%	1.315444	12.30%	19.76%
2.00948	1.25	1.321938	5.76%	1.19488124	4.41%	9.61%

Table 27: Input Voltage Calculation for the Software-Defined Instrumentation Amplifier

Input Voltage	Actual ADC Voltage	Calculated Input Voltage, Assuming Ideal Pots	% Error, w.r.t. Actual	Software-Estimated Actual Input Voltage	% Error, w.r.t. Actual
0.10727	1.425812	0.133979767	24.90%	0.11463	6.86%
0.40825	1.703875	0.431335639	5.65%	0.419321	2.71%
0.7996	1.602	0.796989805	0.33%	0.790224	1.17%
1.20286	0.7775	1.201127991	0.14%	1.212298	0.78%
1.5077	1.598125	1.509317967	0.11%	1.522139	0.96%
1.8005	1.639438	1.866138979	3.65%	1.817292	0.93%
2.00948	1.321938	2.060039793	2.52%	2.014735	0.26%

3.14. Complete Operational Test of the Signal Conditioner Subsystem

With the calibration data obtained for the amplifier and the signal acquisition process validated, the final testing step was using the amplifier to convert incoming analog sensor voltages to meaningful measurements for water quality management. Shown in the figures below are plots corresponding to the input voltage versus environmental measurement value for the pH sensor, RTD (temperature) sensor, and turbidity sensor, respectively. During testing of the RTD, a consistent +3mV offset was observed on the input voltage compared to the expected voltage measurement. To account for this, a -3mV offset was applied to the RTD input; since 3mV is too fine of an offset to apply using the software-defined instrumentation amplifier, this offset was applied in software. This change is reflected in Figure 80 below, showing the raw temperature measurement in blue, the actual environmental temperature in orange, and the calculated temperature in green after the -3mV input offset was applied. As shown in the figure, applying this offset produced a temperature response much more consistent with the ideal response.

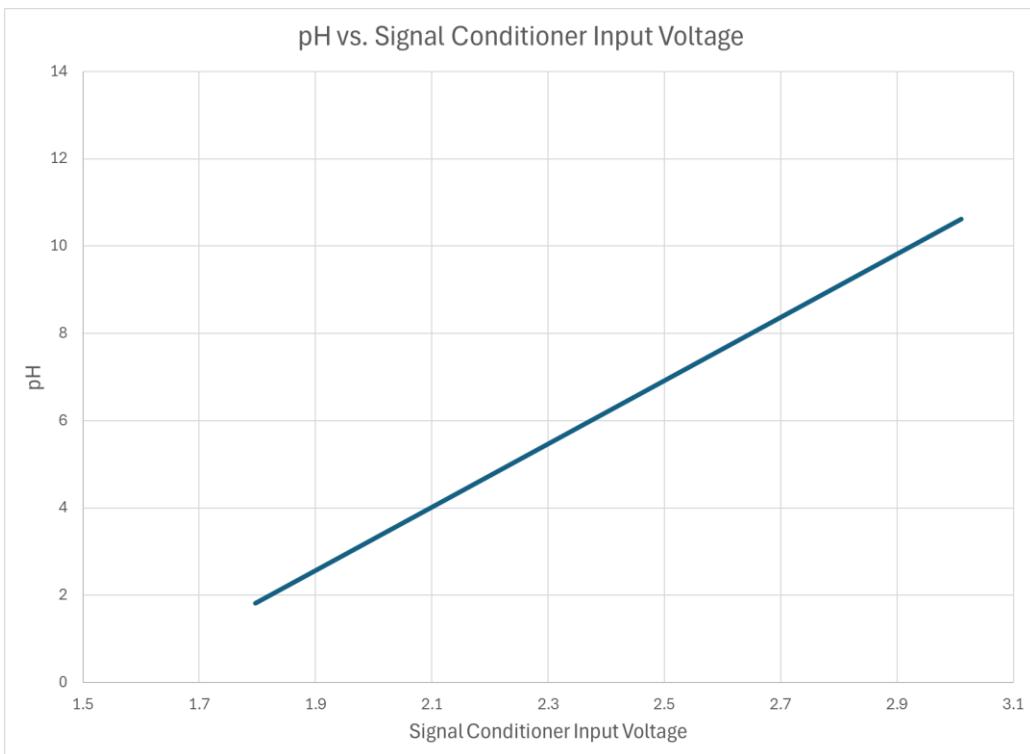


Figure 79: pH Sensor Voltage as measured by the Signal Conditioner versus Actual pH.

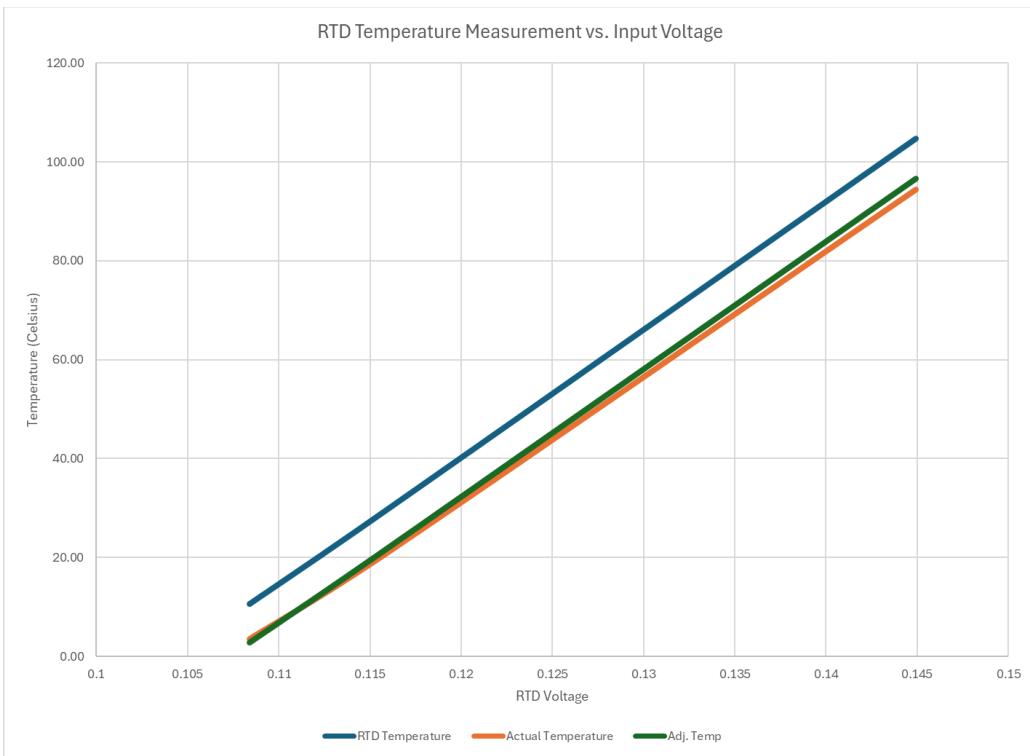


Figure 80: RTD Temperature vs. Input Voltage

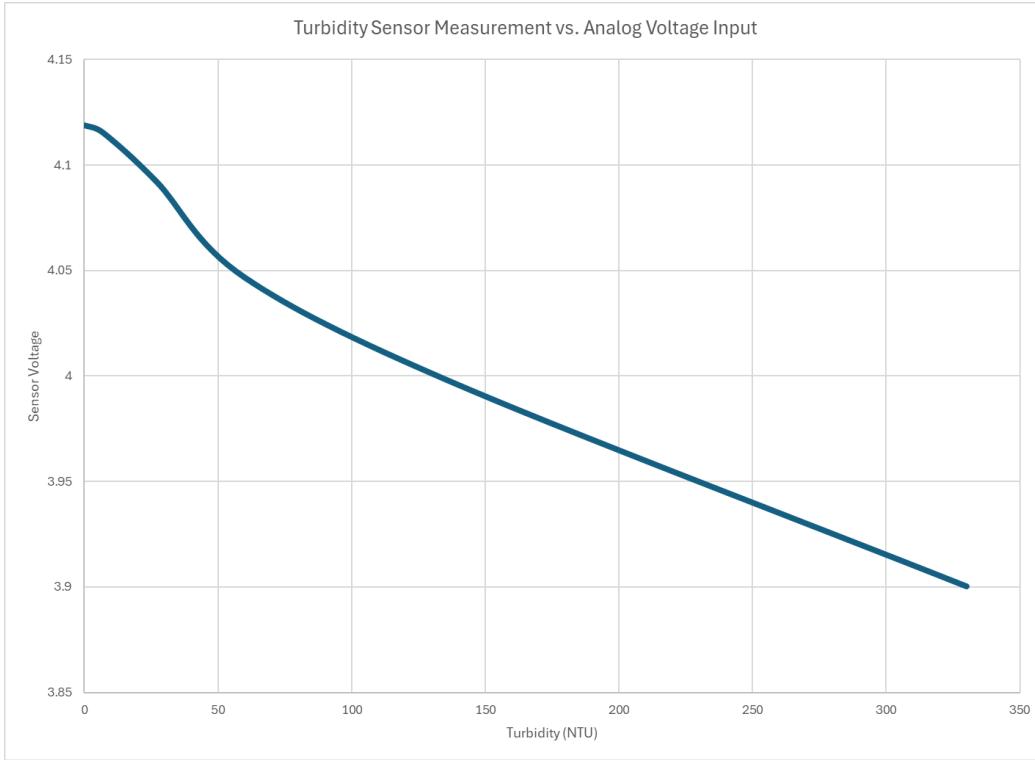


Figure 81: Turbidity Sensor Measurement vs. Input Voltage

Using these characteristic plots, the sensor voltage-measurement relationships could be implemented within the sensor node firmware. Using these plots, further testing was performed with the sensors, as detailed in Test #010 in Appendix D.

The pH sensor saw all tested values, measured using the sensor in buffer solutions with known pH, fall within the ± 0.3 requirement as outlined in the engineering requirements. Since temperature measurement requirements are not directly specified under the MCL or other similar regulations [31], a $\pm 0.35^\circ\text{C}$ tolerance band was adopted for this test, consistent with the maximum rated tolerance band of the Class A PT100 RTD used in our system [52]. Similarly, the estimated temperature reported by the signal conditioner subsystem was within this tolerance band for all temperature measurements tested, as checked against the reading produced by a meat thermometer. The turbidity sensor, however, failed to perform within our engineering requirements for measurement precision. As outlined in section 2.4.12, the turbidity sensor fails to produce a stable and measurable difference in its output voltage across the tested 0.1NTU through 1.0NTU range. Future work would need to be done with this project to stabilize the readings of our

chosen turbidity sensor or identify an alternative solution within the project's budget and other constraints.

Overall, the signal conditioner itself performs as designed, and can produce accurate datapoints that can be reported back to our central gateway. Apart from the turbidity sensor, the signal conditioner produced sensor telemetry within 2% of actual for all tested real-world values, demonstrating its performance at the core of the LoRa Water Quality Management System.

3.15. RP2350 Power Management Test

One of the key features of the sensor node firmware is the sleep mode, which disables the 5V power rail and forces all other components of the design into their respective sleep or power-saving modes. The largest source of these power savings, however, comes from the onboard power management capabilities in the RP2350, controlled through hardware-level register writes in the firmware. To validate this key functionality, a test of the power management setup and features was performed at the component level to provide a foundation for the completed firmware for the sensor node.

This test was performed using the code contained in `rp2350_powman.c` under the `tests/` directory in the GitHub repository linked in Appendix C. This code invokes the hardware-level RP2350 C/C++ SDK directly however, in the final firmware implementation, these invocations, setup data, and hardware-level checks were implemented within the HAL and ultimately utilized within a larger-scoped power state management library file built specifically for the sensor node.

By running the code on a Raspberry Pi Pico 2, the power management capabilities of the RP2350 Microcontroller could be evaluated. This test was not an accurate observation of the IC's power consumption directly, as this development board integrates other components such as a buck regulator, increasing overall current consumption of the module, as the RP2350 IC was not isolated. Since an Ammeter was not directly available, a 1Ω shunt resistor was placed at the power input to the Raspberry Pi Pico 2, and the current draw of the device was measured through the voltage across the resistor using a Keithley 2000 DMM. Though the exact datasheet-listed current consumption was not observed, a

significant reduction in the power consumption of the microcontroller was observed, with a 29.2 times reduction in the current draw of the device being observed. This demonstrated the value of using the power management tools within the RP2350's hardware for the final node firmware implementation.

3.16. Full Node Firmware Test

With component-level and subsystem-level testing of the sensor node complete, the final test performed on this element of the entire integrated system was a complete functional test of the sensor node's firmware, shown in Test #11 in Appendix D. During this test, the progression of the sensor node through the finite state machine stages shown in Figure 21. In each state, the current consumption of the node was tracked to estimate the battery life of the node in its final application.

As shown in Test #11 in Appendix D, the sensor node's firmware successfully entered each state without error. The reset, sample, transmit, and dormant states were all entered without error, and the sensor node successfully transmitted the full telemetry data to the gateway, where it was logged accordingly. A sample of the received packets from the sensor node as seen on the gateway are shown in Figure 82 below.

```
[PACKET TYPE] [TIME, NODE ID, TURBIDITY, TEMPERATURE, PH]
[PAYLOAD] ['2025-12-03T16:08:52.716918', 1, 0.0, 21.916546, 7.983061]
[PAYLOAD] ['2025-12-03T16:22:25.811927', 1, 0.0, 21.988014, 7.980621]
[PAYLOAD] ['2025-12-03T16:35:59.030313', 1, 0.0, 21.988014, 7.967029]
[PAYLOAD] ['2025-12-03T16:49:32.001948', 1, 0.0, 21.988014, 7.951694]
[PAYLOAD] ['2025-12-03T17:03:05.185994', 1, 0.0, 22.059483, 7.959711]
[PAYLOAD] ['2025-12-03T17:16:38.327649', 1, 0.0, 22.059483, 7.937754]
[PAYLOAD] ['2025-12-03T17:30:11.783605', 1, 0.0, 21.916546, 7.944027]
```

Figure 82: Gateway Packet Data Console Log

The full integration test of the firmware however, was unsuccessful in its power management capabilities. As established in Section 2.4.24, the theoretical input current to the node in its active and dormant modes are $152.7mA$ and $228.4\mu A$, respectively. During the active mode, a maximum input current into the sensor node of $135mA$ was observed, well within the estimated current threshold for the node. However, in the dormant mode, input current into the node was $96mA$, over 400 times the theoretical value. After this

observation, some testing was performed to identify the root cause of the failure, including testing and analysis of the LSF0102 Logic Level Translator IC in the dormant mode, however, due to time constraints the source of this issue was never identified.

3.17. Neural Net Prediction Test for Water Quality Metrics

The data was partitioned into training data, validation data, and testing data, with 10% of the data used for testing. The test had 18 batches each with 93 time steps at 30-min intervals. After each time step, the model was used to predict the following time step. Figure 83 shows the RMSE of the model improving as more time steps are fed into the model, as expected. Even though the model initially had a high RMSE due to starting without long-term memory, the results of the test showed a mean RMSE of 0.0291 for the 18 test batches, Figure 84.

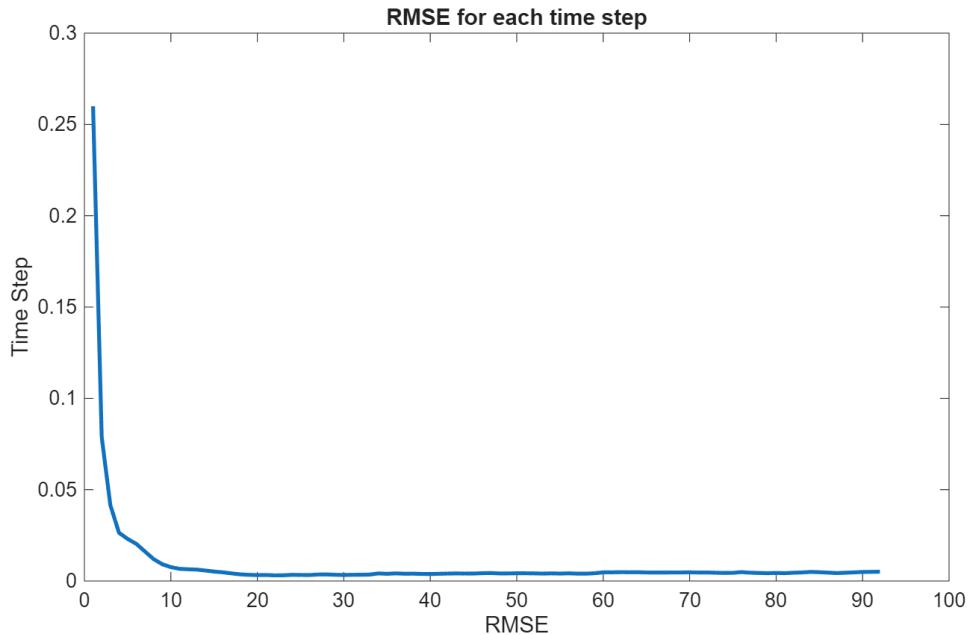


Figure 83: RMSE for Each Time Step, with Open-Loop Forecasting

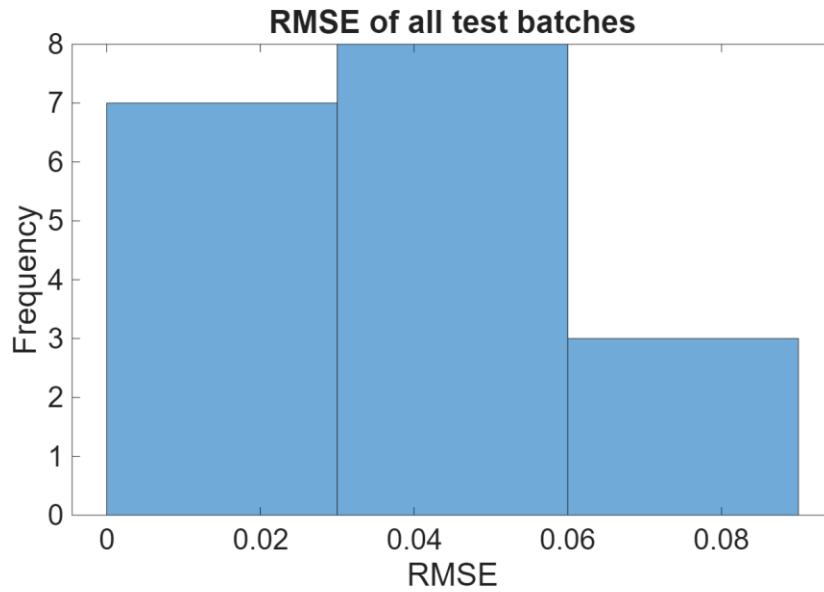


Figure 84: RMSE for the 18 Test Batches, with Normalized Data.

Additionally, an open loop forecasting test was performed. In this test, the model predicted a future time step using only input data. The model began making predictions after each new time step following the first 50th time steps of a test batch. A plot with the test data and the predicted data is shown in Figure 85. For this test, the RMSE was 0.0102 for temperature, 0.0082 for pH, and 0.0058 for turbidity.

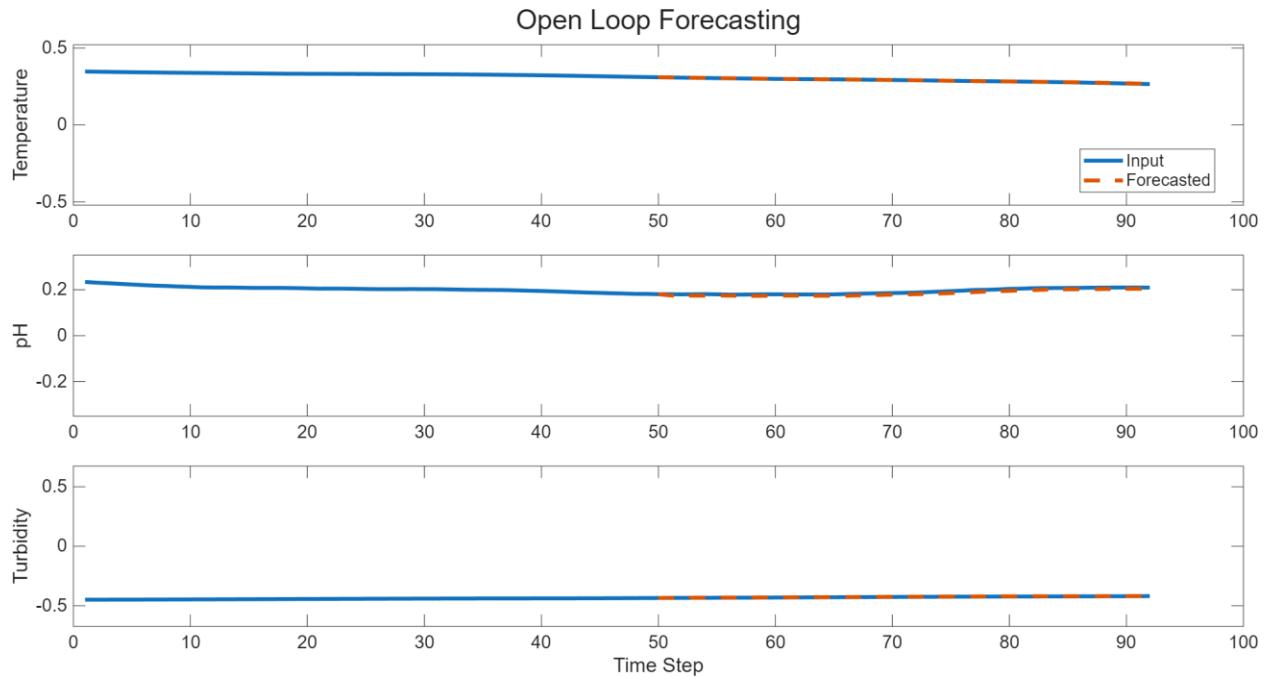


Figure 85: Plot of open-Loop forecast for a single batch with normalized data

The second test performed was a closed loop forecast. On this test, the last 20 timesteps of a batch were predicted using only the first 73 timesteps. The forecast of the neural network was made using both input and prediction data. The results showed that as the number of future timesteps increases the forecast accuracy of the forecast decreased, Figure 86. The 20-timestep-ahead forecast error for this batch was $+0.286^\circ$, 0.033 pH, and 0.317 NTU.

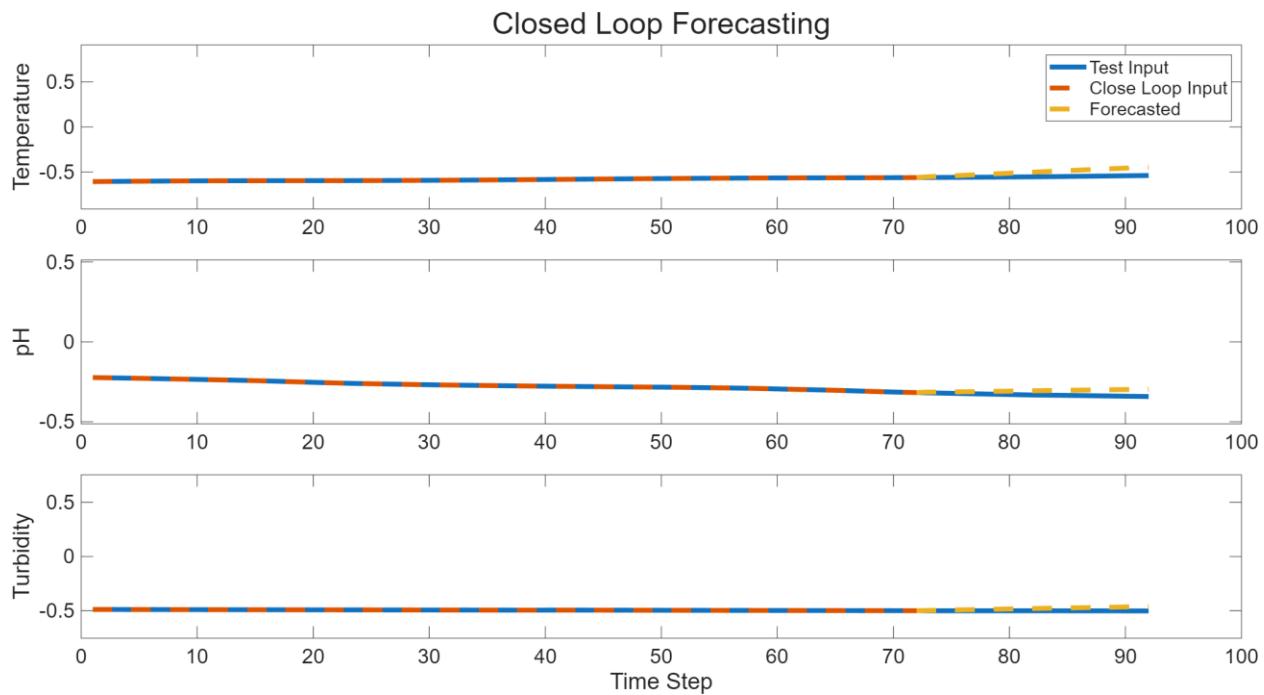


Figure 86. Plot of closed loop forecast for a single batch with normalized data

3.18. Gateway Dashboard Operation

During Test #13, the gateway Raspberry Pi ran the listener script to receive packets from the Gateway Receiver board and log pH, turbidity, temperature, timestamp, and node ID values to a CSV file. The dashboard script then read this CSV file, used the node ID to route each record to the correct node row, and displayed the corresponding sensor values and timestamp on the GUI. For all test cases, pH, turbidity, temperature, and timestamp data were displayed correctly on the dashboard, confirming proper integration between the listener and dashboard scripts. The subsystem test passed.

3.19. Gateway Alert Generation Test

The purpose of this test (Test #21) was to verify that the dashboard correctly detects alert conditions and generates both on screen and email notifications. The CSV log file that the dashboard reads was manually edited to inject specific test cases. For each case, a new row was added so that one parameter was forced out of its configured threshold or the timestamp was set to more than 15 minutes.

Four alert scenarios were evaluated: pH out of range, turbidity out of range, temperature out of range, and a node timeout due to an old timestamp. In every case, the status banner changed from green to red and displayed a message describing the active alert. The resulting dashboard states for pH, turbidity, temperature, and node timeout alerts are shown in Figure 87 below.



Figure 87: Dashboard Alert Displays

At the same time, the dashboard script generated an email for each alert condition using the configured Gmail account. Each email contained a subject line and message body that identified the affected node and the specific parameter or timeout condition that triggered the alert. The corresponding alert emails for the four test cases are shown in Figure 88 below. In all scenarios, the dashboard and email behavior matched the expected results, so the alert generation test passed.

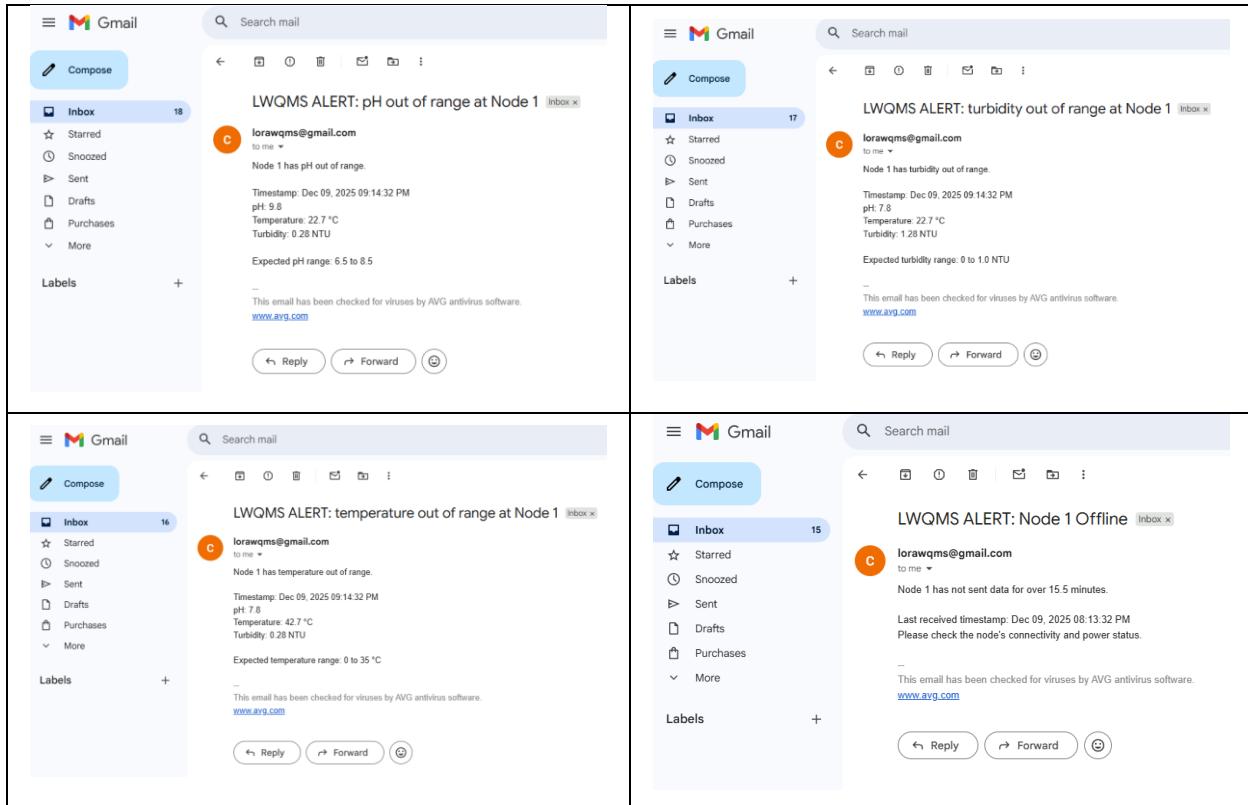


Figure 88: Water Quality Email Alerts

3.20. Full System Integration Test

The final test performed to validate the functionality of the LoRa Water Quality Management System was a complete integration test of the system. In this test, the sensor node was deployed fully autonomously, relying on its own on-board firmware to control and manage data acquisition, wireless communications, power management, and reporting intervals. This test used the final firmware version with high-level driver file `main.c` in the GitHub repository in the Appendix. Similarly, the gateway was also deployed fully autonomously, with the wireless interface communicating directly to the Raspberry Pi 4, and values were automatically updated on the dashboard as new data packets were

received. As the neural network had not been fully tested or integrated with the rest of the system, its operation was omitted from this test.

The sensor node and gateway as deployed for this test are shown in Figure 89 and Figure 90 below, respectively. As seen in these images of the test setup, the water sample used was taken from tap water at the University and measured in cups. A full-scale integration test of the system would require testing in real environmental conditions, but time and availability constraints meant that testing needed to be performed in this manner.

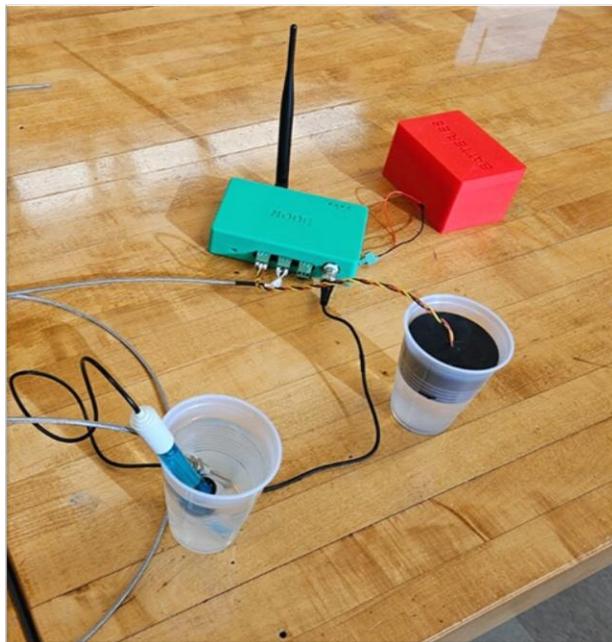


Figure 89: Full System Integration Test Setup for Sensor Node



Figure 90: Full System Integration Test Setup for Gateway

Over the course of the five-hour period, water quality data was transferred from the sensor node to the gateway approximately every 15 minutes. Table 28 below shows the complete water quality data transfer from the node to the gateway over the full test time. This data was then plotted to better visualize fluctuations in the data or diagnose potential water quality data corruption, as shown in Figure 91 below. As seen in both the table and figure, all water quality data stayed within the acceptable range according to our adopted standards for the full duration of the test, consistent with expectations.

Table 28: Water Quality Telemetry from Full-System Integration Test

Time	Sensor Node ID	Turbidity (NTU)	Temperature (C)	pH
2025-12-03T13:12:42.562117	1	0.026688108	22.631227	8.125954
2025-12-03T13:26:16.704028	1	0.043388661	22.631227	8.127348
2025-12-03T13:39:49.616388	1	0.044995797	22.559761	8.117241
2025-12-03T13:53:22.316404	1	0.040772587	22.559761	8.108876
2025-12-03T14:06:54.860937	1	0.04202216	22.488293	8.102951
2025-12-03T14:20:27.530145	1	0.016331479	22.345356	8.097724
2025-12-03T14:34:00.239210	1	0.022817315	22.202419	8.09389
2025-12-03T14:47:33.072449	1	0.018963728	21.916546	7.868399
2025-12-03T15:01:06.140772	1	0.007568628	21.988014	8.056947
2025-12-03T15:01:07.320772	2	0.044477013	22.758014	7.866947
2025-12-03T15:14:39.292568	1	0.018666621	21.988014	8.062523
2025-12-03T15:28:12.521369	1	0.030003862	21.916546	8.034293
2025-12-03T15:41:46.006811	1	0.044904909	21.773609	8.003624
2025-12-03T15:55:19.521855	1	0.037500128	21.845078	7.986895
2025-12-03T16:08:52.716918	1	0.017172061	21.916546	7.983061
2025-12-03T16:22:25.811927	1	0.005171187	21.988014	7.980621
2025-12-03T16:35:59.030313	1	0.021938985	21.988014	7.967029
2025-12-03T16:49:32.001948	1	0.017016039	21.988014	7.951694
2025-12-03T17:03:05.185994	1	0.027982176	22.059483	7.959711
2025-12-03T17:16:38.327649	1	0.005310755	22.059483	7.937754
2025-12-03T17:30:11.783605	1	0.046132796	21.916546	7.944027

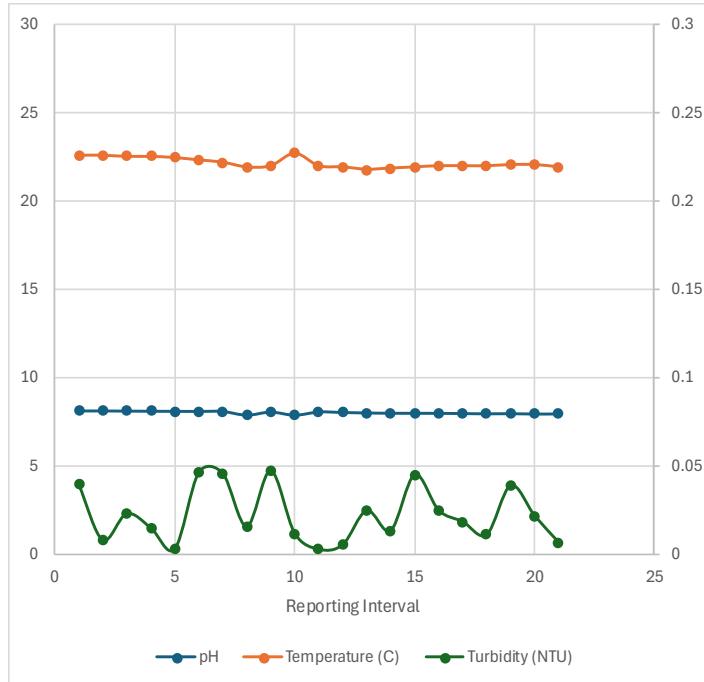


Figure 91: Full System Integration Test Water Turbidity, pH, and Temperature

During the early stages of this test, it was observed that the sleep time interval was highly inaccurate, with observed sleep times exceeding those programmed directly in the firmware by 33%. Though inaccurate, this deviation was consistently observed to be 133%

of the code-specified value. Upon further investigation of this issue, this observation is likely the result of the on-chip 32kHz low-power oscillator (LPOSC) which is used to provide timing for the microcontroller when the device is in a low-power state and the calibrated, external 12MHz crystal used for synchronization of the logic core is not in use. The RP2350's LPOSC has a nominal frequency of 32.768kHz, however it can vary from 26.2144kHz through 39.3216kHz [40]. As this oscillator is being used to provide timing-sensitive sleep intervals, this wide tolerance band of the oscillator resulted in the large deviation seen in the dormant state. To improve long-term manufacturability of the device without needing to calibrate this aspect per-board, an external calibrated LPOSC or real-time clock (RTC) would need to be integrated with the microcontroller layout. For this reason, a sleep interval of 11 minutes was specified in the final code, giving an actual sleep interval of 14.66 minutes.

4. Project Management

4.1. Project Work Breakdown Structure (WBS)

Figure 84, shown on the following page, presents the work breakdown structure (WBS) for the development of the LoRa Water Quality Management System. Based on the scope of the project, five main tasks were outlined: research, design, fabrication, testing & validation, and documentation. Based on the prior skills, experience, and areas of interest of each of the team members, the tasks and subtasks were then assigned to distribute the overall workload as fairly as possible. To make the scope of the overall project more feasible, PLC integration was removed from the WBS as technicians viewed this feature less desirable than other portions of the project.

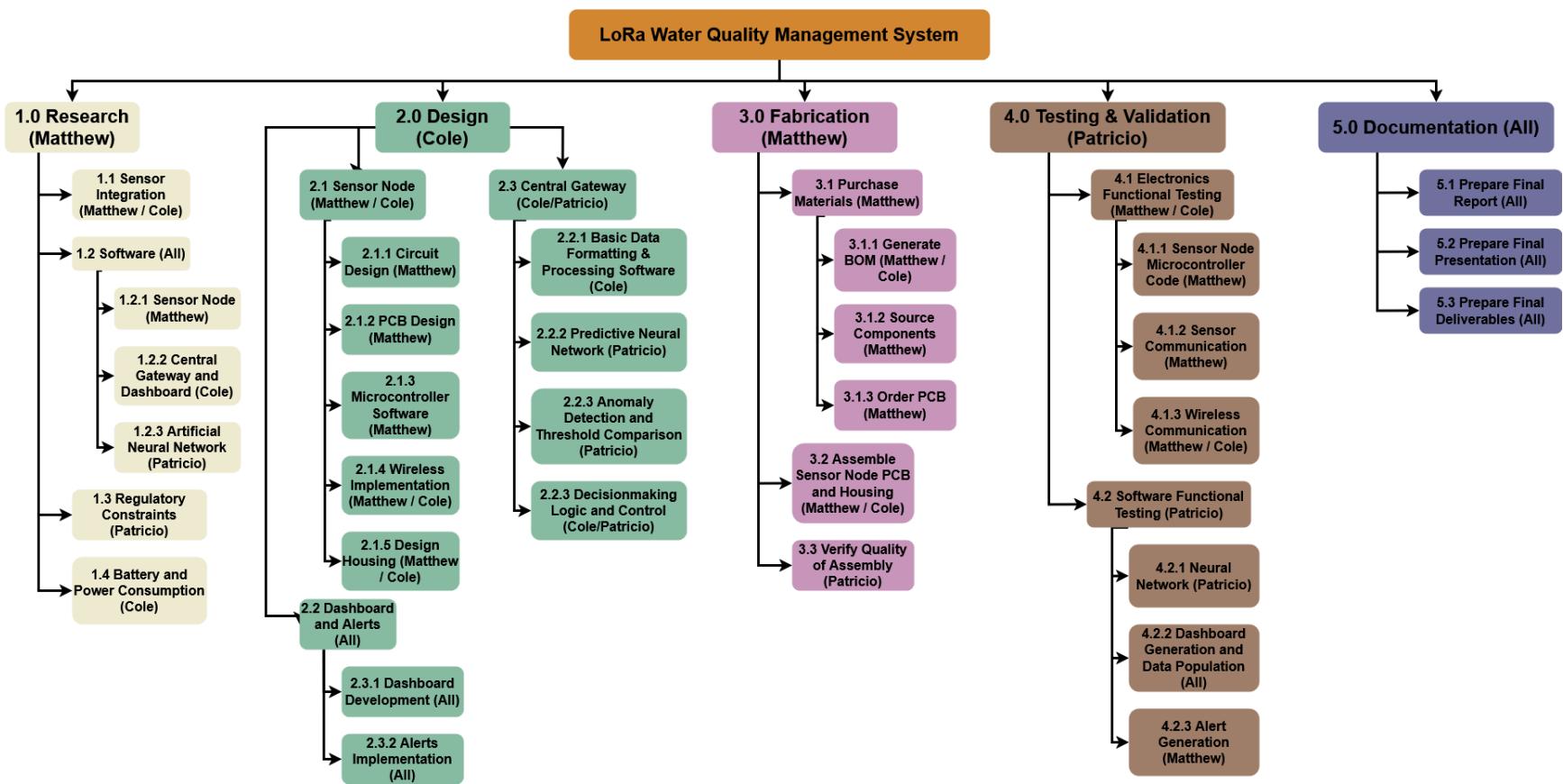


Figure 92: Work Breakdown Structure

4.2. Project Schedule (Gantt Chart)

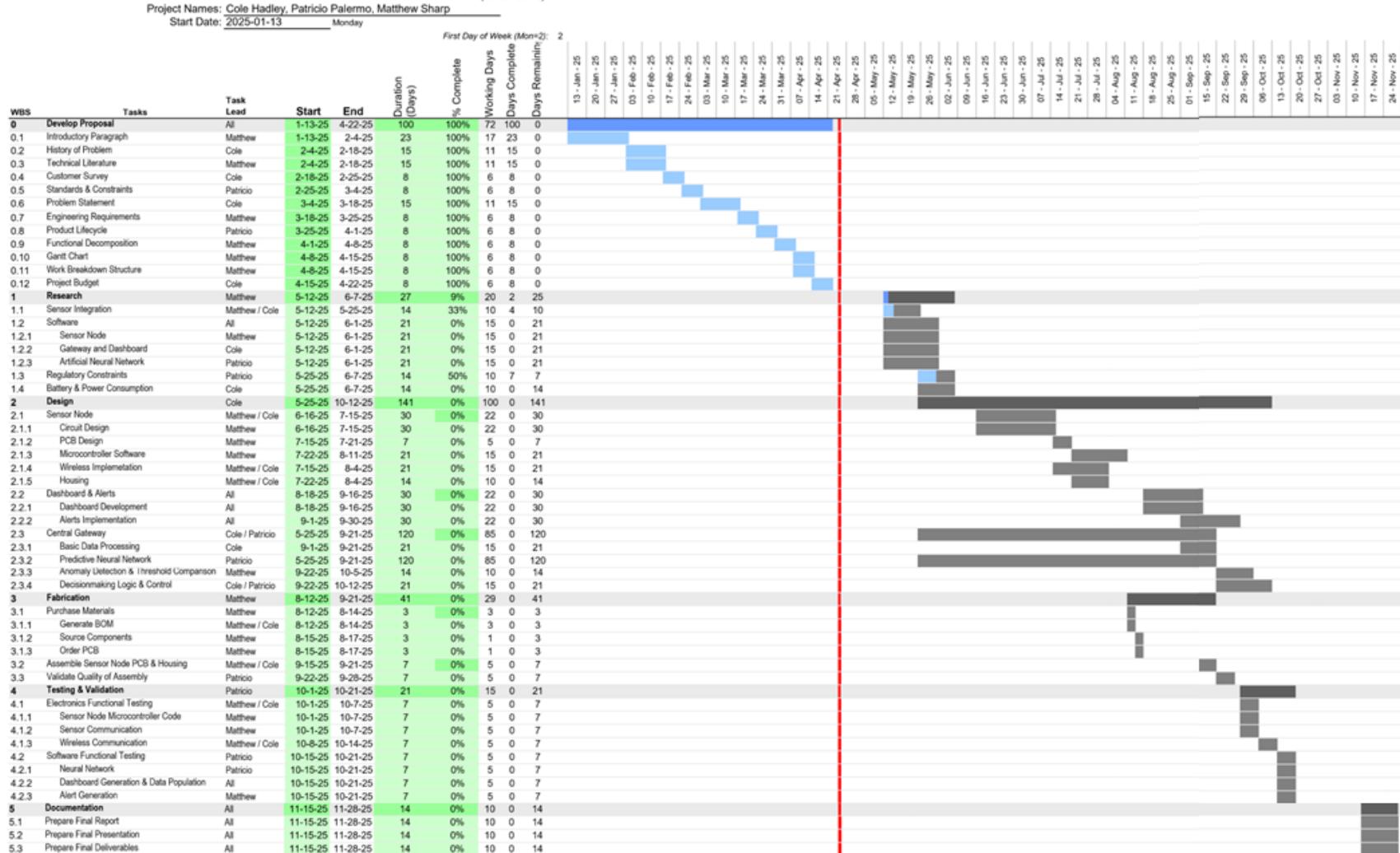
Table 29 presents the project schedule used to complete development of the LoRa Water Quality Management System. The schedule includes the work completed during the proposal semester as well as the subsequent research, design, fabrication, and testing phases. Work continued through the summer and Fall 2025 semester, with milestones adjusted as needed while still meeting the deadline for the Capstone Design Conference. Overall, the Gantt chart provided a roadmap for organizing tasks, tracking progress, and ensuring the project was completed in time for final demonstration and documentation.

Table 29: Gantt Chart

LoRa Water Quality Management System

The Pennsylvania State University

Today's Date: 2025-04-25 Friday
(vertical red line)



4.3. Project Budget

The cost of labor was determined by using the Gantt chart and assuming two hours of labor per allotted workweek of each task, based on the credit hours of the class. The sections of the work breakdown structure were assigned to the labor hours as follows: The proposal development and documentation sections were included as consulting hours; the fabrication section was included as assembly hours; the design and research sections were included as design hours; the testing and validation section were evenly split unto testing and validation hours; finally, for the coding hours, each individual task that required coding, such as the predictive neural network task, was included. The estimated unit cost for labor was obtained from the website Indeed and used to calculate the total labor cost [66]. The total labor cost is \$31,365.22, shown in Table 30.

Table 30: Engineering Labor Cost of Project.

Item	Unit	Unit Cost	# Units	Cost
Design	hr	49.00	235	\$ 13,818.00
Coding	hr	52.96	360	\$ 6,143.36
Assembly	hr	20.06	17	\$ 409.22
Testing	hr	25.54	15	\$ 459.72
Validation	hr	25.54	15	\$ 459.72
Consulting	hr	41.98	200	\$ 10,075.20
Labor Subtotal				\$ 31,365.22

The parts section was initially obtained by analyzing the level-2 decomposition and determining which parts will be needed to implement each function and sub-processes and is now updated to reflect the cost of the parts actually ordered. The cost for all the parts is \$242.51, Table 31.

Table 31: Parts Cost of Project.

Assembly	Item	Qty	Unit Cost	Subtotal
Sensor node	DFRobot Turbidity Sensor	1	\$9.90	\$9.90
	DFRobot pH Sensor	1	\$29.90	\$29.90
	PT100 RTD	1	\$14.00	\$14.00
	PCB components	1	\$47.73	\$47.74
	PCB	1	\$2.00	\$2.00
	D-cell battery holder (4.5 V)	1	\$4.50	\$4.50
	D-cell batteries (4-pack)	1	\$7.00	\$7.00
	Sensor node enclosure	1	\$5.20	\$5.20
	915 MHz antenna	1	\$4.50	\$4.50
			Sensor node total	\$121.63
Gateway	Raspberry Pi 4 (8 GB)	1	\$75.00	\$75.00
	Raspberry Pi Pico 2	1	\$5.00	\$5.00
	915 MHz antenna	1	\$4.50	\$4.50
	PCB components	1	\$10.88	\$10.88
	PCB	1	\$2.00	\$2.00
	Gateway enclosure	1	\$3.50	\$3.50
	Misc cables	1	\$20.00	\$20.00
			Gateway total	\$120.88
			Total parts cost	\$242.51

In total, the project cost is \$31,607.73, shown in Table 32. Most of the cost of the project is due to labor costs, in particular, design and coding, with parts cost being minimal.

Table 32: Total Cost of Project

Categories	Cost
Labor Subtotal	\$ 31,365.22
Parts Subtotal	\$ 242.51
Grand Total	\$ 31,607.73

5. Summary and Conclusion

As outlined in this proposal, the LoRa Water Quality Management system presents an innovative solution to water utility companies for the monitoring and management of drinking water quality. Unlike traditional water quality monitoring systems, which often require labor-intensive manual water testing or sensors that must be hard-wired into legacy SCADA-based systems, the LoRa Water Quality Management System makes use of power-efficient LoRa wireless technology, whose name is derived from “Long Range,” to transmit real-time water quality data back to a central gateway without the need for cellular service, a WiFi connection, or a hard-wired data link. The data transmitted from a network of LoRa-based sensor nodes is processed at a central gateway, which aggregates and summarizes system-wide water quality data in a visual dashboard, alerts technicians and engineers when measured data exceeds legal or utility-defined thresholds, and utilizes AI-driven analytics to predict future system water quality. Implementation of the LoRa Water Quality Management System enables technicians and engineers to monitor water quality system-wide and take corrective action before issues arise.

The prototype implementation discussed in this report demonstrates a strong foundation for a future full-scale implementation of the LoRa Water Quality Management System. The final design facilitates a complete water quality data flow path; water quality metrics are collected at the source through sensors which represent the data as analog voltage, where the incoming signals are processed through an analog frontend and converted to meaningful telemetry data points through digital post-processing. The data is then transmitted wirelessly over LoRa to the central gateway module, where it is written out to a visual dashboard and presented to technicians and engineers through this display. The system then evaluates the incoming data against preset thresholds and generates mobile alerts via email. Finally, a prototype implementation for AI-driven analytics and predictions was demonstrated, allowing the system to predict its future water quality through a LSTM neural network.

The sensor node and gateway modules are implemented as complete, self-contained modules. The sensor node completely integrates all of its functions on a single PCB,

requiring no external hardware apart from the sensors and antenna to be fully functional, making it a strong prototype for a finished product within a completed system. Similarly, the gateway module is also fully integrated within a single housing, demonstrating its strengths as a self-contained module for a prototype of a finalized system.

5.1. After-Action Report

Overall, our team functioned fairly over the course of the proposal and development phases of the project. Since our project was conducive with a divide-and-conquer approach, each team member spearheaded specific portions of the project, reducing conflicts due to individual preferences for working style or disagreements over low-level implementation details. Since individual team members took responsibility for different phases of the project that could be implemented concurrently, this group work structure reduced idle time for all team members, allowing for a much more effective working style. Though each team member was responsible for a different aspect of the project, collaboration was still central to our team's operation, as feedback and support was offered between team members as different phases of the project were completed.

During the development of the proposal, our group was able to define the problem, customer requirements, and engineering requirements such that work could begin on the project as soon as the proposal was finished and approved. This understanding was developed through a comprehensive survey of technicians at three different water utility companies, providing a diverse range of perspectives to inform our final design. Since the design phase of the project could be started directly from our problem definition and understanding of the corresponding engineering metrics, our team was able to hit the majority of the customer needs and engineering requirements. The prototype design demonstrates an ability to track water quality data remotely and fully autonomously, and communicates over a wireless protocol that does not rely on cellular or Wi-Fi infrastructure. Turbidity, temperature, and pH data is tracked every 15 minutes for compliance with the Pennsylvania MCL, and presents the data using a visual dashboard which generates alerts on-screen and remotely via email. However, the system does not achieve the battery life requirement due to high current draw in its dormant state, as

outlined in Section 3.16, and the turbidity data resolution is limited by the performance of the budget turbidity sensor used for this project.

The design phase of this project was informed by numerous electrical engineering courses at Penn State Harrisburg. The hardware design of the sensor node and receiver PCBs was guided by EE 410 for the analog frontend design and power regulation, EE 317 for PCB layout, EE 434 for design of key impedance-controlled traces for the LoRa RF interface and USB 2.0 differential pair, and EE 311 for amplifier design and performance characteristics. The firmware on both boards was informed by CMPEH 417 and EE 461 for an understanding of radio operational characteristics. EE 456/556 was used extensively during the development of the LSTM neural network, and CMPSC 131 was used for software development in Python for the dashboard. In addition to prior coursework, the development of the hardware and software within the system was continuously being informed by component datasheets, manufacturer documentation, regulatory standards, and other literature. In addition to documentation, the project relied extensively on consultation with our technical advisor, Dr. Aldo Morales, as well as the Lab Coordinator, Terry Tirko.

This project offered an excellent opportunity for learning, as much of the project, though informed by prior experience and coursework, required a great deal of additional research and independent learning to implement. The neural network, hardware, firmware, and software design all required additional research and learning beyond what was already known to execute. At the team-level, much learning about the nature of water quality monitoring, water utility treatment systems, and regulatory requirements surrounding this topic needed to be performed. Additionally, cross-functional collaboration was employed throughout the project, requiring team members to learn more about parts of the project that were not their immediately assigned responsibility.

A full after-action matrix for this project is attached in Appendix E of this report.

5.2. Future Work

From a system-level perspective, some significant deficiencies of the current prototype's implementation is the incorporation of only a single sensor node. Future implementations of the project would need to utilize a more robust networking protocol to facilitate the transfer of data from multiple sensor nodes reliably to a single gateway module. One potential avenue to implement this would be through watching for channel activity at the specified LoRa center frequency through the SX1262 LoRa Modulator IC's integrated channel activity detection functionality [48]. This would allow multiple sensor nodes to communicate using LoRa with no changes to the hardware and minimal additions to the sensor node's firmware. Additionally, one of the main disadvantages of LoRa is that its range is severely limited in applications where line-of-sight is not present between the sender and receiver. For this reason, a mesh network would need to be implemented. As shown in Figure 93 below, using a mesh network would enable packet transmission between a sensor node and the gateway even when line-of-sight is not present, as packets could be forwarded along by other sensor nodes to the destination. The current system is implemented as a star topology, as shown in the figure below, as data transfer is only supported from the node to the gateway directly.

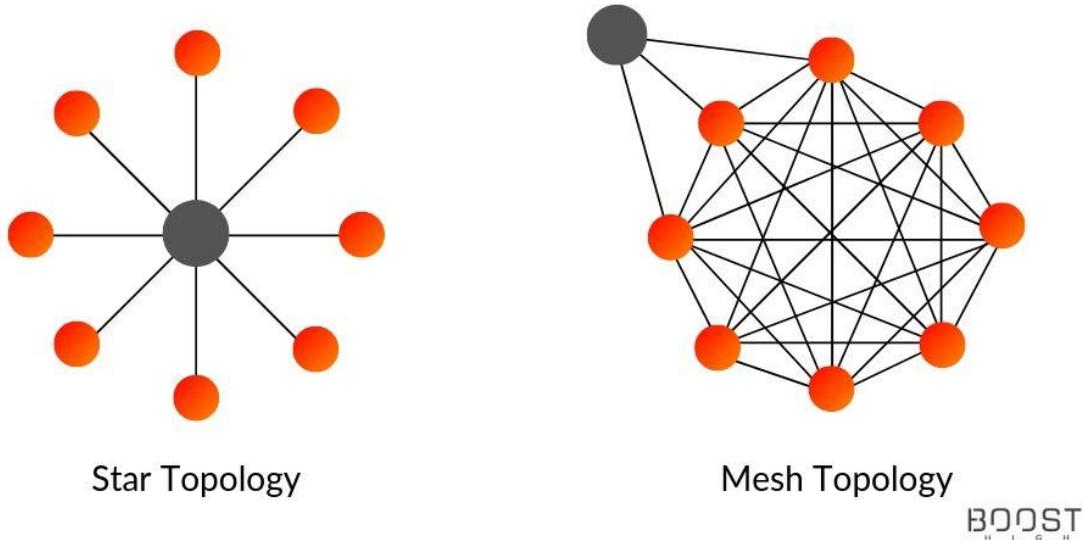


Figure 93: Star and Mesh Networking Topologies [67]

Another significant deficiency in the current wireless data transfer implementation is the limited data rate, with only a single 19-byte packet allowed per sensor node within a 20 second interval, and its lack of encryption. Encryption was not used in the current implementation due to the length of the initialization vector would leave only 3 bytes of data left in the packet with a standard 128-bit approach. Both limitations could be resolved by implementing a LoRa frequency hopping scheme, as under Title 47, Part 15, § 247 of the CFR, this is the only way to bypass this dwell time limitation. Implementing such a scheme would enable longer packets to be sent between devices and enable a fully encrypted peer-to-peer LoRa network.

From a hardware perspective, future designs should include a real-time clock IC interfaced with the RP2350 to reduce uncertainty in the hibernation state sleep time, and provide calibrated reporting intervals to ensure regulatory compliance. Additionally, the analog frontend of the design would benefit greatly from buffered digital potentiometer and sensor outputs, to prevent sensors from being loaded down and to prevent issues with the digital potentiometers creating voltage dividers with the amplifier resistors, as was initially observed during final testing of the design. These buffers could be implemented using a simple operational amplifier in a voltage follower configuration, requiring minimal additional board space and adding a small complexity and BOM cost to the design. Additionally, the pH sensor Pre-Amplifier could be removed from future versions of the design, as the chosen operational amplifier, the TLV9144, provides a extremely high input impedance and would support the negative output voltage of the pH sensor.

To improve the lifespan of the product and eliminate maintenance intervals of the system, the D-Cell battery input could be replaced with a solar-powered DC input. This would mitigate many of the concerns surrounding battery life of the design, however introduce new constraints such as the integration of a rechargeable battery for use overnight or on cloudy days. Since the design is configured to accommodate a DC input between 2.4 and 5.5V, making the sensor node solar powered would require minimal additional design.

From a sensors perspective, an improved version of the design would implement a custom IR-based turbidity sensor, or integrate a more professional-grade device, as the

inexpensive sensor designed for use in washing machines used in our design produced an extremely unstable and unreliable measurement. Integrating a custom or higher-end sensor would greatly improve the performance of the design.

Finally, additional neural network topologies could be investigated to identify potential opportunities for improved performance in the prediction of future water quality data. Since LSTM is one of several neural network designs targeted for time series predictions and analysis, evaluating the performance of these alternatives could reveal a superior architecture with better performance. Additionally, the neural network would need to be fully integrated within the dashboard software to completely incorporated this component of the system. Lastly, the number of parameters in the neural network could be optimized to balance power consumption and accuracy of forecast, depending on the requirements of the system.

References

- [1] A. N. Angelakis, N. Dercas and V. A. Tzanakakis, "Water Quality Focusing on the Hellenic World: From Ancient to Modern Times and the Future," *Water*, vol. 14, no. 12, p. 1887, June 2022.
- [2] Encyclopedia Britannica, "Antonie van Leeuwenhoek," [Online]. Available: <https://www.britannica.com/biography/Antonie-van-Leeuwenhoek>. [Accessed 17 February 2025].
- [3] J. Hassan, "The Victorian and Edwardian Periods," in *The History of Water in Modern England and Wales*, New York, New York, Manchester University Press, 1998, pp. 10-50.
- [4] J. L. Oblinger and J. A. Koburger, "Understanding and teaching the most probable number technique," *Journal of Milk and Food Technology*, vol. 38, no. 9, pp. 540-545, 1975.
- [5] B. Forster and C. A. Pinedo, "Bacteriological examination of waters: membrane filtration protocol," American Society for Microbiology, 2015.
- [6] Environmental Protection Agency, "History of the Clean Water Act," [Online]. Available: <https://www.epa.gov/laws-regulations/history-clean-water-act>. [Accessed 17 February 2025].
- [7] Environmental Protection Agency, "Overview of the Safe Drinking Water Act," [Online]. Available: <https://www.epa.gov/sdwa/overview-safe-drinking-water-act>. [Accessed 17 February 2025].
- [8] S. D. Richardson, "Water Analysis," *Analytical Chemistry*, vol. 71, no. 12, pp. 181-216, 1999.
- [9] J. Collings, "A history of SCADA for the Water Sector," 18 March 2022. [Online]. Available: <https://www.wwdmag.com/utility-management/article/10940688/a-history-of-scada-for-the-water-sector> . [Accessed 17 February 2025].

- [10] G. A. López-Ramírez and A. Aragón-Zavala, "Wireless Sensor Networks for Water Quality Monitoring: A Comprehensive Review," *IEEE Access*, vol. 11, pp. 95120-95142, 2023.
- [11] W. A. Jabbar, T. M. Tubg, M. F. I. Hamidun, A. H. C. Kamarudin, W. Wu, J. Sultan, A. A. Alsewari and M. A. H. Ali, "Development of LoRaWAN-based IoT system for water quality monitoring in rural areas," *Expert Systems with Applications*, vol. 242, p. 122862, 2024.
- [12] M. Miller et al., "IoT in Water Quality Monitoring-Are We Really Here?," *Sensors*, vol. 23, no. 2, p. 960, 2023.
- [13] Honeywell, *Merlin NB-IoT Clip-on LPWAN Communication Module Datasheet*, Honeywell, 2023.
- [14] R. S. Sinha, Y. Wei and S. -H. Hwang, "A survey on LPWA technology: Lora and Nb-IOT," *ICT Express*, vol. 3, no. 1, pp. 14-21, 2017.
- [15] S. Devalal and A. Karthikeyan, "LoRa Technology - An Overview," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2018.
- [16] P. K. Malik, P. Malik, G. R. Kumar, Sneha, R. Abraham and R. Singh, "Design and Implementation of a LoRa-Based Water Quality Monitoring System," in *3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, Ghaziabad, India, 2023.
- [17] In-Situ, *VuLink Data Logger and Telemetry Datasheet*, In-Situ, 2024.
- [18] In-Situ, *Aqua TROLL Multiparameter Sondes Datasheet*, In-Situ, 2024.
- [19] T. Murivhami, L. K. Tartibu and I. O. Olayode, "Applications of Artificial Neural Network in the Prediction of Water Quality Index," in *14th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, Cape Town, South Africa, 2023.

- [20] M. Hayat and H. Winkler, "Exploring the Basic Features and Challenges of Traditional Product Lifecycle Management Systems," in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Kuala Lumpur, Malaysia, 2022.
- [21] Sensorex, "TX3100 Intelligent pH/ORP Transmitter/Controller," [Online]. Available: <https://sensorex.com/product/new-tx3100-intelligent-ph-orp-transmitter-controller/>. [Accessed 17 February 2025].
- [22] MJK Automation, "CE TX3100 Technical Documentation," [Online]. Available: https://mjk.se/wp-content/uploads/2022/02/ce_tx3100.pdf. [Accessed 17 February 2025].
- [23] Phuong Hai Technical Service & Trading Co., "TX3100 pH/ORP Transmitter User Manual," [Online]. Available: <https://phuonghai.com/download/22635952.pdf>. [Accessed 17 February 2025].
- [24] 93rd Congress, "S.433 - Safe Drinking Water Act," 16 December 1974. [Online]. Available: <https://www.congress.gov/bill/93rd-congress/senate-bill/433/text>. [Accessed 2 December 2025].
- [25] Environmental Protection Agency, "Control of Lead and Copper," 7 June 1991. [Online]. Available: <https://www.ecfr.gov/current/title-40/chapter-I/subchapter-D/part-141/subpart-I>. [Accessed 2 December 2025].
- [26] National Science Foundation, "NSF/ANSI 61: Drinking Water System Components – Health Effects," 2016. [Online]. Available: https://d2evkimvhqav.cloudfront.net/documents/NSF-ANSI_61_watermarked.pdf?v=1594929800. [Accessed 2 December 2025].
- [27] Federal Communications Commission, "RF Safety FAQ," [Online]. Available: <https://www.fcc.gov/engineering-technology/electromagnetic-compatibility-division/radio-frequency-safety/faq/rf-safety>. [Accessed 2 December 2025].

- [28] Federal Communications Commission, "15.247 Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz.," 1 April 2020. [Online]. Available: <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15/subpart-C/subject-group-ECFR2f2e5828339709e/section-15.247>. [Accessed 2 December 2025].
- [29] Environmental Protection Agency, "Used Household Batteries," [Online]. Available: <https://www.epa.gov/recycle/used-household-batteries>. [Accessed 2 December 2025].
- [30] National Archives, "Code of Federal Regulations, Title 40, Part 141," 2020. [Online]. Available: <https://www.ecfr.gov/current/title-40/chapter-I/subchapter-D/part-141> . [Accessed 2 March 2025].
- [31] Pennsylvania Department of Environmental Protection, "Maximum Contaminant Levels, Maximum Residual Disinfectant Levels, and Treatment Technique Requirements," March 2024. [Online]. Available:
<https://greenport.pa.gov/elibrary//PDFProvider.ashx?action=PDFStream&docID=1675228&chksum=&revision=4&docName=MAXIMUM+CONTAMINANT+LEVELS%2C+MAXIMUM+RESIDUAL+DISINFECTANT+LEVELS%2C+AND+TREATMENT+TECHNIQUE+REQUIREMENTS>. [Accessed 24 April 2025].
- [32] National Archives, "Code of Federal Regulations, Title 40, Part 143, Subpart A," 2020. [Online]. Available: <https://www.ecfr.gov/current/title-40/chapter-I/subchapter-D/part-143/subpart-A> . [Accessed 2 March 2025].
- [33] "IEEE Guide for EMF Exposure Assessment of Internet of Things (IoT) Technologies and Devices, IEEE Standard 1528.7," 2020. [Online]. Available:
<https://standards.ieee.org/ieee/1528.7/7306/>. [Accessed 2 March 2025].
- [34] Texas Instruments, "TMUX13xx 5-V, Bidirectional 8:1, 1-Channel and 4:1, 2-Channel Multiplexers with Injection Current Control, TMUX1309 Datasheet," October 2022. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tmux1309.pdf>. [Accessed 2 December 2025].

- [35] Texas Instruments, "TLV914x 18V, Rail-to-Rail Input and Output, 125kHz Low Power (7µA/Channel) Operational Amplifier, TLV9144 Datasheet," December 2024. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tlv9144.pdf>. [Accessed 27 November 2025].
- [36] Microchip Technology, "7/8-Bit Single/Dual I2C Digital POT with Volatile Memory, MCP4651 Datasheet," 2013. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/22096b.pdf>. [Accessed 27 November 2025].
- [37] Microchip Technology, "16-Bit Analog-to-Digital Converter with I2C Interface and On-Board Reference, MCP3425 Datasheet," 2009. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/22072B.pdf>. [Accessed 27 November 2025].
- [38] Atmel Corporation, "8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, ATMEGA328P Datasheet," 2015. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. [Accessed 27 November 2025].
- [39] Espressif Systems, "ESP32-WROOM-32 (ESP-WROOM-32) Datasheet," 2018. [Online]. Available: https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf. [Accessed 27 November 2025].
- [40] Raspberry Pi Ltd., "RP2350 Datasheet, A Microcontroller by Raspberry Pi," 2023-2025. [Online]. Available: <https://pip-assets.raspberrypi.com/categories/1214-rp2350/documents/RP-008373-DS-2-rp2350-datasheet.pdf?disposition=inline>. [Accessed 27 November 2025].
- [41] Raspberry Pi Ltd., "The C/C++ SDK," 2025. [Online]. Available: https://www.raspberrypi.com/documentation/microcontrollers/c_sdk.html. [Accessed 27 November 2025].

- [42] Raspberry Pi Ltd., "Hardware Design with RP2350: Using RP2350 microcontrollers to build boards and products," 2024. [Online]. Available: <https://datasheets.raspberrypi.com/rp2350/hardware-design-with-rp2350.pdf>. [Accessed 27 11 2025].
- [43] Energizer Brands, LLC., "Alkaline Manganese Dioxide Handbook and Application Manual," 2018. [Online]. Available: https://data.energizer.com/pdfs/alkaline_appman.pdf. [Accessed 27 November 2025].
- [44] Murata Electronics, "CHIP COIL (CHIP INDUCTOR) for Automotive infotainment/comfort equipment LQH3NPZxxxxMEx Murata Standard Reference Specification [AEC-Q200]," [Online]. Available: <https://search.murata.co.jp/Ceramy/image/img/P02/JELF243A-9140.pdf>. [Accessed 27 November 2025].
- [45] Texas Instruments, "TPS6300x High-Efficient Single Inductor Buck-Boost Converter With 1.8-A Switches, TPS6300x Datasheet," 2015. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tps63001.pdf>. [Accessed 27 November 2025].
- [46] Texas Instruments, "LM2660 Switched Capacitor Voltage Converter, LM2660 Datasheet," December 2014. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm2660.pdf>. [Accessed 27 November 2025].
- [47] Waveshare Electronics, "Core1262 LF/HF LoRa Module, SX1262 chip, Long-Range Communication, Anti-Interference, Suitable for Sub-GHz band," 2025. [Online]. Available: <https://www.waveshare.com/core1262-868m.htm>. [Accessed 27 November 2025].
- [48] Semtech, "sx1261/2 Datasheet," December 2024. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/RQ000008nKCH/hp2iKwMDKWI34g1D3LBf_zC7TGBRlo2ff5LMnS8r19s. [Accessed 27 November 2025].

- [49] GeeksForGeeks, "Reliable Data Transfer (RDT) 3.0," 23 July 2025. [Online]. Available: <https://www.geeksforgeeks.org/computer-networks/reliable-data-transfer-rdt-3-0/>. [Accessed 27 November 2025].
- [50] Semtech, "LoRa Calculator," [Online]. Available: <https://www.semtech.com/design-support/lora-calculator>. [Accessed 27 November 2025].
- [51] DFRobot, "Turbidity Sensor SKU SEN0189," [Online]. Available: https://wiki.dfrobot.com/Turbidity_sensor_SKU_SEN0189. [Accessed 27 November 2025].
- [52] J. Wu, "Application Note: A Basic Guide to RTD Measurements," Texas Instruments, 2023.
- [53] Microchip Technology, "MCP3425 Product Listing," 2025. [Online]. Available: <https://www.microchip.com/en-us/product/mcp3425>. [Accessed 27 November 2025].
- [54] J. Wu, "Application Note: A Basic Guide to I2C," Texas Instruments, 2022.
- [55] Macronix International Co., Ltd., "MX25L3233F Datasheet," 19 February 2021. [Online]. Available: <https://www.macronix.com/Lists/Datasheet/Attachments/8933/MX25L3233F,%203V,%2032Mb,%20v1.7.pdf>. [Accessed 27 November 2025].
- [56] Winbond, "3V 128M-BIT Serial Flash Memory with Dual/Quad SPI," 24 December 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/7161/256_W25Q128JV.pdf. [Accessed 27 November 2025].
- [57] Texas Instruments, "LSF0102 2 Channel Auto-Bidirectional Multi-Voltage Level Translator for Open-Drain and Push-Pull Applications, LSF0102 Datasheet," April 2024. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lsf0102.pdf>. [Accessed 27 November 2025].

- [58] MathWorks, "Long Short-Term Memory Networks (LSTM)," MathWorks, [Online]. Available: <https://www.mathworks.com/discovery/lstm.html>. [Accessed 3 December 2025].
- [59] Q. Government, "Colmslie Site Water Quality Monitoring Buoy," 23 May 2025. [Online]. Available: <https://www.data.qld.gov.au/dataset/brisbane-river-colmslie-site-water-quality-monitoring-buoy/resource/0ec4dacc-8e78-4c2a-aa70-d7865ec098e2>.
- [60] JLCPCB, "Multilayer high precision PCB's with impedance control," 2025. [Online]. Available: <https://jlpcb.com/impedance>. [Accessed 28 November 2025].
- [61] Energizer Brands , "Energizer E-95 Datasheet," [Online]. Available: <https://data.energizer.com/pdfs/e95.pdf>. [Accessed 2 December 2025].
- [62] PSemi, "SPDT High Power UltraCMOS 10 MHz-3.0 GHz RF Switch, PE4259 Datasheet," 2023. [Online]. Available: <https://www.psemi.com/pdf/datasheets/pe4259ds.pdf>. [Accessed 27 November 2025].
- [63] Texas Instruments, "TL08xx FET-Input Operational Amplifiers, TL082H Datasheet," September 2025. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tl082h.pdf>. [Accessed 27 November 2025].
- [64] S. Afzal, "I2C Timing: Definition and Specification Guide (Part 2)," 3 October 2016. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/i2c-timing-definition-and-specification-guide-part-2.html>. [Accessed 27 November 2025].
- [65] A. Lincoln, *Gettysburg Address*, 1863.
- [66] Indeed, "Salaries," [Online]. Available: <https://www.indeed.com/career/salaries?from=gnav-title-webapp>. [Accessed 21 April 2025].

[67] Boost High, "How Does Mesh Network Allow IoT Devices To Communicate?," [Online]. Available: <https://boosthigh.com/mesh-network-for-iot-devices/>. [Accessed 10 December 2025].

Appendix A: Customer Survey

Introduction

Thank you for taking the time to complete this survey. We are a group of Electrical Engineering Students at Penn State Harrisburg. Our idea for our senior capstone project is to use a network of wireless battery-operated sensors to autonomously collect real-time data on a variety of water quality metrics. The telemetry data will be communicated wirelessly back to a central base station, where we will utilize AI-driven analytics to provide insights and make predictions about the health of the system and more efficiently allocate water resources.

Thank you for your time and expertise!

Section 1 - About You:

1. Job Title:

2. Company:

Section 2: Current Process

1. **How You Monitor Water Quality**
 - Which tools or tests do you use now to check water quality (for example, pH meters, turbidity, chlorine tests)?

 - How often do you collect or check water quality data?

 - What are some challenges you face with your current water quality monitoring system?

Section 3: Design Specifications

We plan to implement a network of wireless sensors that monitor a variety of parameters contributing to overall water quality. This data will then be transferred to a central hub which can then process that data and generate alerts as needed.

Sensor Features

1. On a scale from 1-10, with 10 being extremely important, how important would it be for the sensor equipment be battery operated?

1 2 3 4 5 6 7 8 9 10

2. On a scale from 1-10, with 10 being extremely important, how important is it for each sensor node to have a long battery life (>10 years of operation). If being battery operated is not important or undesirable, leave this question blank.

1 2 3 4 5 6 7 8 9 10

3. How often would you be willing to replenish batteries the sensor nodes? If battery operation is not important or undesirable, leave this question blank.

Every: Year 1-3 years 3-5 years 5-7 years 7-10 years 10-15 years 15+ years

4. On a scale from 1-10, with 10 being extremely helpful, how helpful would it be if the sensors could send data from far away (even across large facilities) without needing access to Wi-Fi, Cell Service, or a Wired Connection?

1 2 3 4 5 6 7 8 9 10

5. Please rank the following water quality metrics, with rank 1 being the highest importance to be monitored:

- | | |
|--|--|
| <input type="checkbox"/> Turbidity | <input type="checkbox"/> Dissolved Oxygen |
| <input type="checkbox"/> pH | <input type="checkbox"/> Temperature |
| <input type="checkbox"/> Conductivity | <input type="checkbox"/> Total Dissolved Solids |
| <input type="checkbox"/> Chlorine Concentration | <input type="checkbox"/> Contaminant Concentration (PFOA/PFOS) |
| <input type="checkbox"/> Other (Please Specify): | |

Base Station

On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to:

1. Provide a dashboard containing real-time data?

1 2 3 4 5 6 7 8 9 10

2. Produce alerts when measured data falls outside of pre-set thresholds?

1 2 3 4 5 6 7 8 9 10

3. Send alerts directly to technicians (text message, phone notification, pager-like device, etc.) to address issues faster?

1 2 3 4 5 6 7 8 9 10

4. Use predictive software tools on water quality measurements to identify and address possible concerns before they arise?

1 2 3 4 5 6 7 8 9 10

5. Integrate with existing PLC systems or other existing controls equipment?

1 2 3 4 5 6 7 8 9 10

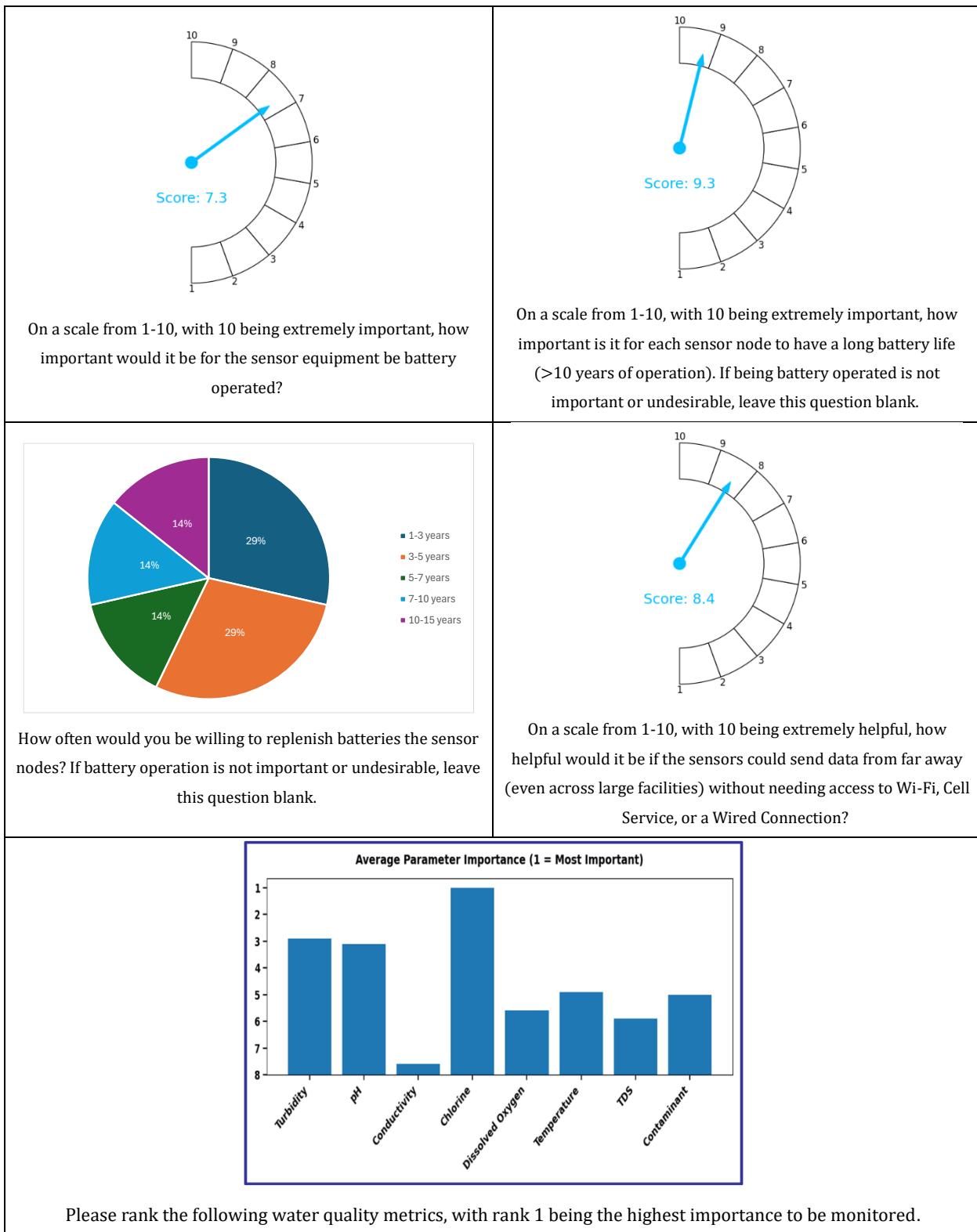
If this is important to you, and you know, what platform/PLC manufacturer is used?

6. If you know, does your team have any budget range or cost limit for new equipment?

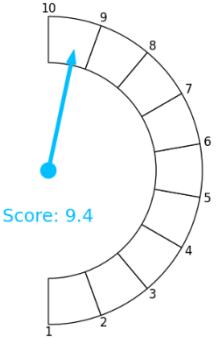
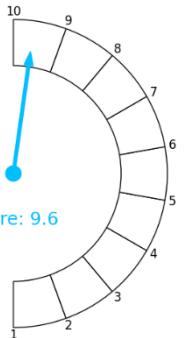
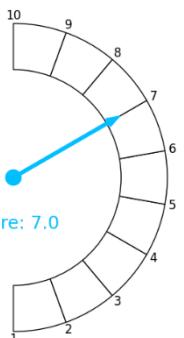
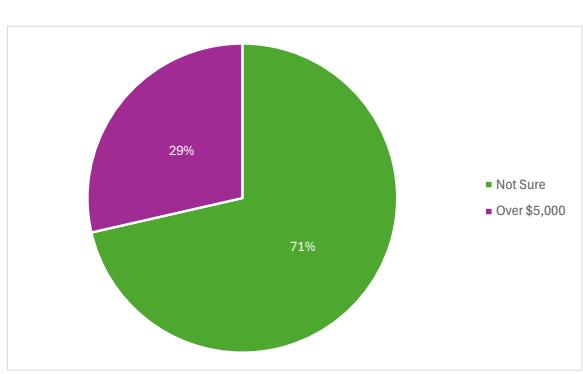
- Under \$1,000 | \$1,000–\$2,500 | \$2,500–\$5,000 | Over \$5,000 | Not sure

Appendix B: Customer Survey Results

Sensor Features



Base Station

 <p>Score: 9.4</p> <p>On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to: Provide a dashboard containing real-time data?</p>	 <p>Score: 9.6</p> <p>On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to: Produce alerts when measured data falls outside of pre-set thresholds?</p>						
 <p>Score: 8.7</p> <p>On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to: Send alerts directly to technicians (text message, phone notification, pager-like device, etc.) to address issues faster?</p>	 <p>Score: 7.0</p> <p>On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to: Use predictive software tools on water quality measurements to identify and address possible concerns before they arise?</p>						
 <p>Score: 7.6</p> <p>On a scale from 1-10, with 10 being extremely desirable, how desirable is it for the system to: Integrate with existing PLC systems or other existing controls equipment?</p>	 <table border="1"> <thead> <tr> <th>Budget Range</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Not Sure</td> <td>71%</td> </tr> <tr> <td>Over \$5,000</td> <td>29%</td> </tr> </tbody> </table> <p>If you know, does your team have any budget range or cost limit for new equipment?</p>	Budget Range	Percentage	Not Sure	71%	Over \$5,000	29%
Budget Range	Percentage						
Not Sure	71%						
Over \$5,000	29%						

Appendix C: Sensor Node Firmware & PCB Design Access

Complete access to the sensor node's firmware, schematic, and PCB layout can be obtained through the project's GitHub Repository:

<https://github.com/msharp176/LoRa-Water-Quality-Management-System>

The folder structure of the repository is as follows:

- Gateway: Code & PCB Designs for the Gateway
 - Dashboard: Python Code for the system dashboard
 - LoRa Receiver: PCB Design, RP2350 Firmware, and Console Port Logging Python Script for the dedicated LoRa receiver module
 - Neural Network: Code and other information for Neural Network Implementation
- Prototyping and Testing: Various PCBs and Code used during the prototyping and initial testing phases of this project
- Sensor Node: Master Directory for Sensor Node PCB, firmware, documentation, and test results
 - Code: Complete, functional firmware for the Sensor Node
 - Bin: Final Binary Executable for RP2350 microcontroller (.uf2)
 - Lib: Custom-Implemented drivers for the sensor node, written in C
 - Sx126x: Semtech-Provided driver files, with custom-written HAL and radio interrupt service routines
 - Stash: Top-Level firmware files for the node and gateway. Not included in compilation unless contents of one file is pasted in main.c
 - Sys_Utils: Node-level abstracted layers of code (e.g. LoRa implementation, RDT3.0 implementation), custom-written for this project.
 - Tests: Code used for testing various components and custom drivers.
 - Docs: Datasheets, PCB Design Writeup, and Annotated Bill-of-Materials (BOM) for the final PCB design
 - PCB Layout: Full KiCAD implementation of PCB design
 - Simulation: Simulation Files and Results for the various simulations performed for the project

Appendix D: Test Plan Tables

Test Author:	Matthew Sharp							
Test Name:	Digital Potentiometer (Digipot) Operation			Test ID #:	001			
Description:	Operational Test of the MCP4651 Digital Potentiometer			Test Type:	Component			
Name of Tester:	Matthew Sharp			Date:	9/10/2025			
Hardware/Software Version:	N/A			Time:	1:00 PM			
Test Setup: $V_{DD} = 5V$, P0A/P1A connected to V_{DD} , P0B/P1B connected to GND. Voltage at the wiper pins (P0W/P1W) measured using a Keithley 2000 DMM with respect to GND. Pass criterion is within 0.5% of calculated wiper voltage. Wiper setting is selected using I^2C communication from Raspberry Pi Pico 2 through LSF0102 level translator IC with commands in accordance with MCP4651 datasheet, allowing 3.3V logic level compatibility with digipot 5V logic.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	Wiper Setting (0-256)	Wiper-Port B resistance	V_{wiper} (calculated)	$V_{wiper_{P0}}/V_{wiper_{P1}}$				% error 0/1
1	0	0Ω	0V	0.006V/0.006V	X			N/A
2	64	12.5kΩ	1.25V	1.253V/1.245V	X			0.24%/0.4%
3	128	25kΩ	2.5V	2.496V/2.492V	X			0.16%/0.32%
4	192	37.5kΩ	3.75V	3.747V/3.745V	X			0.08%/0.13%
5	256	50kΩ	5V	4.999V/4.998V	X			0.002%/0.004%

6	Disconnected	0Ω	0V	0.0002V/0.0002V	X			Open Circuit Measured Between P0A/P1A and P0W/P1W
---	--------------	----	----	-----------------	---	--	--	---

Test Summary: Digital potentiometer successfully produced reference voltages highly consistent with expectations.

Test Author:	Matthew Sharp							
Test Name:	Level Translator Operation			Test ID #:	002			
Description:	Operational Test of the LSF0102 Logic Level Translator IC			Test Type:	Component			
Name of Tester:	Matthew Sharp			Date:	9/10/2025			
Hardware/Software Version:	N/A			Time:	2:00 PM			
Test Setup: Test setup similar to Figure 8-1 in LSF0102 datasheet. $V_{RefA} = 3.3V$, $V_{RefB} = 5V$. Value of Pull-Up Resistors on input and output varied to measure resulting rise time for both a $100kHz I^2C$ t ($10\mu s$ period) transaction as well as a $400kHz I^2C$ ($2.5\mu s$ period) transaction. Pass condition is 20/80 rise time less than or equal to 10% of the input signal period, to ensure reliable long-term operation of I^2C bus.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	Clock Frequency	Pull-Up Resistance	t_{rise_in}	t_{rise_out}				In %, Out %
1	100kHz	$10k\Omega$	989ns	1948ns		X		9.9%, 19.5%
2		$5k\Omega$	570ns	1094ns		X		5.7%, 10.9%
3		$2k\Omega$	248ns	446.5ns	X			2.5%, 4.5%
4		$1k\Omega$	116.5ns	220ns	X			1.1%, 2.2%
5		500Ω	N/A	N/A		X		I2C bus scan fails at digipot address, ACK

								pulse likely causing issues with low pull-up resistors
6	400kHz	10kΩ	839ns	1161ns		X		5V amplitude not achieved on output due to slow rise time
7		5kΩ	504ns	881.5ns		X		5V not achieved on output
8		2kΩ	226.5ns	446ns		X		8.2%, 16.2%
9		1kΩ	112.5ns	218.5ns	X			4%, 7.9%
10		500Ω	N/A	N/A		X		See 500R comment above

Test Summary:

Maximum 1kΩ resistor to be used on the 5V output side. Maximum 2kΩ pull-up resistor on the 3.3V input side.

Test Author:	Matthew Sharp		
Test Name:	Dual 4:1 Analog Multiplexer Operation	Test ID #:	003
Description:	Operational test of TMUX1309 Dual 4:1 Analog Mux	Test Type:	Component
Name of Tester:	Matthew Sharp	Date:	9/10/2025
Hardware/Software Version:	N/A	Time:	3:00 PM

Test Setup: $V_{DD} = 5V$. Output selection lines are tied to GPIO pins of Raspberry Pi Pico 2. 1Vpkpk, 10kHz Sine, Square, Sawtooth, and 2V DC level applied to the 4 inputs of the multiplexer. Selection of the output is rotated between each input and the output voltage and SNR are measured on the oscilloscope. Pass condition is signal attenuation less than ~5% and SNR of no less than 60dB.

	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	SEL0, SEL1	Input Selected	Output Voltage Attenuation	Output SNR				Noise floor of mux measured at $179\mu V$ RMS.
1	0,0	Sine	5.2%	70dB	X			Though over 5%, within this is still acceptable.
2	0,1	Square	0%	72dB	X			
3	1,0	Sawtooth	2.5%	70dB	X			
4	1,1	DC	0.4%	N/A	X			SNR not valuable as 2V DC level not comparable to 1V AC signals.
5	EN# = HIGH	NONE	N/A	N/A	X			No output observed. Expected behavior.

Test Summary: Test demonstrated that the TMUX1309 analog multiplexer can effectively select between our sensor inputs. Some highly attenuated signals observed at output when output disabled, will be caught by passive RC filter at output in application.

Test Author:	Matthew Sharp							
Test Name:	Switched Capacitor Operation			Test ID #:	004			
Description:	LM2660 Switched Capacitor Voltage Inverter Functional Test			Test Type:	Component			
Name of Tester:	Matthew Sharp			Date:	10/14/2025			
Hardware/Software Version:	N/A			Time:	8:00 PM			
Test Setup: LM2660 IC connected similarly to Fig. 14 in LM2660 datasheet with additional capacitor on input voltage rail. $V_{IN} = +5V$, $C_1 = C_2 = C_{in} = 100\mu F$. Negative output rail loaded down with varying load resistances to test output voltage drop. Pass condition is a successful voltage inversion with $V_{out} \leq -0.95V_{in}$.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	R_L	I_L	V_{out}	V_{out}/V_{in}				
1	$100k\Omega$	$50\mu A$	-4.99	-0.998	X			
2	$5.1k\Omega$	$0.980mA$	-4.97	-0.994	X			
3	560Ω	$8.93mA$	-4.86	-0.972	X			
Test Summary: LM2660 observed to take a few milliseconds to set up, likely due to set up of oscillation. All test cases shown to keep voltages within passable limits.								

Test Author:	Matthew Sharp		
Test Name:	Buck-Boost Regulator Operation	Test ID #:	005
Description:	TPS63000 Buck-Boost Regulator Operational Test	Test Type:	Component
Name of Tester:	Matthew Sharp	Date:	9/15/2025
Hardware/Software Version:	N/A	Time:	11:00 AM
<p>Test Setup: TPS63000 Buck-Boost Regulator IC Connected in accordance with Fig. 2 from datasheet with corresponding component values in Table 1. A feedforward capacitor was placed in parallel with R_1 according to Eq. 2 in the datasheet.</p>			
<p>To achieve the desired output voltage of 3.3V, the following component values will be used:</p> $R_1 = 560k\Omega, R_2 = 100k\Omega, C_{ff} = 4pF$			
<p>To achieve the desired output voltage of 5V, the following component values will be used:</p> $R_1 = 910k\Omega, R_2 = 100k\Omega, C_{ff} = 2.4pF$			
<p>The input voltage was varied from 1.8 to 5.5V, with a constant load resistor of 12Ω and 18Ω for 3.3V and 5V output, respectively, drawing $275mA$ from the output of the regulator.</p>			
<p>Pass condition is if the measured V_{out} (determined using Keithley 2000 DMM) is within 5% of the expected value.</p>			

	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	Input Voltage	Output Voltage	V_{out}	% difference				
1	1.8V	3.3V	3.25V	1.5%	X			
2	2.5V	3.3V	3.18V	3.6%	X			
3	3.5V	3.3V	3.15V	4.5%	X			
4	4.5V	3.3V	3.15V	4.5%	X			
5	5.5V	3.3V	3.18V	3.6%	X			
6	1.8V	5.0V	5.1V	2%	X			
7	2.5V	5.0V	5.07V	1.4%	X			
8	3.5V	5.0V	5.07V	1.4%	X			
9	4.5V	5.0V	5.05V	1%	X			
10	5.5V	5.0V	5.1V	2%	X			
Test Summary: Correct output voltage observed over full range of input voltages, paralleling the expected range produced by the batteries. Wider deviation observed at 3.3V input, but still within allowable range.								

Test Author:	Matthew Sharp							
Test Name:	16-bit Analog-to-Digital Converter Test			Test ID #:	006			
Description:	MCP3425 Operation			Test Type:	Component			
Name of Tester:	Matthew Sharp			Date:	10/7/2025			
Hardware/Software Version:	N/A			Time:	8:00 PM			
Test Setup: Input of MCP3425 connected to Rigol Power Supply DC output, value read back from ADC compared against measured value on Keithley 2000 DMM. Communication with module performed over I^2C , with commands/data structure formatted in accordance with the datasheet. Pass condition is observed value within 1% of Keithley 2000 DMM								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Test Voltage	Equivalent Code	Voltage Read by ADC	DMM Measured Voltage				
1	0.25V	0x0fa0	0.247438	0.247953	X			0.21% Error
2	0.80V	0x3200	0.794938	0.796589	X			0.21% Error
3	1.15V	0x47e0	1.144625	1.147126	X			0.22% Error
4	1.75V	0x6d60	1.744625	1.74838	X			0.21% Error
5	2V	0x7d00	1.993750	1.99790	X			0.21% Error
Test Summary: This test was a complete success. ADC proven to be precise, returning values all at the same margin of error compared to the actual value. Minimal error detected between ADC reading and Keithley (<0.3%), meaning that this IC will provide highly accurate telemetry for our system.								

Test Author:	Matthew Sharp							
Test Name:	SPI Flash Operation			Test ID #:	007			
Description:	MX25L3233FM2I SPI NOR Flash Operational Test			Test Type:	Component			
Name of Tester:	Matthew Sharp			Date:	9/26/2025			
Software Version:	In development			Time:	4:30 PM			
Test Setup: SPI pins connected to SPI bus of Raspberry Pi Pico 2. HOLD# and WP# connected to VDD. Write/Read/Erase/Power Down operations tested and validated over SPI communication as outlined in datasheet.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Operation Type	Expected Behavior	Observed Behavior					
1	Write Data	Data written stored at selected address	All data successfully written		X			
2	Read Data	Data readback success	Written data read back without error		X			
3	Chip Erase	All bits on chip set to 1	Entire contents of memory set to 0xff		X			
4	Deep Power Down	Chip unresponsive to commands except for wake	ID Read failed when chip powered down, successful readback after wake		X			
Test Summary: Successful operational test of flash IC. All behaviors identical to those outlined in datasheet.								

Test Author:	Matthew Sharp							
Test Name:	LoRa Modulator Test		Test ID #:	008				
Description:	Core1262 HF Module Operational Test		Test Type:	Component				
Name of Tester:	Matthew Sharp		Date:	10/7/2025				
Hardware/Software Version:	N/A		Time:	7:00 PM				
Test Setup: SPI Pins of Core1262 module connected to SPI pins of Raspberry Pi Pico 2. Receive and Transmit Operations tested according to datasheet. 915MHz band monitored using Software-Defined Radio to observe transmitted signal strength. Input current to module monitored using a 1.5Ω shunt resistor at 3.3V power input.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Operation Type		SDR Signal Strength	Transmit/Receive Errors?				
1	Packet Transmit		-55dB	None	X			48.7mA peak TX Current
2	Packet Receive		-55dB	None	X			
3	Deep Sleep Mode		N/A	N/A	X			1.6mA Idle Current, 31.6uA Sleep Current (50x reduction)
Test Summary: LoRa Transceiver Module successfully transmits and receives packets as anticipated. Sleep current measured higher than the datasheet value (160nA), however this module integrates an RF switch into the overall design, likely accounting for the added current consumption.								

Test Author:	Cole Hadley							
Test Name:	pH Sensor Pre-Amplifier Operation			Test ID #:	009			
Description:	Operational Test of the TL082 JFET input op-amp based pre-amplifier for the pH sensor			Test Type:	Subsystem			
Name of Tester:	Matthew Sharp, Cole Hadley			Date:	9/16/2025			
Hardware/Software Version:	N/A			Time:	1:00 PM			
Test Setup: TL082-based pre-amplifier circuit constructed in accordance with project schematic drawings. A single TL082 was replaced with two cascaded TL081s. Since the TL082 is a dual TL081 IC, this design is compatible and suitable for testing (TL082 not available in equipment room). The output of the pH sensor was tied to the input of the subsystem, and the output voltage was measured using a Keithley 2000 DMM.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	pH of solution	Sensor Output Voltage	Theoretical Output Voltage	Measured Output Voltage				
1	4	170 mV	1.66 V	1.76 V	X			5.8% difference
2	7	2 mV	2.00 V	2.11 V	X			5.4% difference
3	11	-290 mV	2.58 V	2.56 V	X			0.78% difference
Test Summary: pH Sensor and Corresponding Pre-Amplifier circuit consistent with theory and simulations. TL082 Operational Amplifier shown to provide reliable buffering of the pH sensor without loading down its high output impedance.								

Test Author:	Matthew Sharp							
Test Name:	Signal Conditioner Operation			Test ID #:	010			
Description:	Complete Operational Test of the Signal Conditioner Subsystem			Test Type:	Subsystem			
Name of Tester:	Matthew Sharp, Cole Hadley			Date:	12/03/2025			
HW/SW Version:	1.0			Time:	12:00 pm			
Test Setup: Differential inputs of turbidity, temperature, and pH sensors connected to inputs of analog multiplexer. Raspberry Pi Pico 2 sets gain, reference, and DC offset values on digital potentiometers over I^2C , as well as selecting the input signal based on the desired reading. Analog voltage then read back on ADC over I^2C , and this value is formatted and interpreted by software on the RP2350 Microcontroller. Pass condition is an estimated pH within 0.3 of actual, turbidity within 0.1NTU for values below 1 NTU, and temperature readings within 0.35°C (Class A RTD tolerance). Turbidities were measured against calibrated and tested solutions using professional Hach Turbidimeter, temperatures were validated against those measured using a meat thermometer, and pH values were tested against buffer solutions with known pH.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	Measurement Type	Input Value	Firmware-Computed Sensor Voltage	Equivalent Measurement Value				
1	Turbidity	0.1 NTU	4.1188V	0.00 NTU		X		Continuation of observed instability in turbidity meter
2		1 NTU	4.1187V	0.00 NTU		X		See above test

3	Temperature	15.5°C	0.111354	15.35°C	X			0.15°C difference
4		36.0°C	0.119511	35.87°C	X			0.13°C difference
5		53.5°C	0.126612	53.74°C	X			0.24°C difference
6	pH	7.0	2.546311	7.26	X			0.26 difference
7		4.0	2.112351	4.11	X			0.11 difference

Test Summary: All measured values for temperature and pH were within the tolerance bands outlined in our engineering requirements. The maximum error observed in both of these sections was 1.8% and 0.4%, respectively. The turbidity measurements, however, were highly inconsistent, with a negligible difference in the output voltage between 0.1 and 1NTU, a performance identical to that seen in prior tests.

Test Author:	Matthew Sharp							
Test Name:	Full Firmware Test			Test ID #:	011			
Description:	Full Test of the LoRa Water Quality Management System Sensor Node Firmware			Test Type:	Subsystem			
Name of Tester:	Matthew Sharp			Date:	12/1/2025			
Software Version:	1.0			Time:	12:00 PM			
Test Setup: Fully assembled Sensor Node PCB connected to battery power. Manual testing of power-savings mode, sensor measurement mode, transmit/receive operations. Sensor node placed in automatic mode, and real-time telemetry, successful mode selection, power consumption, and error conditions observed. Pass condition is successful mode entry/switch, accurate telemetry, successful data transmission, and entry into the dormant state without error.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	State	Expected Behavior	Observed Behavior?					
1	Reset	Sensor Node completes POST and enters active mode	Node POSTs successfully without error. Current draw = 100mA		X			
2	Sample	Sensor node autonomously collects data from each sensor	Node collects data from all sensor inputs according to accuracy established previously. Current draw = 120mA		X			

3	Transmit	Sensor node transmits measured data to gateway	Node transmits packet successfully, received at gateway without error. Current draw = 135mA.	X			Approximately 15mA less than theory
4	Dormant	Sensor node enters power-saving mode	Node enters power saving mode. Current draw = 96mA.		X		More than 400 times higher than theory
Test Summary: Overall, test demonstrated a fully functional firmware design. Power draw less than theoretical in the active state, but significantly higher in the dormant mode, impacting estimated battery life.							

Test Author:	Cole Hadley								
Test Name:	Turbidity Sensor Operation				Test ID #:	012			
Description:	Operational Test of the DFRobot Turbidity Sensor				Test Type:	Component			
Name of Tester:	Cole Hadley				Date:	10/16/2025			
Hardware/Software Version:	N/A				Time:	5:00pm			
Test Setup: Different kaolin–water standards were prepared to span low to high turbidity. Powered by a 5V source, the sensor's output was measured as the voltage across a 4.7 kΩ load resistor. For each standard, the probe was immersed, and a range of voltage readings was recorded over a 2-minute period. Output voltage was measured with a Keithley 2000 DMM. Actual NTU was verified using a Hach 2100P turbidimeter.									
	INPUTS		OUTPUTS			Pass	Fail	N/A	Comments
Test	Turbidity	Actual Turbidity (Avg)	Sensor Output Voltage (Start → End, 2 min)		ΔV (mV)				
1	0.2 NTU	0.2 NTU	4.117 → 4.116		-2			X	
2	5 NTU	7 NTU	4.113 → 4.107		-6			X	
3	25 NTU	27 NTU	4.101 → 4.086		-15			X	
4	50 NTU	60 NTU	4.049 → 4.043		-6			X	
5	100 NTU	140 NTU	3.996 → 3.966		-31			X	
6	300 NTU	330 NTU	3.971 → 3.825		-146			X	Unstable
Test Summary:									

Sensor voltage decreased with increasing NTU as expected. Over the 2-minute measurement window, readings gradually declined possibly due to particle settling. Above approximately 300 NTU, both NTU and voltage became unstable (widespread).

Test Author:	Cole Hadley		
Test Name:	Gateway Dashboard Operation	Test ID #:	013
Description:	Operational test of homebase dashboard which displays node sensor readings.	Test Type:	Subsystem
Name of Tester:	Cole Hadley, Matthew Sharp	Date:	12/03/2025
Hardware/Software Version:	N/A	Time:	12:00 pm

Test Setup: The gateway Raspberry Pi runs the listener script, which receives packets from the Gateway Receiver board and logs pH, turbidity, temperature, timestamp, and node ID values to a CSV file. The dashboard script reads this CSV file, uses the node ID to route each record to the correct node row, and displays the corresponding timestamp and sensor values on the GUI.

	INPUTS		OUTPUTS		Pass	Fail	N/A	COMMENTS
Test	Data Input		Data Displayed?					
1	pH		Yes		X			
2	Turbidity		Yes		X			
3	Temperature		Yes		X			
4	Timestamp		Yes		X			

Test Summary:

All fields were correctly logged to the CSV file. The dashboard used the node ID to route data to the appropriate node row and correctly displayed pH, turbidity, temperature, and timestamp values. Test passed.

Test Author:	Matthew Sharp							
Test Name:	LoRa Range Test			Test ID #:	014			
Description:	Range test of packet transmission with sx1262 LoRa module			Test Type:	Component			
Name of Tester:	Matthew Sharp, Cole Hadley			Date:	9/29/2025			
Hardware/Software Version:	N/A			Time:	3:00 PM			
Test Setup: sx1262 LoRa transceiver configured to spreading factor 10 (SF10), 125kHz bandwidth, implicit header, 8 symbol preamble, and +22dBm output power. Range tested according to theoretical range calculated here: https://www.semtech.com/design-support/lora-calculator . Tests performed according to https://drive.google.com/file/d/1Qt5gT8M0mH9Oey-S0vBJBsi6oiEY80 /view?usp=sharing (import .KMZ file into google earth).								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Location	Distance to Receiver	Packet Received?					
1	Susquehanna Overlook-City Island	550m	Yes		X			
2	City Island-West Fairview Boat Launch	2,230m	Yes		X			
3	Fort Hunter Boat Launch-Heritage Park Boat Launch	1,400m	Yes		X			

Test Summary:

LoRa Range Testing Successful. Highest tested range around 1/2 of theoretical range. Potential for longer range testing time/location permitting. All packets received with no errors.

Test Author:	Cole Hadley							
Test Name:	RTD PT100 Test			Test ID #:	015			
Description:	Operational Test of the RTD PT100 temperature sensor			Test Type:	Component			
Name of Tester:	Cole Hadley			Date:	10/8/2025			
Hardware/Software Version:	N/A			Time:	12:00pm			
Test Setup: The RTD will be connected to a $4.7\text{k}\Omega$ resistor and 5V source forming a voltage divider. The RTD will be placed in water of various temperatures, and the output of the voltage divider will be measured with a Keithley 2000 DMM.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Ideal Water Temperature	Measure with digital thermometer	Expected Output	Measured Output				
1	0°C	3	105.36mV	107.68mV	X			2.2% Difference
2	25°C	22	112.89mV	114.61mV	X			1.5% Difference
3	100°C	96	141.61mV	143.72mV	X			1.5% Difference
Test Summary: RTD PT100 Test Successful. All measured outputs were less than 5% different between the expected (theoretical) output.								

Test Author:	Matthew Sharp				
Test Name:	RP2350 Power Management		Test ID #:	019	
Description:	Power Management Firmware Library Test for RP2350		Test Type:	Component	
Name of Tester:	Matthew Sharp		Date:	12/1/2025	
Hardware/Software Version:	1.0		Time:	1:00 PM	
Test Setup: RP2350, integrated using Raspberry Pi Pico 2 microcontroller board, is placed on a breadboard with an external LED connected through a 100Ω resistor. The RP2350, running code based on the firmware libraries and HAL used for this project, boots into its active state, blinking an LED. After 10 seconds, the RP2350 enters its dormant state, consistent with P1.7 in the datasheet. In both cases, the current draw from the Rigol DP832A 3-channel DC power supply was monitored by measuring the voltage across a 1Ω shunt using a Keithley 2000 DMM. Pass condition is a dormant current reduction of at least 10 times.					
	INPUTS		OUTPUTS		Comments
Test	Mode of Operation	Measured Current Draw	Power Consumption	Pass	Fail
1	Active	21mA	105mW	X	
2	Dormant	0.72mA	3.6mW	X	29.2x current reduction
Test Summary: Current draw into the RP2350 significantly reduced, with a 29.2 times reduction in current when entering P1.7. Current draw still greater than datasheet value, however, likely due to additional components on board, such as a switching regulator.					

Test Author:	Matthew Sharp		
Test Name:	Full System Integration Test	Test ID #:	020
Description:	Test of the Sensor Node, Gateway, and Dashboard over a multiple hour test period.	Test Type:	System
Name of Tester:	Matthew Sharp	Date:	12/3/2025
Hardware/Software Version:	1.0	Time:	1:00 PM
Test Setup: Sensor node deployed with sample water solution over a multiple-hour period. Validation of packet transmission to gateway performed every 15 minutes according to standards was performed. At each interval, the true values of the solution were compared against the reported values to evaluate the accuracy of the signal conditioner subsystem. Finally, the test was performed over a multiple-hour period to ensure no errors would be observed in a simulated “real-world” environment.			
See Table 28 in Section 3.20 for results.			
Test Summary: Successful full integration test. Data successfully moved from raw sensor inputs in the water onto the node, then transferred wirelessly to the dashboard, demonstrating a fully-integrated prototype.			

Test Author:	Cole Hadley							
Test Name:	Gateway Alert Generation Test		Test ID #:	021				
Description:	Verification of alert generation on the gateway dashboard and email notification when sensor thresholds are violated or a node stops reporting.		Test Type:	Subsystem				
Name of Tester:	Cole Hadley		Date:	12/03/2025				
Hardware/Software Version:	N/A		Time:	12:00 pm				
Test Setup: The CSV log file that the dashboard script reads is manually edited to inject specific test cases. For each test, a row is added or modified to create an out-of-range parameter (pH, turbidity, or temperature) or a timestamp older than the 15-minute timeout. The gateway dashboard script is then run to verify that the status banner updates correctly and that an alert email is generated and received at the configured Gmail account.								
	INPUTS		OUTPUTS		Pass	Fail	N/A	Comments
Test	Alert Condition		Dashboard Banner Updated?	Email Generated?				
1	pH out of range		Yes	Yes	X			
2	Turbidity out of range		Yes	Yes	X			
3	Temperature out of range		Yes	Yes	X			
4	No update from node for more than 15 minutes		Yes	Yes	X			
Test Summary: All alert conditions in the CSV file caused the gateway dashboard status banner to display the correct alert message and triggered an email notification to the test account. The alert generation functionality passed.								

Appendix E: Capstone After-Action Review Matrix

Item	1	2	3	4	Score
Understanding of customer needs	We didn't understand the customer at all	We understood about half of the customer needs	We understood most of the customer needs	We fully understood all customer needs	4
Product delivery based on customer needs	We didn't meet any of the customer's needs	We met about half of the customer's needs	We met most of the customer's needs	We completely met all of the customer's needs	3
Product delivery based on engineering requirements	We didn't meet any of the engineering requirements	We met about half of the engineering requirements	We met most of the engineering requirements	We completely met all of the engineering reqts	3
Use of skills from engineering courses	We did not use anything learned in previous courses	We used minimal skills from courses – 1 course	We used a number of skills from courses – 2-3 courses	We used skills learned in 4 or more previous courses	4
Importance of information literacy (research, etc.)	We did no research at all to accomplish the project	We did minimal research; 1 or 2 sources at the most	We used a number of sources for the project	We were continuously researching (the entire project)	4
Importance of human consulting/networking	We didn't need to consult/network at all	We did minimal human consulting/networking	We regularly consulted or networked	We were consulting almost weekly for the entire project	4
Engaging in lifelong learning as an individual	I see absolutely no need for continuing to learn	There may be a few things I need to learn in the future	As jobs change, there will be many things to learn	My life as an engineer will involve continuous learning	4
Lifelong learning as a team (If applicable)	I see no value in the team for lifelong learning	There are a few things I can learn from team members	There are many things to learn from team members	The team is critical for life-long learning in the future	4
Functioning as a team (if applicable)	Our team was completely dysfunctional	We managed as a team (barely) – team roles were confusing	The team functioned well most of the time	This was the best team experience I've ever had	4

Appendix F: Sensor Node PCB Schematic

The following pages contain the full schematic-level implementation of the sensor node used for the LoRa Water Quality Management System.