

Chapter 3

Methodology

3.1 System Design and Overview

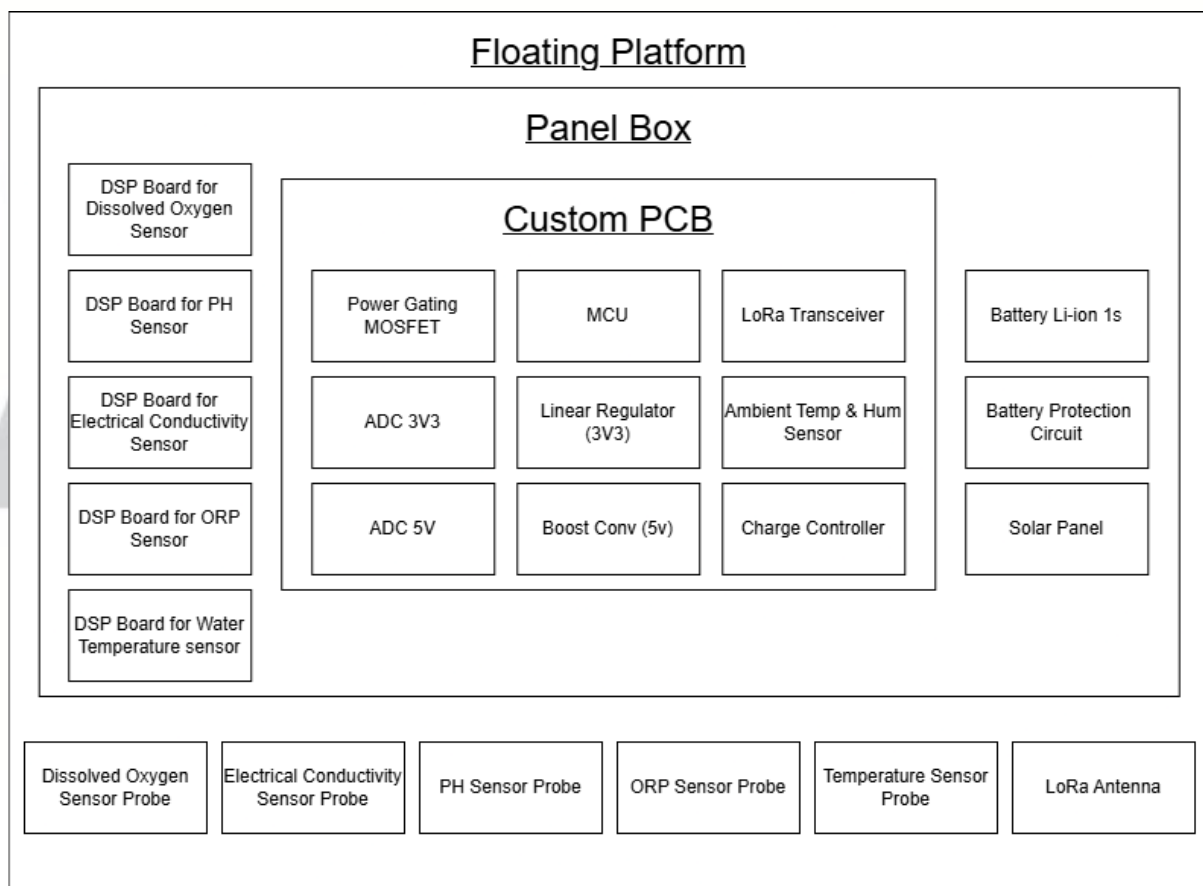


Figure 3.1: System Overview

The IoT system is designed for water quality monitoring in shrimp farming, providing real-time data critical for sustainable aquaculture. The design integrates four main subsystems, each addressing a specific functional requirement:

1. **Sensor Reading:** The system measures vital water quality parameters using five sensors:

pH, electrical conductivity (EC), dissolved oxygen (DO), oxidation-reduction potential (ORP), and battery voltage. Two Analog-to-Digital Converter (ADC) modules are employed because the available ADC channels (four per module) necessitate a dual configuration. One ADC operates at 3.3V to support the pH, EC, and DO sensors, while a second ADC operates at 5V for the ORP sensor and battery voltage measurement (the latter of which can reach up to 4.2V). Additionally, a water temperature sensor communicates via a 1-wire interface, and an ambient air temperature and humidity sensor is interfaced via I2C, both directly connected to the main Microcontroller Unit (MCU).

2. **LoRa Transmission:** For long-range, low-power wireless communication, the system integrates an SX1276 LoRa transceiver connected via SPI to the MCU. Although its selection was largely based on availability, the SX1276 meets the required range and low power criteria while adhering to regulatory compliance, as it is designed for commercial node deployment.
3. **Power Management:** To ensure uninterrupted operation in remote conditions, the system employs a small solar panel coupled with a charge controller to maintain a 1s Li-ion battery. The battery is sized to provide at least one week of capacity even for earlier prototypes with conservative safety margins. Power management strategies include the use of deep sleep modes for the ESP32 and a MOSFET-based power gating solution that switches off non-essential peripherals during idle periods to maximize energy conservation.
4. **Waterproof and Floating Platform:** The physical design incorporates a standard 2.5 kg buoy as a floating platform, ensuring stability and durability in aquatic environments. The enclosure is an IP-rated panel box, and all cable glands are similarly rated, guaranteeing reliable sensor connections and system integrity under harsh environmental conditions.

3.2 System Breakdown Structures

A system workflow diagram is included to visualize the work and consideration required to create the system.

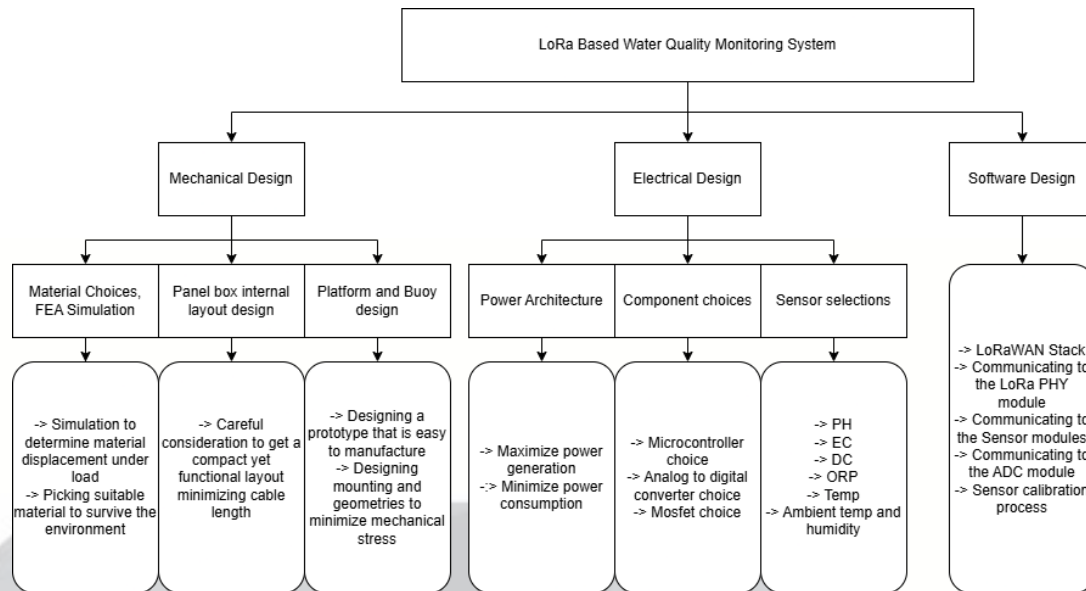


Figure 3.2: System Breakdown Diagram

3.3 Mechanical Design and Selection

This section details the mechanical design and selection process for the project and is organized into three main parts:

1. **Assembly Overview and Design:** Presents the overall system and assembly considerations.
2. **Part Design:** Explains the design of individual components, including the rationale behind the material and dimensional choices.

3.3.1 Assembly Overview and Design

The overall assembly was designed using CAD software to ensure optimal integration and functionality of all mechanical components. The design process focused on several key aspects to facilitate reliable operation and ease of manufacturing, including robust material selection, leveraging off the shelf components, and easy assembly process.

The main structure of the assembly is built using 2020 aluminum extrusions, selected for their high strength-to-weight ratio, corrosion resistance, and ease of machining. Tee-nuts and bolts secure the extrusions to form a robust framework that maintains structural integrity under operational loads. This framework is bolted to an acrylic main plate, creating a stable base for sensor integration, while cable ties secure the life buoy to ensure the system withstands mechanical stresses during deployment. Moreover, the aluminum profile incorporates a slanted angle designed to be tangent to the life buoy, allowing it to sit flush against the buoy. This design feature not only provides a mechanical safety net but also reduces the tension on the cable ties.

The main plate serves as the backbone of the assembly, allowing for the attachment of critical components such as sensors, the panel box, and the structural frame. A solar panel is mounted atop the panel box using a nylon spacer to ensure proper clearance and heat dissipation, and epoxy adhesive is used for a secure, long-lasting bond. The panel box is secured to the main plate by 4 Stainless Steel 304 M5 bolts,

Ease of manufacturing was central to the design, with emphasis on using standard components and minimizing custom fabrication where possible. The design facilitates the use of common machining processes and assembly techniques. For example:

- **Modular Design:** The assembly is modular, allowing individual parts to be pre-fabricated and later integrated.
- **Tolerance Analysis:** Critical dimensions have been specified to ensure optimal fit and function, while non-critical dimensions are allowed greater variability to reduce cost and simplify manufacturing. For example, the stringent tolerance required for the main plate justifies the use of a laser cutting process, whereas the dimensions of the aluminum

profile are less critical and can be accurately achieved with simpler, less expensive cutting methods.

- **Ease of Assembly:** Bolt-fastening methods and cable ties have been selected not only for their strength and reliability but also because they allow for straightforward disassembly in case maintenance is required.

Validation and Simulation Prior to finalizing the design, simulations were performed to verify the structural integrity of the assembly under expected load conditions. Finite Element Analysis (FEA) was used to validate that the acrylic main plate would handle both static and dynamic loads without significant deformation. Interference checking in the CAD environment ensured that all components fit together as intended without clashes, reducing potential issues during prototyping and production.

Assembly Views The following figures illustrate various views of the complete assembly to provide a comprehensive understanding of the design:

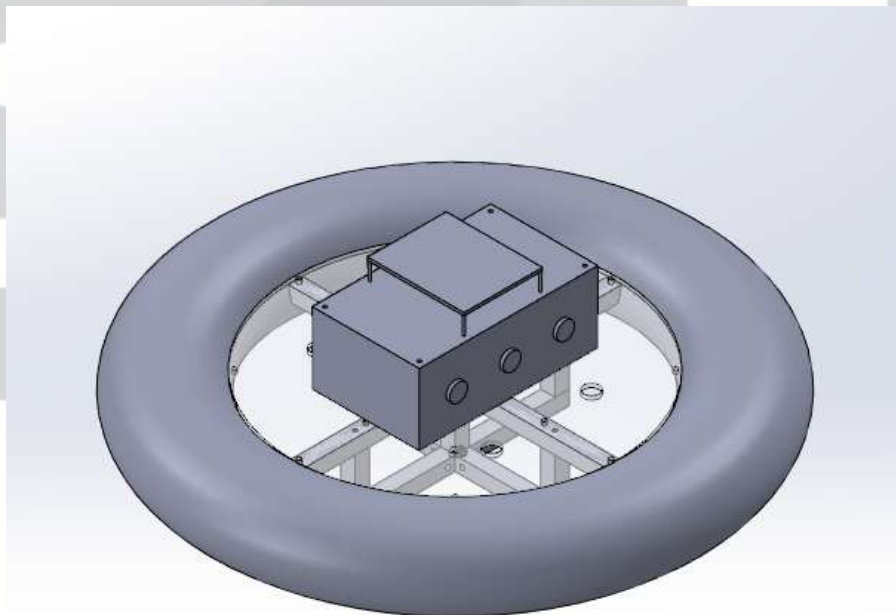


Figure 3.3: CAD Design - Isometric View of the Assembly

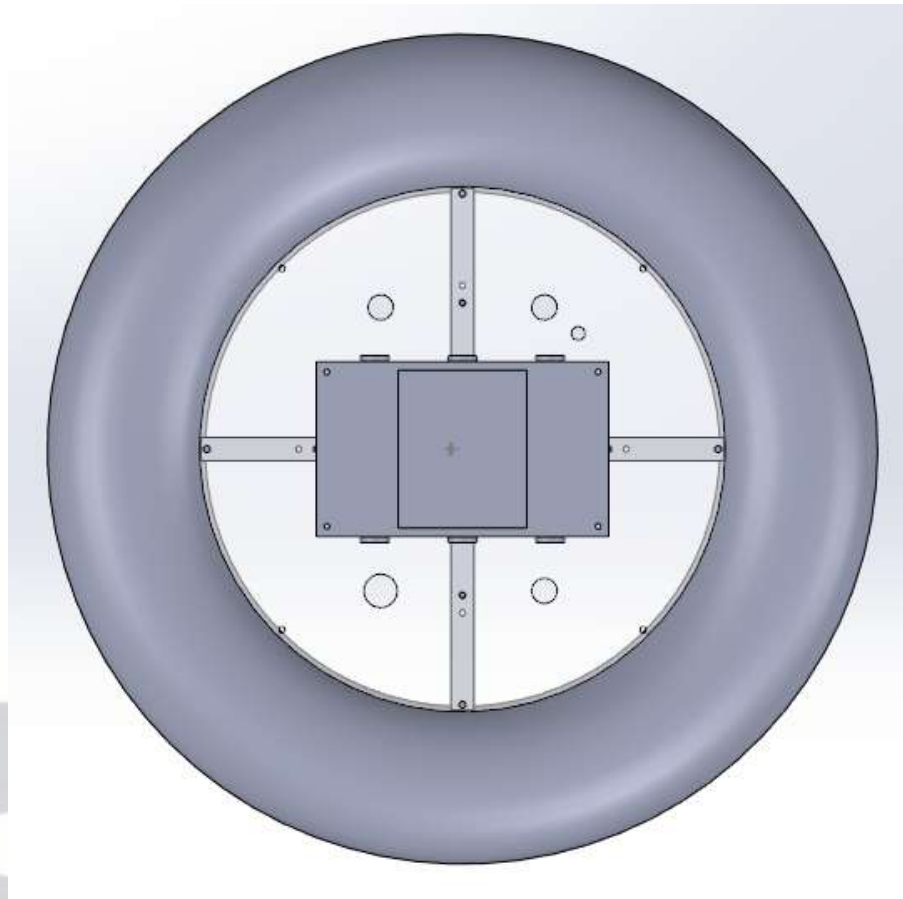


Figure 3.4: CAD Design - Top View of the Assembly

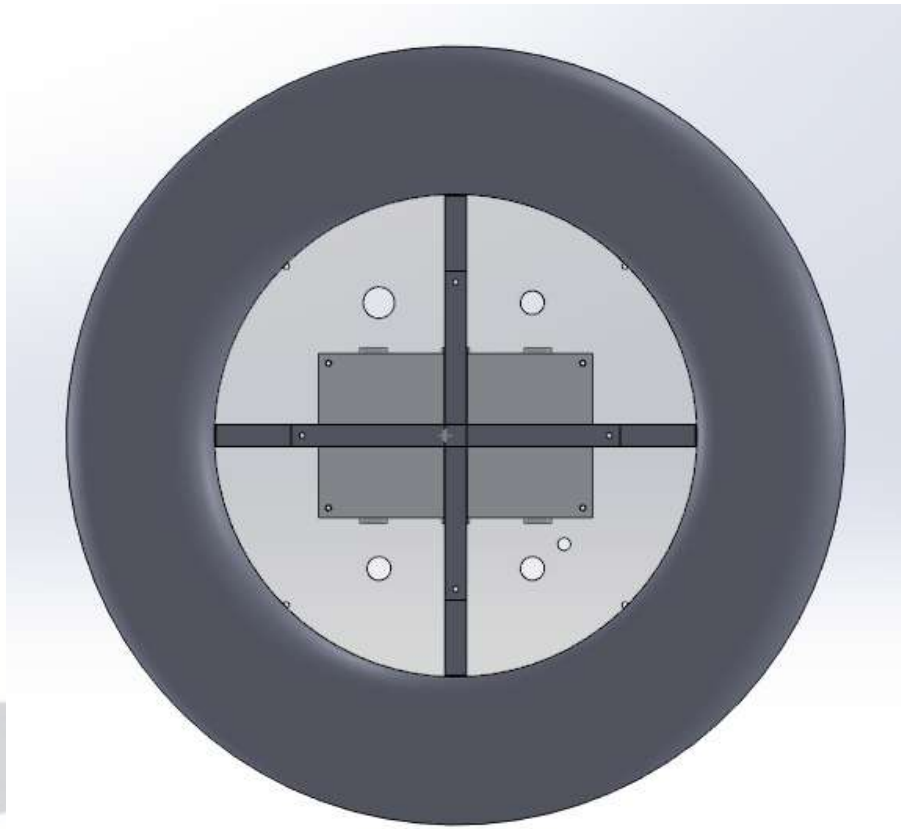


Figure 3.5: CAD Design - Bottom View of the Assembly

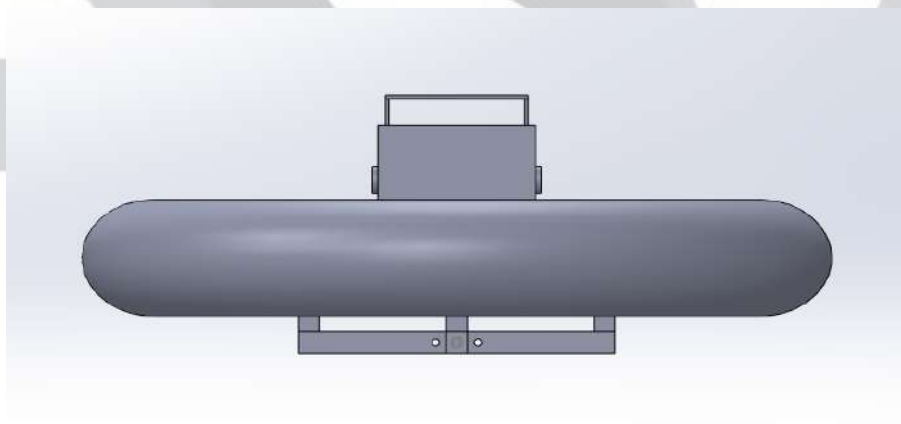


Figure 3.6: CAD Design - Front View of the Assembly

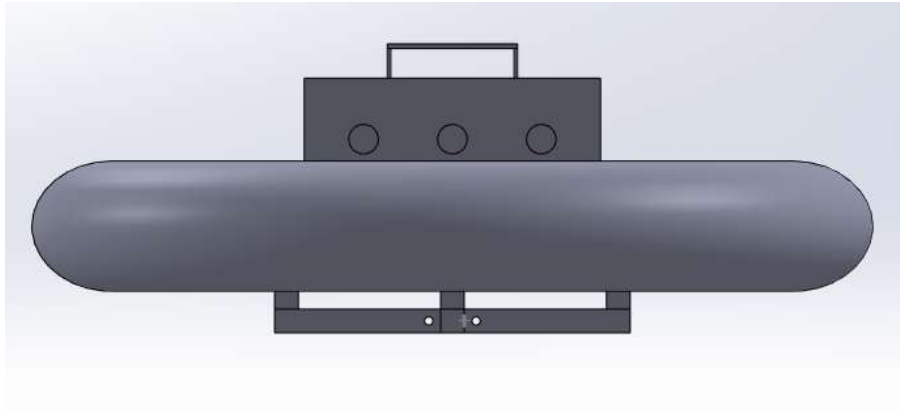


Figure 3.7: CAD Design - Side View of the Assembly

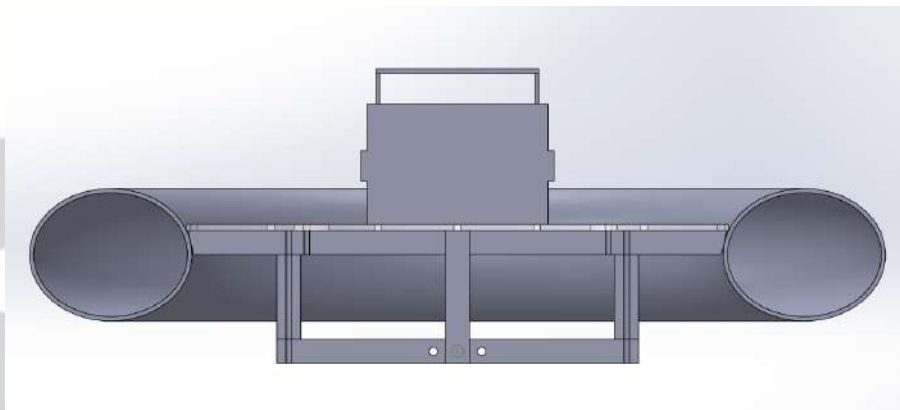


Figure 3.8: CAD Design - Section View of the Assembly

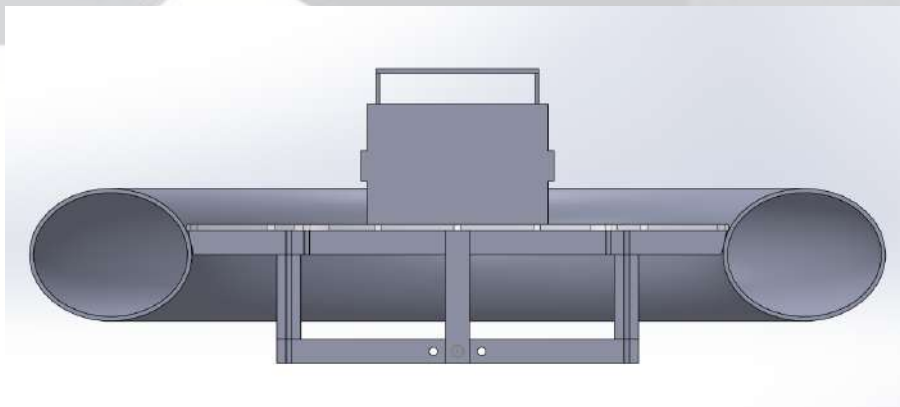


Figure 3.9: CAD Design - Section View of the Assembly

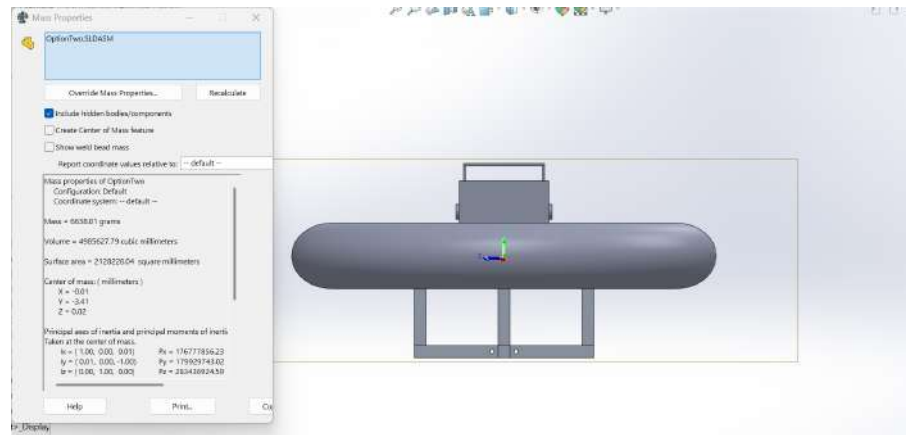


Figure 3.10: CAD Design - Mass and center of gravity estimation on CAD

3.3.2 Part Design

Each component of the assembly was meticulously engineered to meet the specific functional requirements and performance criteria of the system. In this section, individual components are analyzed in terms of their design rationale, material selection, and dimensional considerations.

Main Structure

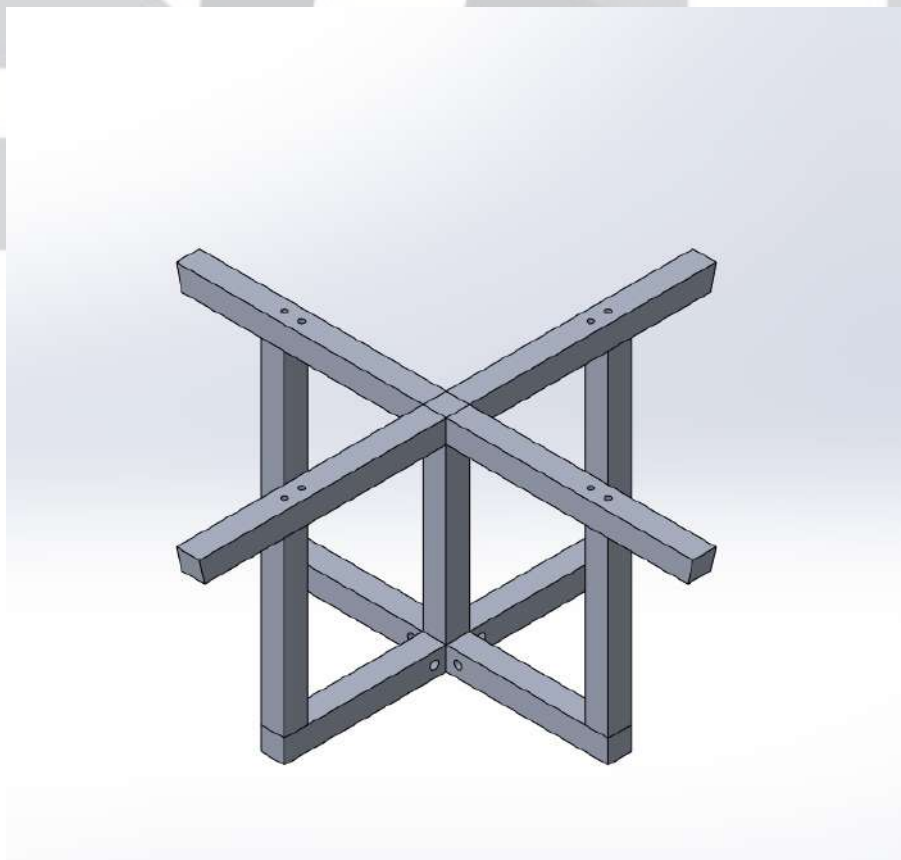


Figure 3.11: CAD design – Isometric view of the main structure

The main structure is fabricated from 2020 aluminum extrusion, selected for its widespread availability, excellent strength-to-weight ratio, and ease of assembly. This component forms the backbone of the entire assembly, providing a rigid framework that supports all mounted sensors and subsystems.

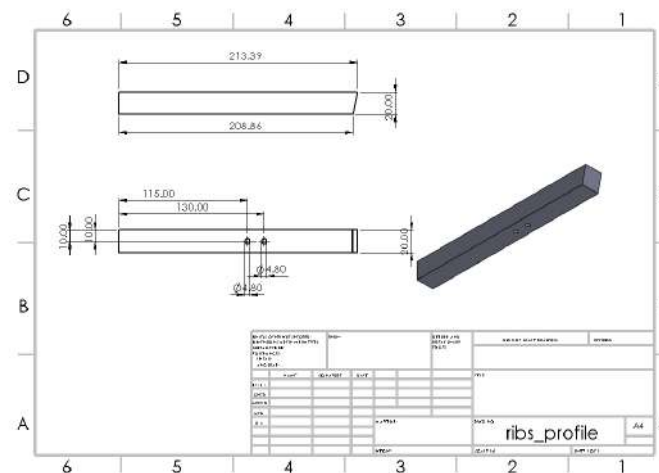


Figure 3.12: CAD drawing – Profile of the ribs

Ribs The rib components, as depicted in the CAD drawings, incorporate a slanted end that is tangent to the life buoy to ensure a flush fit. Two mounting holes are incorporated into the design: the first, located near the straight edge, is a 5.5 mm through-hole intended for an M5 bolt that anchors the main structure to the main plate; the second, situated closer to the slanted edge, facilitates the attachment of cable ties, thereby securing both the main plate and the structure to the life buoy. This second hole also serves as an alignment marker for the feet pieces. The ribs are mechanically connected to the feet and the hook connector using L-brackets, tee nuts, and M5 bolts to ensure a robust and reliable joint.

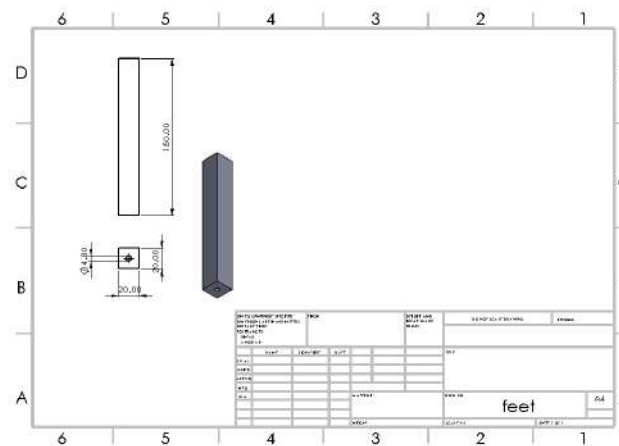


Figure 3.13: CAD drawing – Profile of the feet

Feet The feet are designed to provide sufficient vertical clearance to accommodate the sensor array. A foot length of 150 mm was determined to be optimal based on the spatial requirements. These components are fastened to the rib and cage components via L-brackets, tee nuts, and M5 bolts.

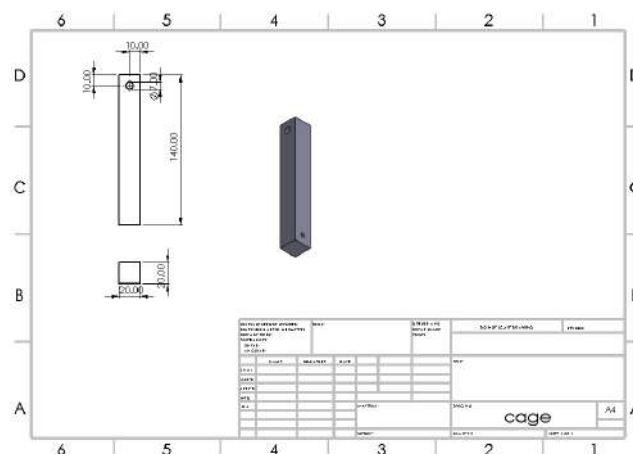


Figure 3.14: CAD drawing – Profile of the cage

Cage The cage serves as a linkage between the feet and the central connector. It features a 7 mm hole that provides an anchoring option for securing the buoy in the pond environment. The cage, integrated with the feet and connector, is assembled using L-brackets, tee nuts, and M5 bolts, ensuring both mechanical stability and ease of assembly.

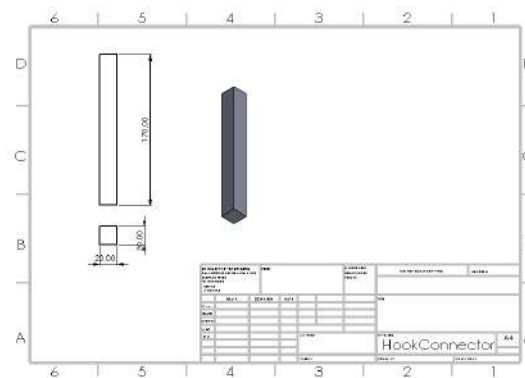


Figure 3.15: CAD drawing – Profile of the connector

Connector Positioned at the center of the main structure, the connector is dimensioned to align with the feet, providing a flat contact surface for placement on the floor. It functions as a critical junction that interlinks the ribs and cage, and is secured using L-brackets, tee nuts, and M5 bolts, ensuring consistent alignment and structural integrity throughout the assembly.

Main Plate

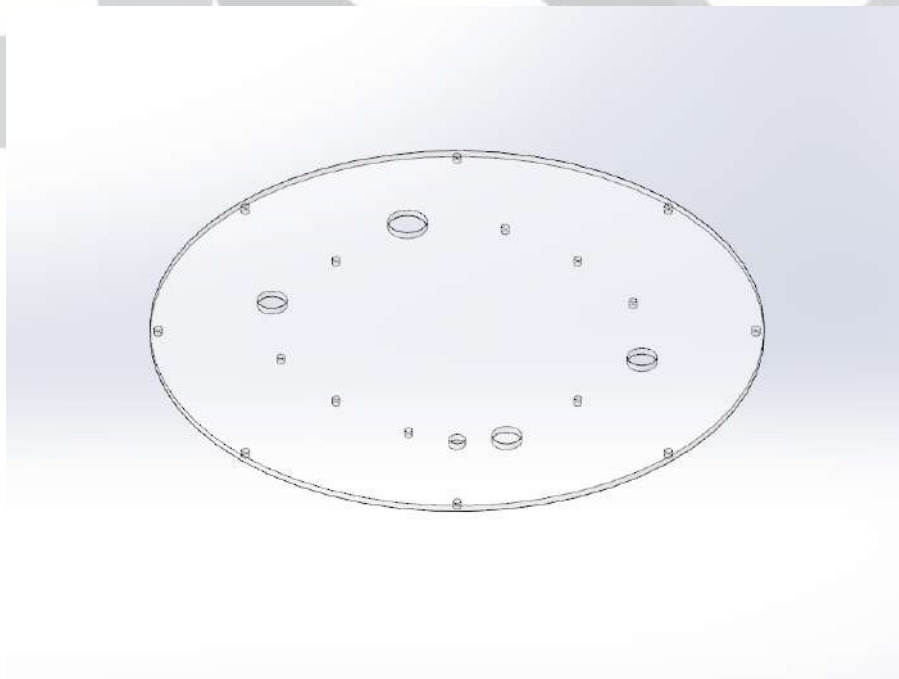
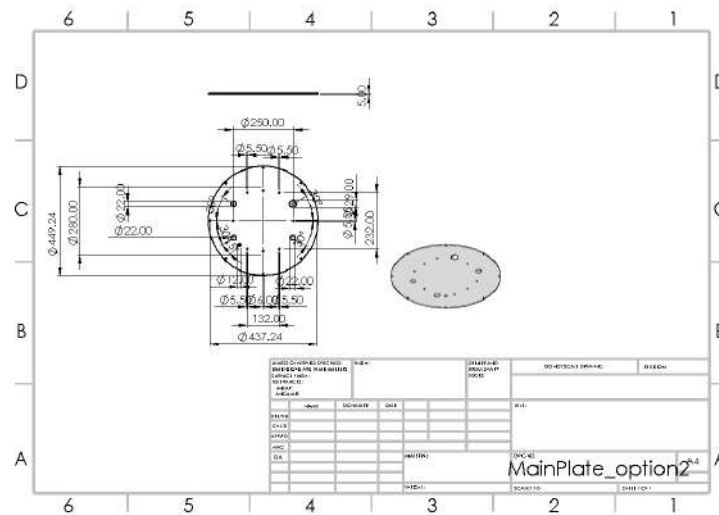


Figure 3.16: CAD Design - Main Plate Isometric view



Design consideration: Mounting ways, dimension Material consideration: FEA Manufacturing process consideration: Laser cut

The primary purpose of the main plate is to provide a platform to place and connect the sensors and the panel box. For this, the main requirement are for the plate to be rigid as to provide a good support for our components, inert as not to disturb the aquaculture, easily precision machined for ease of manufacturing, and cost effective.

To determine the material, a simple finite element analysis (FEA) was conducted to evaluate the rigidity of two candidate materials—stainless steel 304 (SUS304) and acrylic—for use in the system. These materials were considered due to its availability and its compatibility with a laser cutting process. Two thicknesses were considered for each material (2 mm and 5 mm) under the following conditions:

1. A static load of 100 N was uniformly distributed over the top of the plate.
2. The plate was assumed to be rigidly attached to the aluminum main structure.
3. The fixture was modeled as being positioned underneath the aluminum main structure.

Table 3.1 summarizes the key FEA results, including the maximum displacement, maximum von Mises stress, and maximum strain for each material and thickness combination.

Table 3.1: FEA Results for Different Materials and Thicknesses

Material	Thickness	Max Displacement (mm)	Max von Mises Stress (N/m ²)
SUS304	2 mm	8.108×10^{-2}	3.772×10^6
SUS304	5 mm	6.492×10^{-3}	6.587×10^5
Acrylic	2 mm	4.793	6.306×10^6
Acrylic	5 mm	3.535×10^{-1}	1.729×10^6

Material	Thickness	Max Strain
SUS304	2 mm	1.127×10^{-5}
SUS304	5 mm	2.305×10^{-6}
Acrylic	2 mm	6.816×10^{-4}
Acrylic	5 mm	1.352×10^{-4}

Although the FEA indicates that stainless steel is significantly more rigid than acrylic, weight is also an important consideration. Based on the CAD design, the projected weights for the combined main structure and main plate are provided in Table 3.2. Note that these values are estimates due to the simplified CAD model but serve to illustrate the relative differences.

Table 3.2: Projected Weight of Assembly for Different Materials and Thicknesses

Material	Thickness	Weight (grams)
SUS304	2 mm	4797.92
SUS304	5 mm	8546.15
Acrylic	2 mm	2673.93
Acrylic	5 mm	3236.16

In this design, a 5 mm acrylic plate was selected. While stainless steel exhibits superior rigidity, the 5 mm acrylic provides adequate stiffness and is considerably lighter, which is advantageous for overall system weight and handling.

The design also features a series of holes were made to accommodate mounting for panel box, sensors (using cable glands as an adjustable probe holder), main aluminum structure using tee nuts, and some holes for the cables ties used to secure it to the life buoy.

The dimension was designed to be tangent to the life buoy when placed on the main structure.

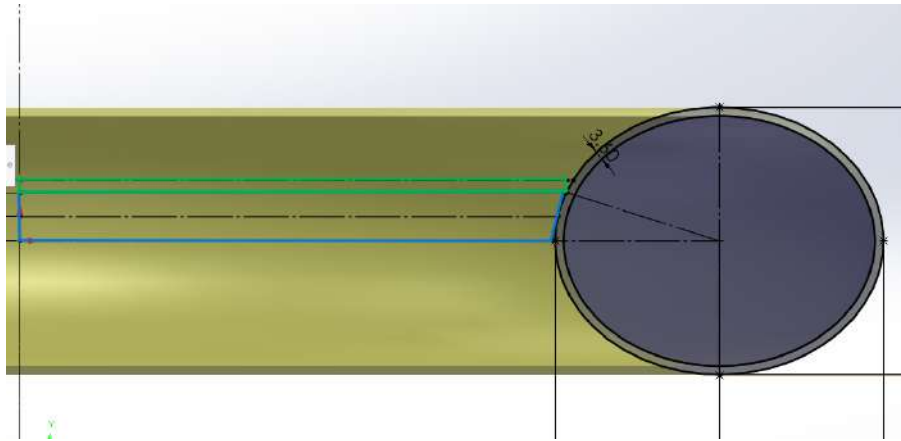


Figure 3.18: CAD Design - Section Geometry of the Life Buoy

Here, green is the main plate and blue is the main structure. This geometry will allow both the structure and the plate to sit snugly on the life buoy.

Panel Box

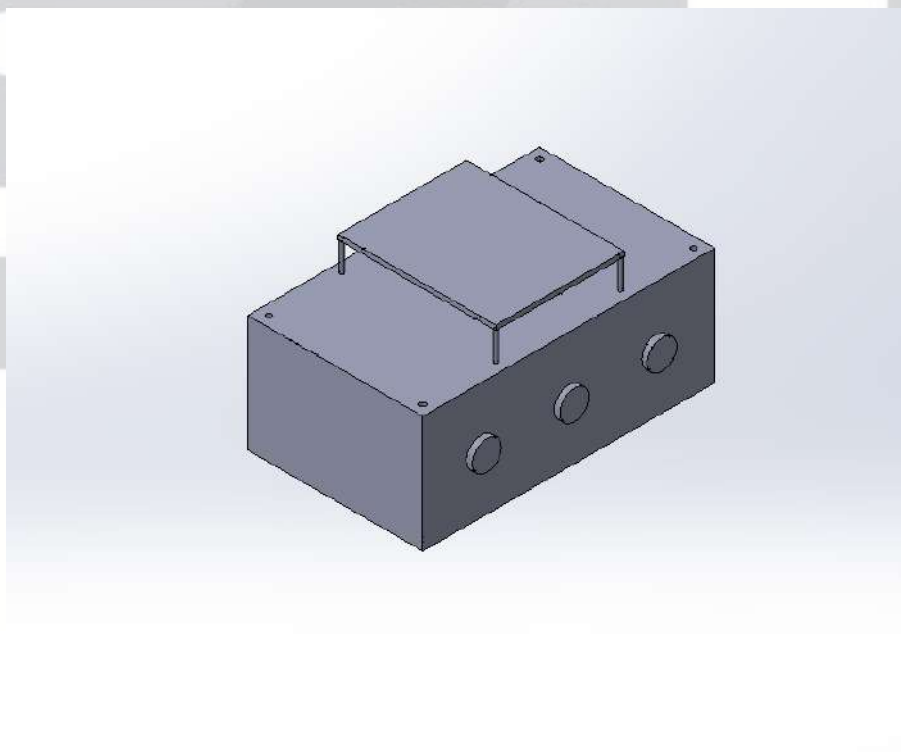


Figure 3.19: CAD design – Isometric view of the panel box

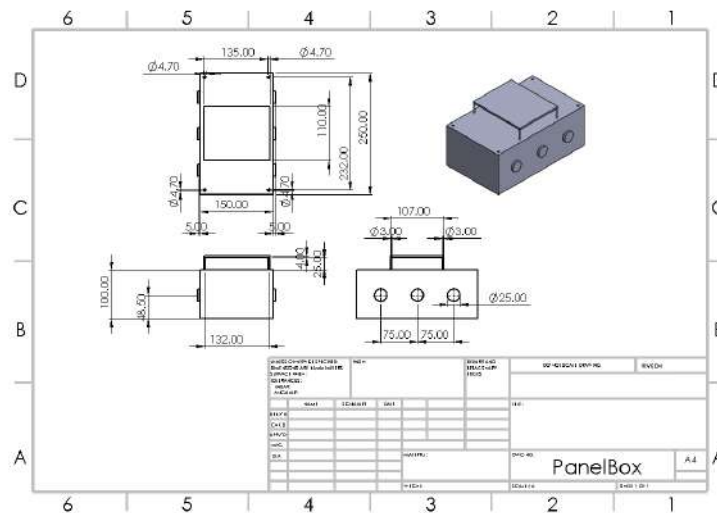


Figure 3.20: CAD drawing – Panel box technical details

The panel box selected for this design is an ABS IP66-rated enclosure from Durabox, with external dimensions of approximately 150 mm × 250 mm × 100 mm. This dimension is chosen to comfortably host the main pcb, batteries, and signal processing boards for the sensor probe. Six precisely cut openings facilitate cable pass-throughs for the sensor probe and antenna, with IP68-rated cable glands ensuring water tightness. Additionally, a compact 1.5 W solar panel is mounted on the top of the panel box using epoxy adhesive and spacers, which help manage thermal loading and maintain proper ventilation.

Life Buoy

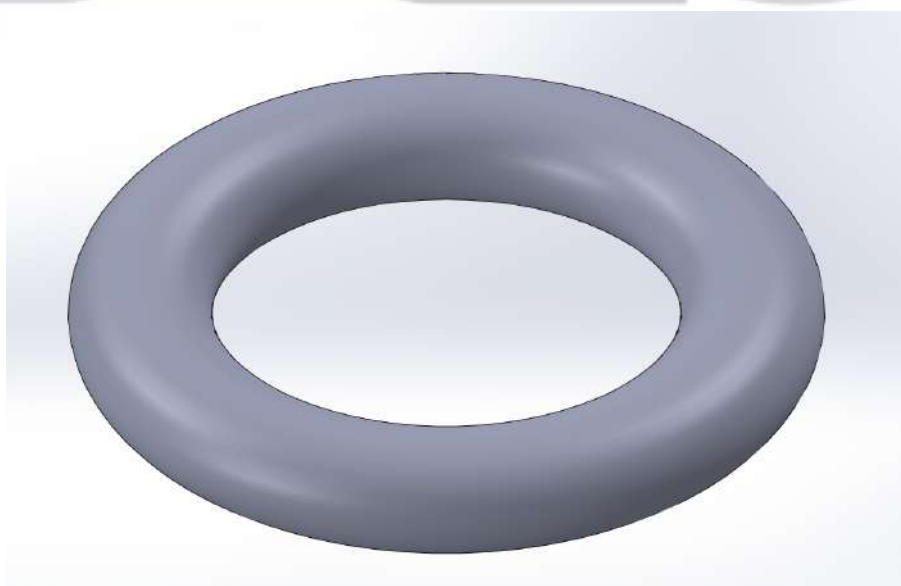


Figure 3.21: CAD design – Isometric view of the Life buoy

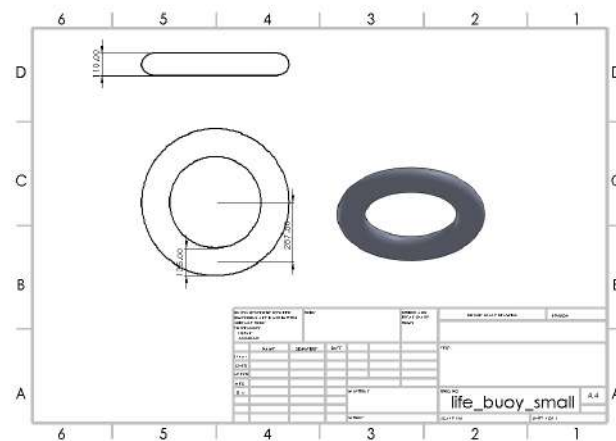


Figure 3.22: CAD drawing – Life buoy technical details

The design incorporates a standard 2.5 kg life buoy for several key reasons:

- **Availability:** Standard life buoys are mass-produced and readily available in small quantities.
- **Cost Effectiveness:** Mass production ensures affordability.
- **Reliability:** These buoys adhere to rigorous safety standards, ensuring dependable performance.
- **Appropriate Buoyancy:** According to standard specifications, a 2.5 kg life buoy provides a minimum buoyancy of 14.5 kg, which is ideal for supporting our system that weighs approximately 6 kg.

Panel Box Internal Layout

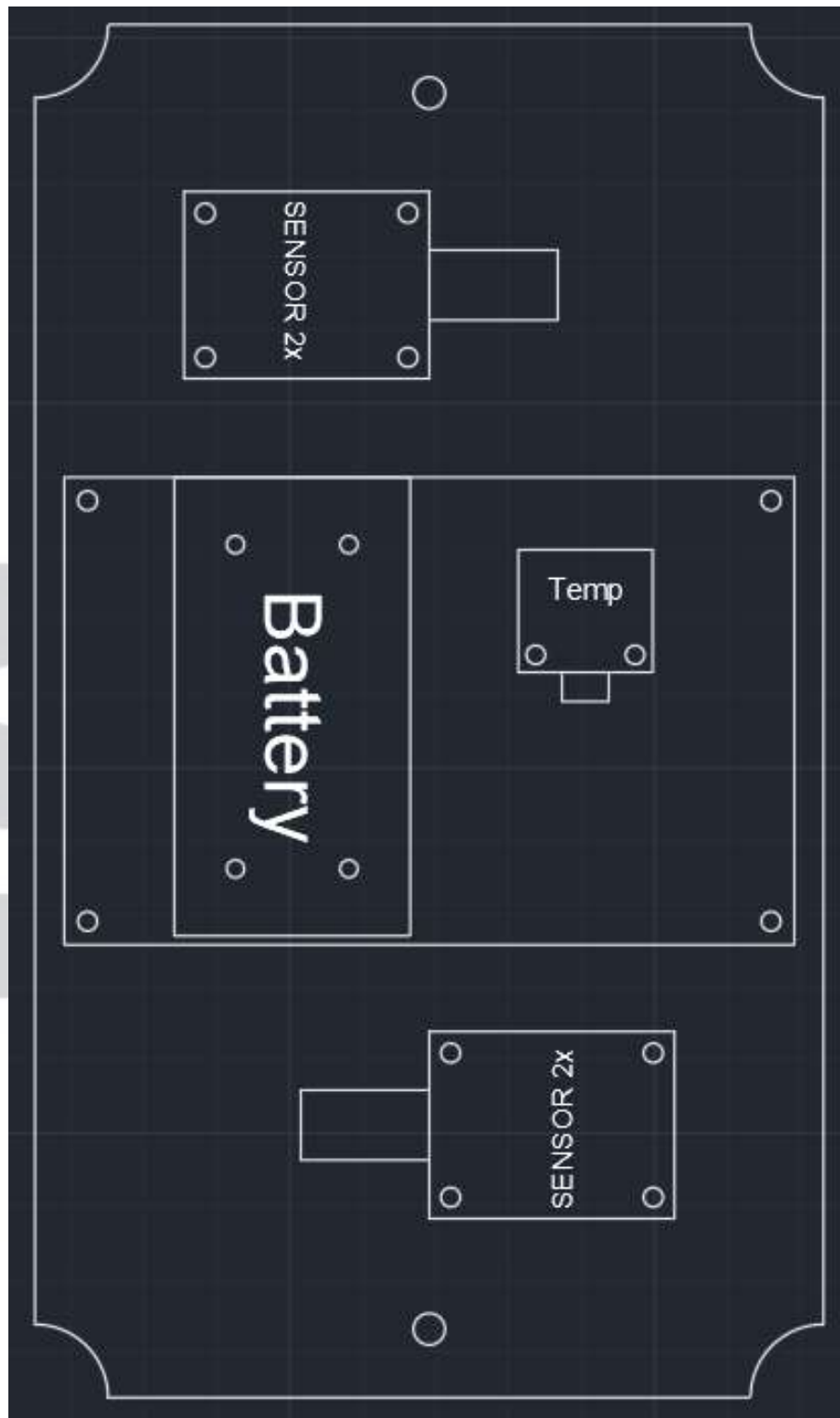


Figure 3.23: CAD design – Internal panel layout

The panel box is designed to house the system's electronic and control components with optimal space efficiency. The internal layout is arranged in two layers:

- **First Layer:** This layer accommodates the battery, the temperature sensor board, the electrical conductivity sensor board, and the dissolved oxygen sensor board.
- **Second Layer:** The ORP sensor board and the pH sensor board are mounted on top of the first layer, with the main PCB positioned above the temperature and battery boards. Nylon spacers are used to elevate the PCB to the appropriate height.

A 5 mm acrylic plate, precision-cut by laser to accommodate the necessary mounting holes, serves as the substrate for securing the components. All fasteners employed in the mounting process are nylon—using M3 fasteners for the individual components and M4 fasteners to affix the acrylic plate to the panel box, preventing unwanted electrical connections while maintaining adequate mechanical strength.



3.4 Electrical Design and Selection

This chapter presents the electrical subsystem of the prototype, including an overview of the architecture, detailed part design by function, and PCB layout with connector choices.

3.4.1 Electrical System Overview

Provide here a block diagram of the entire electrical system, showing power generation, power management, sensors, data acquisition, processing, and communication paths.

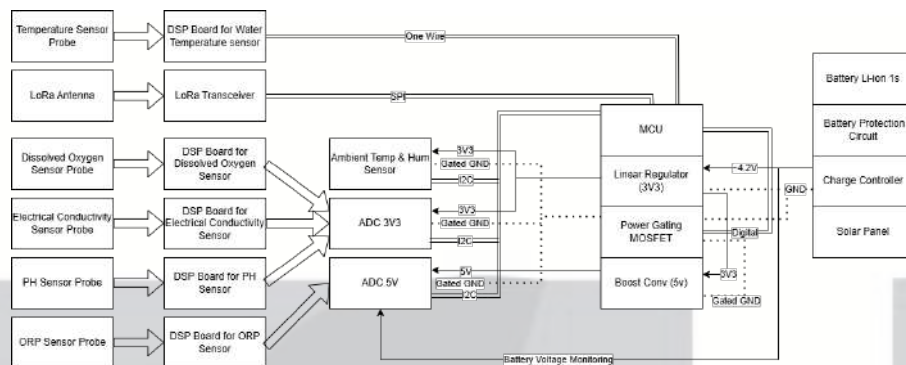


Figure 3.24: Electrical Connection Diagram

Legend:

|| Pair-Lines: Signal

— Solid Lines: Power

Dotted Lines: GND

⇒ Thick Arrow: Signal + Power

Figure 3.24 shows the overall electrical architecture. The ESP32 microcontroller sits at the center, orchestrating the following sequence:

- **Sensor Reading:** External ADS1115 ADCs digitize analog sensor signals (pH, EC, DO, ORP, salinity, and battery voltage) and an I²C-based SHT20 measures ambient temperature and humidity.
- **Data Transmission:** Digitized data are formatted and sent via the SX1276 LoRa module over a LoRaWAN link.
- **Power Management:** A solar panel and CN3065 charge controller maintain a single-cell Li-ion battery. A mini boost converter steps the voltage (≈ 3.3 V) up to 5 V for peripherals, while an LDO provides a stable 3.3 V rail.

- **Power Gating:** An N-channel MOSFET controlled by the ESP32 cuts power to non-essential peripherals (ADCs, sensors, LoRa module, boost converter) during deep sleep, minimizing quiescent current.
- **Sleep Cycle:** After transmission, the ESP32 enters deep sleep, waking at scheduled intervals to repeat the measurement and transmission cycle.

3.4.2 Part Design by Function

Power Generation & Supply

Battery & Protection Circuit

- **Battery sizing:** To estimate the required battery capacity, a simplified prototype was constructed. The measured current consumption at 3.3 V is:
 - **Active mode:** 57.54 mA
 - **Sleep mode:** 12.32 mA



Figure 3.25: Simplified prototype



Figure 3.26: Active state current draw



Figure 3.27: Sleep state current draw

The following assumptions were used:

1. Continuous operation for one week (168 h) without recharge.
2. Prototype consumption approximates the final system.
3. Safety factor of 2 to account for inefficiencies and aging.
4. Maximum active time of 2 min per hour (rest in sleep mode).

Because a linear regulator is employed, the battery current equals the load current. The average current draw is therefore

$$\begin{aligned} I_{\text{avg}} &= I_{\text{active}} \times \frac{t_{\text{active}}}{60} + I_{\text{sleep}} \times \frac{t_{\text{sleep}}}{60} \\ &= 57.54 \text{ mA} \times \frac{2}{60} + 12.32 \text{ mA} \times \frac{58}{60} = 13.83 \text{ mA}. \end{aligned}$$

The capacity needed for one week is

$$C_{\text{week}} = I_{\text{avg}} \times T_{\text{week}} = 13.83 \text{ mA} \times 168 \text{ h} = 2323 \text{ mA h.}$$

Applying the safety factor,

$$C_{\text{required}} = 2 \times C_{\text{week}} = 4646 \text{ mA h.}$$

Accordingly, two Sony 18650 Li-ion cells (each 2600 mA h) connected in parallel—providing a total of 5200 mA h—were chosen to satisfy the requirement while allowing margin for real-world conditions.

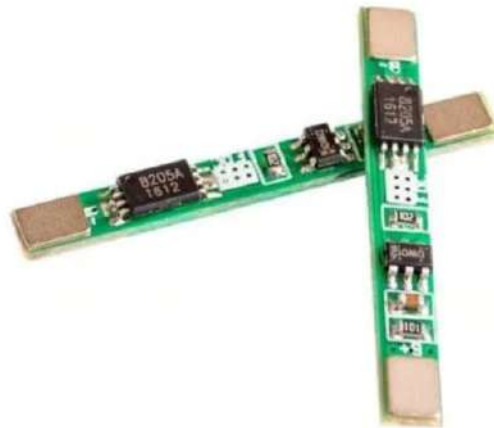


Figure 3.28: Li-ion Protection Circuit PCB

The protection circuit must safeguard a single-cell Li-ion battery against over-charge, over-discharge, and short-circuit/over-current events (3 A). A compact module based on the DW01-P protection IC and FS8205A dual MOSFET was selected.

- **DW01-P single-cell protection IC:**

- Over-charge detect: 4.25(5) V
- Over-charge release: 4.23(5) V
- Over-discharge detect: 2.50(10) V
- Over-current trip threshold: 3 A
- Standby current: <3 μ A

These thresholds and ultra-low quiescent current are specified in the DW01-P datasheet [35].

- **FS8205A dual N-channel MOSFET array:**

- Continuous drain current: 6 A
- Drain-source voltage: 20 V
- On-resistance: 27 m Ω (max at 6 A)

The FS8205A performs the high-current switching as directed by the DW01-P and provides robust over-current protection [36].

The module's small footprint (40 mm×4 mm×3 mm) enables seamless integration into the PCB or enclosure.

Solar Panel The solar panel was selected based on the energy requirements of the system and practical deployment constraints. To determine the minimum energy production needed, the weekly energy consumption was first estimated. From the battery sizing analysis, the system requires approximately 4646 mA h over the course of one week. Assuming a maximum Li-ion battery voltage of 4.2 V, the total energy requirement is:

$$E_{\text{required}} = 4.2 \text{ V} \times 4.646 \text{ A h} = 19.5132 \text{ W h}.$$

A commercially available polycrystalline silicon solar panel was selected as a candidate for evaluation. It is rated at 6 V and 350 mA (2.1 W peak power) and features compact dimensions of 135 mm × 110 mm × 2 mm, making it mechanically suitable for integration with the enclosure.

To validate the panel's real-world performance, it was tested under bright sunlight using a CN3065-based linear charge controller, which regulates current and voltage for single-cell Li-ion batteries [37]. The measured output was:

$$P = V \times I = 6 \text{ V} \times 0.1908 \text{ A} = 1.1448 \text{ W}.$$

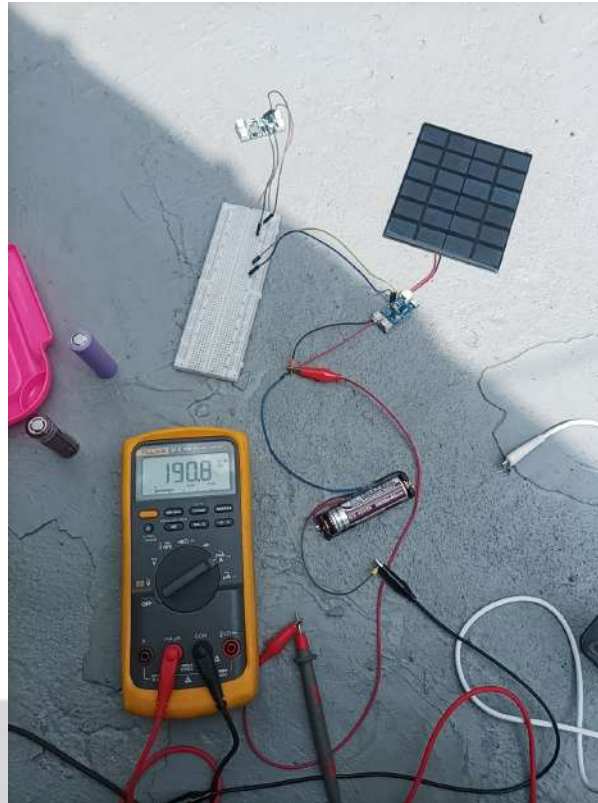


Figure 3.29: Solar panel – initial power output validation

Assuming an average of 5 h of effective sunlight per day, the total weekly energy generation is:

$$E_{\text{generated}} = 1.1448 \text{ W} \times 5 \text{ h/day} \times 7 \text{ d} = 40.068 \text{ W h.}$$

This exceeds the estimated requirement of 19.5 W h, confirming that the selected solar panel is sufficient to sustain continuous operation under typical conditions. Therefore, based on both energy performance and mechanical compatibility, this solar panel was chosen for the final system.

Charge Controller The charge controller must meet two essential requirements: it must safely regulate charging from a solar panel that delivers up to 6 V and 350 mA, and it must support low-power operation suitable for an intermittently powered, energy-efficient embedded system.

A CN3065-based linear charger IC, manufactured by Consonance, was selected for this application. According to its datasheet, the CN3065 supports an input voltage range of 4.4 V to 6 V and delivers a constant charging current up to 500 mA, which comfortably exceeds the panel's maximum expected output [37].

In addition to meeting electrical input and output criteria, the CN3065 includes features that make it particularly well-suited for solar-powered and low-duty-cycle systems. Specifically:

- It integrates a power MOSFET and eliminates the need for external blocking diodes or current-sense resistors, minimizing the BOM and power losses.
- It enters a deep sleep mode when the input voltage is absent, with a standby current draw of less than 3 μA , which helps preserve battery life during extended low-light conditions.

Given its compatibility with the solar panel's characteristics and its built-in low-power operation features, the CN3065 is deemed an appropriate and efficient solution for the system's power management requirements.

Voltage Regulation To meet the voltage requirements of different subsystems, both step-up (boost) and step-down (linear) voltage regulation are implemented in the design.

Boost Converter The boost converter must step up the 3.3 V battery voltage to a regulated 5 V rail and supply up to 500 mA peak, while remaining compact, cost-effective, and readily available.

A mini boost module employing the MT3608 step-up converter IC (Aerosemi Technology Co., Ltd.) was chosen. According to its datasheet, the MT3608 is a constant-frequency (1.2 MHz), current-mode step-up converter in a 6-pin SOT-23 package, designed for low-power, space-constrained applications. Key specifications include:

- **Input voltage range:** 2 V to 24 V
- **Maximum output current:** 2 A
- **Efficiency:** Up to 93%
- **Dimensions:** Approximately 11 mm $\times$ 22 mm

Its ability to deliver up to 2 A comfortably exceeds our 500 mA requirement, providing ample headroom for surge currents.

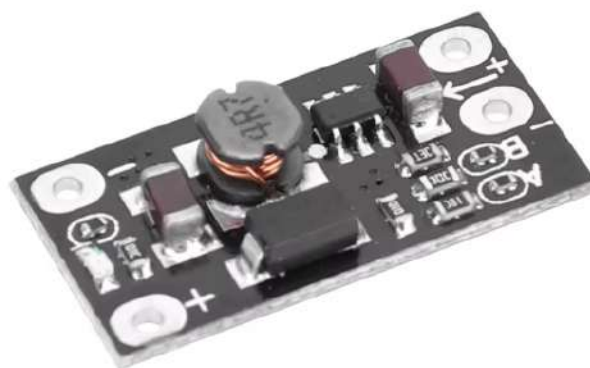


Figure 3.30: Mini boost converter module

Linear Regulator.



Figure 3.31: Firebeetle board from dfRobot

The design also requires a step-down regulator to supply a stable 3.3 V rail to digital components such as the microcontroller and sensor interfaces. Critical requirements include a low dropout voltage, excellent quiescent current performance, and high availability.

To meet these needs, the low-dropout (LDO) linear regulator integrated into DFRobot's FireBeetle board was chosen. It offers a maximum discharge current of 600 mA at 3.3 V, making it well-suited for the expected current load of the digital subsystem. As the FireBeetle platform is explicitly designed for low-power IoT applications, the regulator it employs also demonstrates optimized quiescent current and efficient standby behavior, aligning well with the power-saving goals of this project. [38].

Power Gating MOSFET To minimize quiescent current during deep sleep, all non-essential peripherals (ADS1115 ADCs, SHT20 sensor, SX1276 LoRa module, boost converter, and analog sensors) are switched off via a single N-channel MOSFET. The MOSFET selection criteria included:

- **Low on-resistance** (to minimize voltage drop under load)
- **Logic-level gate threshold** (to fully turn on at 3.3 V)
- **Adequate current rating** (to handle transient currents up to 500 mA)
- **Low gate charge** (to reduce switching energy)

A Hottech SI2300 (SOT-23) was chosen. According to its datasheet [39], the SI2300 provides:

- $V_{DS(max)} = 20 \text{ V}$

- $I_{D(cont)} = 5 \text{ A}$
- $R_{DS(on)} = 22 \text{ m}\Omega$ (typ.) @ $V_{GS} = 4.5 \text{ V}$, $35 \text{ m}\Omega$ (typ.) @ $V_{GS} = 2.5 \text{ V}$
- $V_{GS(th)} = 0.5 \text{ V to } 1.0 \text{ V}$ @ $I_D = 250 \mu\text{A}$
- $Q_g \approx 3 \text{ nC}$

The gate is driven directly by an ESP32 GPIO, the source is tied to ground, and the drain feeds the negative rails of all gated peripherals, effectively disconnecting them during deep sleep.

Energy Buffering A bulk capacitor of $470 \mu\text{F}$ is placed on the 3.3 V rail to buffer against transient inrush currents and peak load demands during power-up of gated peripherals. This value was determined empirically: configurations with smaller capacitance resulted in brownout faults on the ESP32 when peripherals were reactivated. The chosen capacitor maintains stable voltage during startup surges and ensures reliable operation.

Pull-Up and Pull-Down Resistors Four resistors are used to ensure defined logic levels and safe charging/discharging behavior:

1. **Gate series resistor ($1 \text{ k}\Omega$):** Placed between the ESP32 GPIO and MOSFET gate to limit inrush gate current and protect the microcontroller pin.
2. **Gate pull-down resistor ($10 \text{ k}\Omega$):** Tied between MOSFET gate and ground to keep the FET off unless actively driven.
3. **Output bleed resistor ($10 \text{ k}\Omega$):** Connected across the boost converter's 5 V output and ground to discharge output capacitors when power is removed, preventing floating voltages.
4. **I²C address resistor ($4.7 \text{ k}\Omega$):** Used on one ADS1115 module's address pin to set its I²C address, allowing two identical ADCs to coexist on the same bus.

Liquid Sensor Specifications

Table 3.3 summarizes the key specifications of each liquid-phase sensor used in the system, including measurement range, output format, supply requirements, and interface type. Citations refer to the manufacturer's product pages.[40]–[44]

Sensor	Range	Output	Supply	Interface
pH Sensor	0 pH to 14 pH	0 V to 3 V	3.3 V	Analog
EC Sensor	10 mS cm ⁻¹ to 100 mS cm ⁻¹	0 V to 3.2 V	3.3 V	Analog
DO Sensor	0 mg L ⁻¹ to 20 mg L ⁻¹	0 V to 3 V	3.3 V	Analog
ORP Sensor	-2000 mV to 2000 mV	0.5 V to 4.5 V	5 V	Analog
Temp Sensor (DS18B20)	-55 °C to 125 °C	–	3.3 V	1-Wire

Table 3.3: Specifications of sensors

ADS1115 Analog-to-Digital Converter

Accurate digitization of analog sensor outputs is critical for reliable water-quality monitoring. While the ESP32's internal ADC offers convenience, its limited resolution, higher noise, and non-linearity can compromise measurement fidelity for low-frequency, high-precision applications. An external ADC with superior performance is therefore warranted.

ADC Performance Criteria Key characteristics for our application include:

- **Resolution:** Sufficient bits to resolve small changes in sensor voltage (e.g., 16-bit).
- **Noise Floor and Linearity:** Low input-referred noise and high linearity across the input range.
- **Programmable Gain:** Adjustable full-scale range to maximize effective resolution.
- **Sampling Rate:** Adequate for quasi-static sensors (no more than a few samples per second).
- **Channel Count:** Enough analog inputs for five liquid sensors plus battery voltage monitoring.
- **Interface and Supply Flexibility:** Support for both 3.3 V and 5 V logic levels and I²C communication.

ADS1115 Candidate The Texas Instruments ADS1115 meets these requirements:

- 16 bit successive-approximation resolution with programmable gain amplifier allowing input ranges ± 256 mV to ± 4.096 V [45].
- Typical input noise < 3 μ V rms and linear characteristic over the full scale [45].
- Four multiplexed channels; additional units can be added on the I²C bus [45].
- Data rates up to 860 SPS, more than sufficient for slow-varying sensors [45].
- I²C interface and supply range 2.0 V to 5.5 V [45].

ESP32 Internal ADC Comparison By contrast, the ESP32's SAR ADC provides only 12-bit resolution and exhibits a non-linearity of up to ± 1 LSB and a typical noise floor around 2 mV rms, making it unsuitable for high-precision analog measurements [46].

Experimental Validation To quantitatively compare the noise performance of the ADS1115 external ADC versus the ESP32's internal ADC, a controlled noise measurement was performed. A precision reference voltage of 1.650 V was applied to both converters, verified to ± 0.001 V using a Fluke 87V digital multimeter.

Procedure:

1. The reference voltage was connected to a single channel of the ADS1115 and to the ADC0 input of the ESP32.
2. For the ADS1115, 295 consecutive readings were recorded at 128 SPS with PGA set to ± 6.144 V [45].
3. For the ESP32 internal ADC, 357 consecutive readings were sampled via the SDK's default ADC driver at maximum resolution (12 bits) [46].
4. All measurements were logged and converted back to voltage using each converter's full-scale range.

Results:

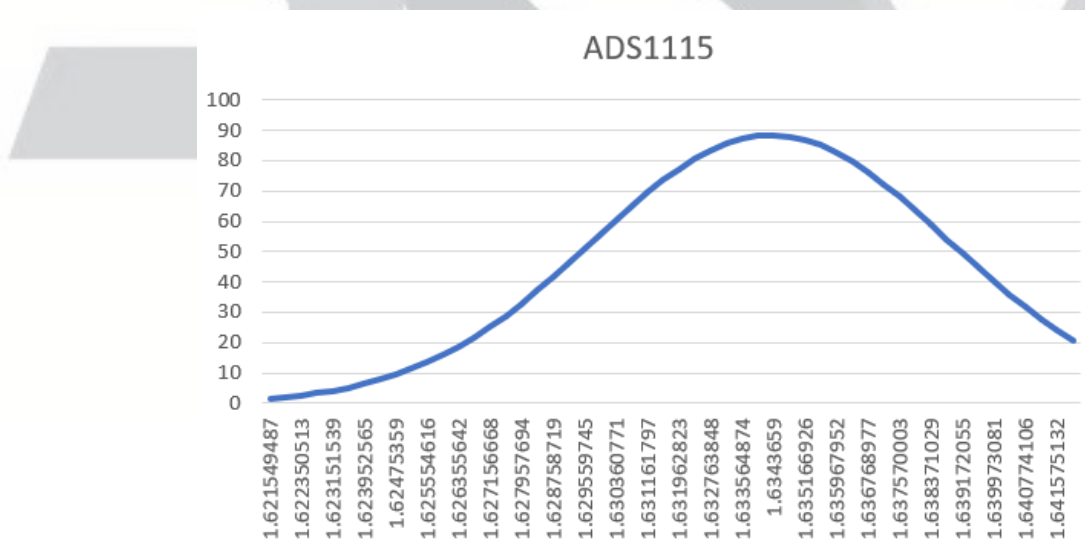


Figure 3.32: ADS1115 Readings

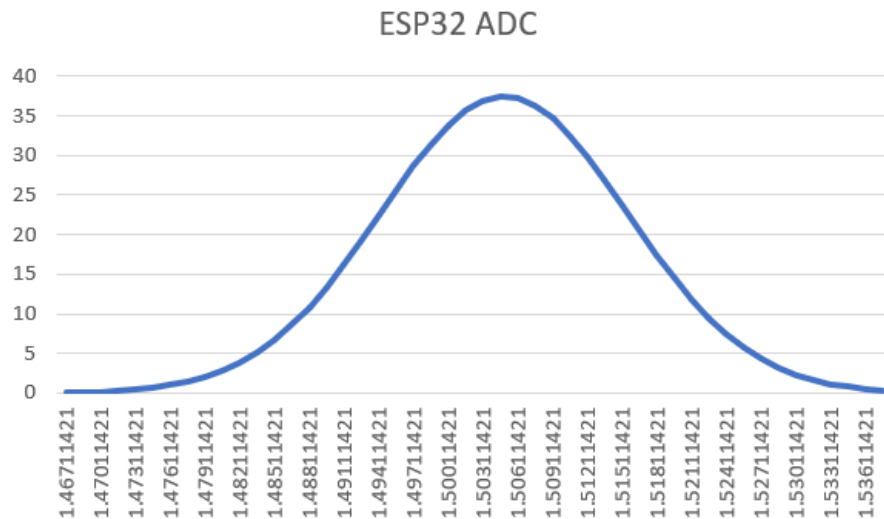


Figure 3.33: ESP32 Internal ADC Readings

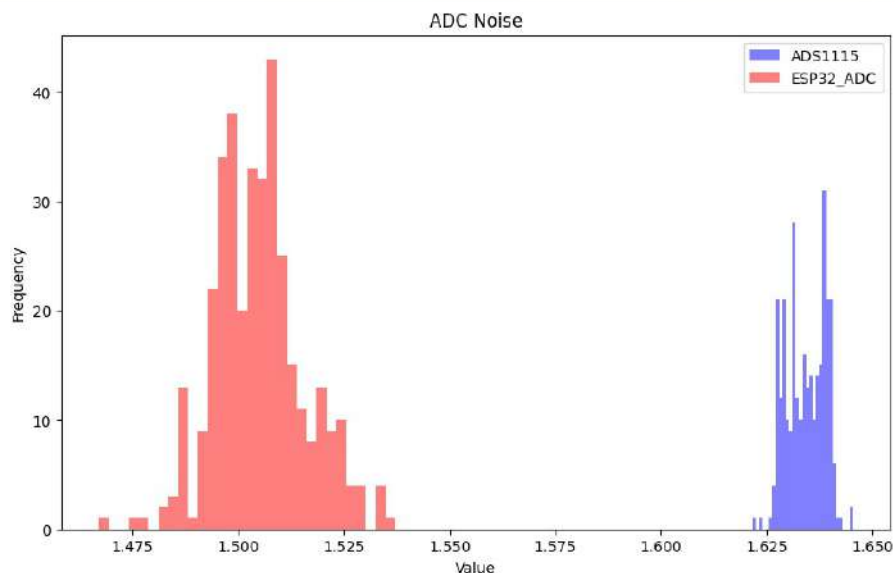


Figure 3.34: Noise Comparison: ADS1115 vs ESP32

ADC	Mean (V)	Std. Dev. (V)	Error (V)
ADS1115	1.634 299	0.004 51	0.015 701
ESP32 Internal	1.504 924	0.010 66	0.145 076

Table 3.4: Comparison of noise and accuracy between ADS1115 and ESP32 internal ADC

Discussion: The ADS1115 exhibits a standard deviation of 4.51 mV, compared to 10.66 mV for the ESP32’s internal ADC—over twice the noise level. The mean reading error (difference from the true 1.650 V) is also an order of magnitude smaller for the ADS1115. These results confirm that the ADS1115 provides substantially lower noise and higher accuracy for precision sensor measurements.

Channel Allocation & Supply Architecture Six analog inputs are required: five liquid sensors plus battery monitoring. Two ADS1115 modules are deployed:

- Module 1 at 3.3 V for the PH Sensor, EC Sensor, and DO Sensor
- Module 2 at 5 V for battery-voltage sensing and ORP Sensor.

This dual-supply approach optimizes dynamic range and simplifies level shifting.

Level Shifting Since the ESP32 operates at 3.3 V logic and the second ADS1115 module is powered from 5 V to maximize its input range, a bidirectional I²C level shifter is employed. This ensures safe voltage translation between the 5 V bus of the ADS1115 and the 3.3 V GPIOs of the ESP32, preserving signal integrity without loading either side.

SHT20 Sensor

A secondary sensing function is provided by the Sensirion SHT20, which monitors internal enclosure temperature and humidity to detect thermal stress or moisture ingress.

Sensor Overview The SHT20 is a fully calibrated humidity and temperature sensor in a compact 3 mm × 3 mm DFN package. It integrates a capacitive humidity transducer, band-gap temperature sensor, amplifier, A/D converter, and digital processing, delivering linearized, 14-bit temperature and 12-bit humidity readings via an I²C interface [47].

Key Specifications

- **Temperature**
 - Range: −40 °C to 125 °C
 - Resolution: 0.01 °C (14-bit)
 - Accuracy: ±0.3 °C typical [47]
- **Humidity**
 - Range: 0 %RH to 100 %RH
 - Resolution: 0.04 %RH (12-bit)
 - Accuracy: ±3 %RH typical [47]
- **Supply**
 - Voltage: 2.1 V to 3.6 V (operating)
 - Current: 0.15 μA (sleep), 200 μA to 330 μA (measuring) [47]
- **Interface:** I²C (up to 100 kHz), digital 2-wire protocol

Application in System Installed on the inner electronics panel, the SHT20 provides real-time enclosure temperature and humidity feedback. Its readings inform fault detection (e.g., water ingress) and thermal management, complementing the primary external water-quality sensors.

ESP32 Microcontroller

The core of the data-acquisition and LoRaWAN stack is the Espressif ESP32 microcontroller. Its key specifications are [46]:

- **Processor:** Dual-core Xtensa® 32-bit LX6 up to 240 MHz
- **Memory:** 520 KiB SRAM, 448 KiB ROM, plus support for up to 16 MB external SPI flash
- **Wireless:** IEEE 802.11 b/g/n Wi-Fi, Bluetooth v4.2 BR/EDR & BLE
- **ADC:** Two 12-bit SAR ADCs, up to 18 channels
- **Peripherals:** SPI, I²C, UART, I²S, PWM, DAC, touch sensors, SD/MMC, Ethernet MAC®
- **Deep-sleep current:** Typical 10 μ A (RTC domain active)

To simplify integration and prototyping, the DFRobot FireBeetle ESP32 development board is employed. The FireBeetle incorporates the ESP32 MCU, a 3.3 V regulator, and standard headers on a compact PCB. [38]. Measured in-circuit deep-sleep current on the FireBeetle board is approximately 50 μ A, enabling multi-day operation on battery power.

SX1276 LoRa Module and Antenna Integration

The system employs the Semtech SX1276 transceiver to implement the physical (PHY) layer of the LoRa communication protocol. LoRa (Long Range) modulation utilizes chirp spread spectrum (CSS) techniques to achieve long-range communication with low power consumption. This modulation provides high sensitivity and robust interference immunity, making it suitable for low-power wide-area network (LPWAN) applications.

SX1276 Transceiver The SX1276 operates over a frequency range of 137 MHz to 1050 MHz, supporting various modulation schemes including LoRa™, FSK, GFSK, MSK, GMSK, and OOK. Key specifications include:

- **Maximum link budget:** 168 dB
- **Output power:** Up to 20 dBm (100 mW)
- **Receiver sensitivity:** Down to 20 dBm

- **Data rate:** Programmable up to 300 kbit s⁻¹
- **Low RX current:** 9.9 mA
- **Sleep current:** 200 nA (register retention)
- **Integrated packet engine:** Supports payloads up to 256 B with CRC

These features enable the SX1276 to provide long-range communication with low power consumption, suitable for battery-powered IoT devices [48].

Antenna Configuration An external antenna is connected to the SX1276 via an SMA connector. The antenna specifications are as follows:

- **Type:** Waveshare LPWA Outdoor Waterproof Antenna
- **Gain:** 5 dBi
- **Impedance:** 50 Ω
- **Frequency range:** 868 MHz to 915 MHz
- **Connector:** SMA male
- **Cable length:** 3 m
- **Antenna length:** Approximately 11 cm
- **Polarization:** Vertical
- **Waterproof rating:** IP65

Although the antenna is optimized for 915 MHz, it is utilized in the system operating within the AS923 frequency band (915 MHz to 928 MHz). This choice is due to limited availability of antennas specifically designed for the AS923 band. The antenna's specifications are suitable for the intended application, providing adequate performance for the system's communication requirements [49].

Integration Considerations The LoRa PHY is implemented using a LilyGO TTGO LoRa32 development board, labeled "LORA32" on its metal RF shield. This module integrates the Semtech SX1276 LoRa transceiver and features an SMA female antenna connector and standard 2.54 mm pin-header spacing facilitating seamless mechanical and electrical integration with the custom PCB.

Communication between the SX1276 and the ESP32 MCU on the FireBeetle is facilitated via an SPI bus (MISO, MOSI, SCK, CS, and IRQ lines), matching the interface requirements specified in the SX1276 datasheet .

An external Waveshare 5 dBi, 50 Ω monopole antenna with SMA male connector is used. The 3 m coaxial cable allows optimal antenna placement outside the enclosure to minimize signal attenuation by ABS walls. A ground-plane clearance area surrounds the SMA connector on the PCB to maintain antenna impedance and radiation efficiency. These layout choices maximize link reliability and range in the AS923 band.

3.4.3 PCB Design and Connector Choices

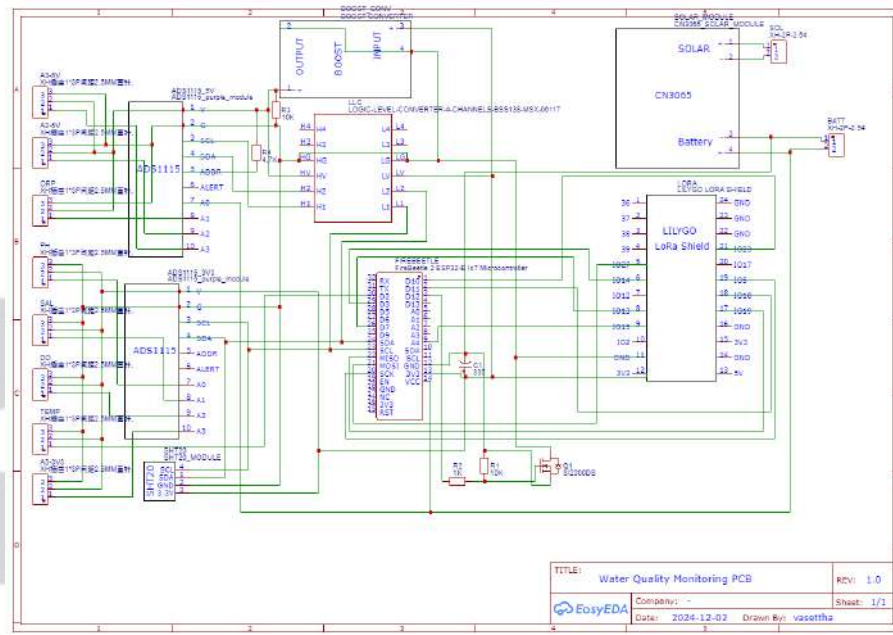


Figure 3.35: PCB Schematic

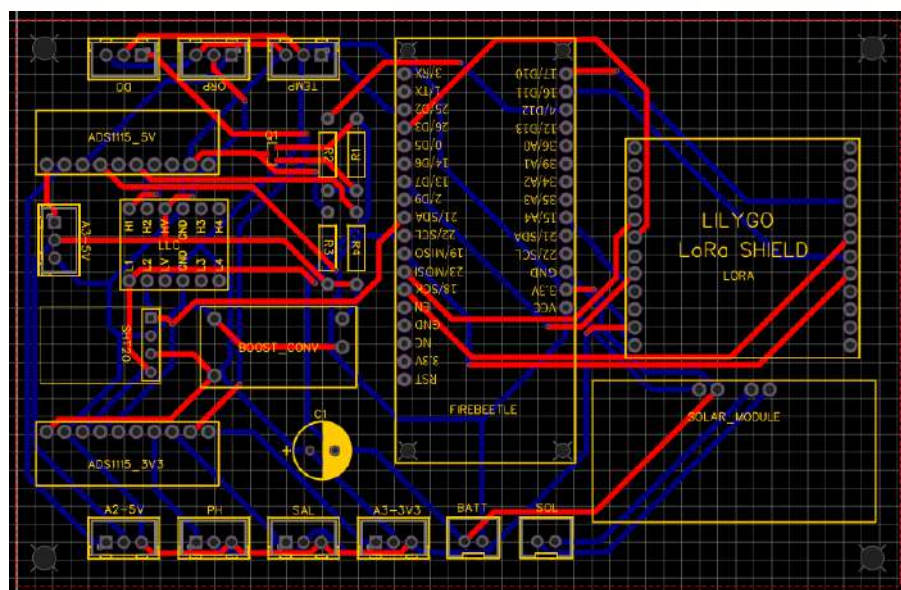


Figure 3.36: PCB Trace

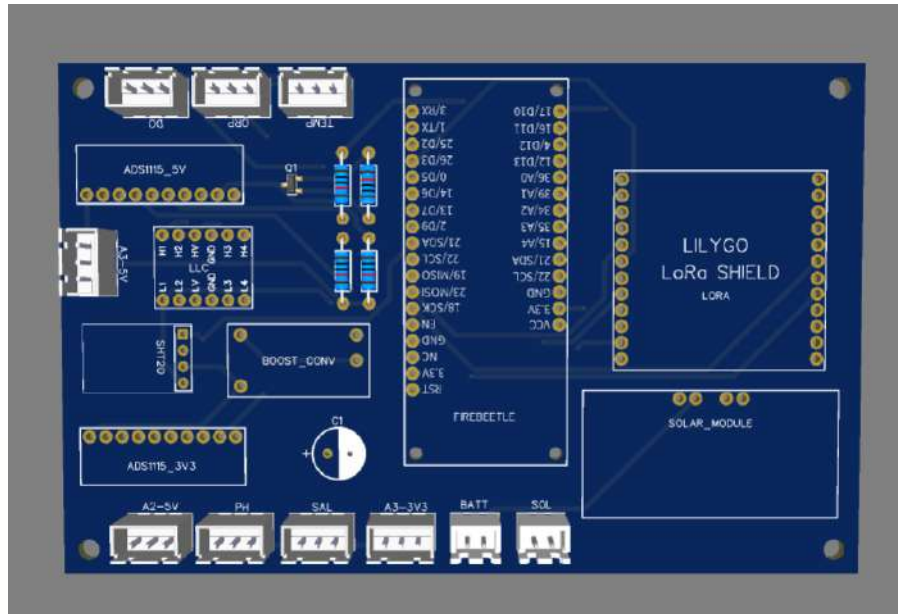


Figure 3.37: PCB 3D Visualization

The custom two-layer PCB measures 125 mm × 80 mm and carries the DFRobot FireBeetle ESP32 development board centrally. The top layer features a solid ground plane, with an isolated keep-out region around the SMA antenna connector to preserve RF performance and minimize coupling to nearby traces. Four M3 mounting holes, one at each corner, are milled for secure attachment within the enclosure.

Connector Types All wire-to-board connections use JST XH housings with 2.54 mm pitch:

- **Antenna:** An SMA female connector for the SX1276 LoRa module's RF output.
- **Power:** Two 2-pin connectors—one for the Li-ion battery, one for the CN3065 charge controller—carrying V_+ and V_- .
- **Sensor buses:**
 - Four 3-pin connectors (V_+ , V_- , signal) to the 3.3 V ADS1115 module for pH, EC, DO, and one spare channel.
 - Three 3-pin connectors to the 5 V ADS1115 module for ORP, battery-voltage calibration, and one spare channel.
 - One 3-pin connector for the DS18B20 temperature sensor (V_+ , V_- , 1-Wire data).

All headers are clearly labeled in silkscreen and positioned along the PCB edge to minimize cable lengths and simplify assembly.

Design Rules To ensure manufacturability and reliable signal integrity, the following fabrication constraints were applied:

- **Trace width:** 0.8 mm
- **Clearance:** 0.2 mm
- **Via diameter:** 0.61 mm (drill 0.305 mm)

These dimensions facilitate standard PCB fabrication processes while providing adequate current-carrying capacity and isolation.

This layout balances RF performance, mechanical robustness, and ease of wiring, resulting in a reliable platform for the low-power IoT sensor node.

3.5 Program and Control Design

3.5.1 Software System Overview

The overall software architecture comprises two tightly-coupled pieces: the on-node firmware and a cloud-hosted application server.

On each sensor node, we run an ESP32-based program that powers the sensors, performs ADC-based readings, applies calibration and temperature-compensation algorithms, then packages the results into a binary payload sent via LoRaWAN (Sections 2.3 and 2.4). These packets are uplinked to The Things Network (TTN), which handles network-layer de-duplication, retries, and metadata (RSSI, timestamp).

In parallel, a Python/Flask “application server” subscribes over MQTT to TTN’s uplink topic `v3/{app-id}@ttn/devices/{dev-id}/up`, receiving every JSON message as soon as it arrives. There it:

1. **Parses** the incoming JSON, extracting `decoded_payload` fields (e.g. `ph`, `salinity`, `waterTemperature`, etc.) and RSSI/UTC timestamp.
2. **Logs** each record into an SQLite database (via SQLAlchemy), storing both the parsed floats and the raw JSON for audit or re-processing.
3. **Hosts** a real-time web dashboard (Flask + Chart.js) and a historical log viewer (jQuery DataTables), allowing on-the-fly trend plots and table lookups for any device.

This cloud component ensures end-to-end traceability—from sensor calibration through network transport to data visualization—and decouples data analysis from embedded constraints.

3.5.2 Cloud Server Application

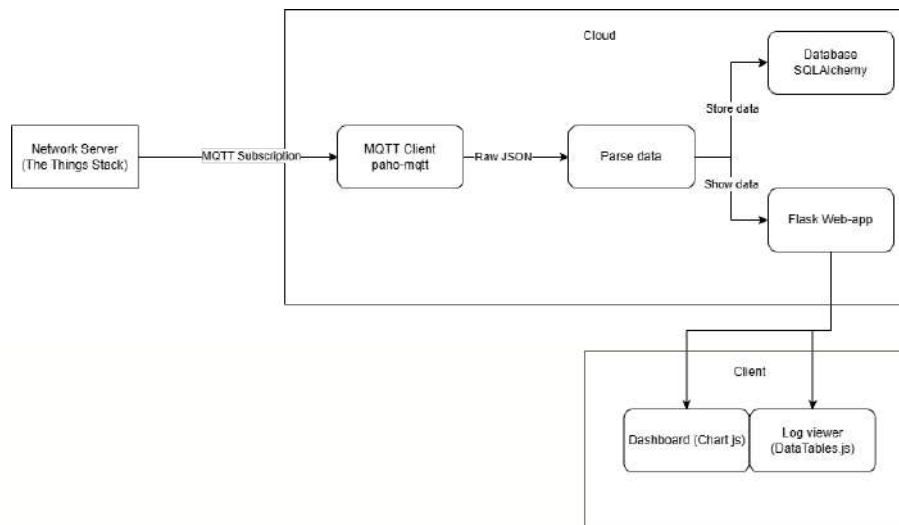


Figure 3.38: Cloud Server Architecture

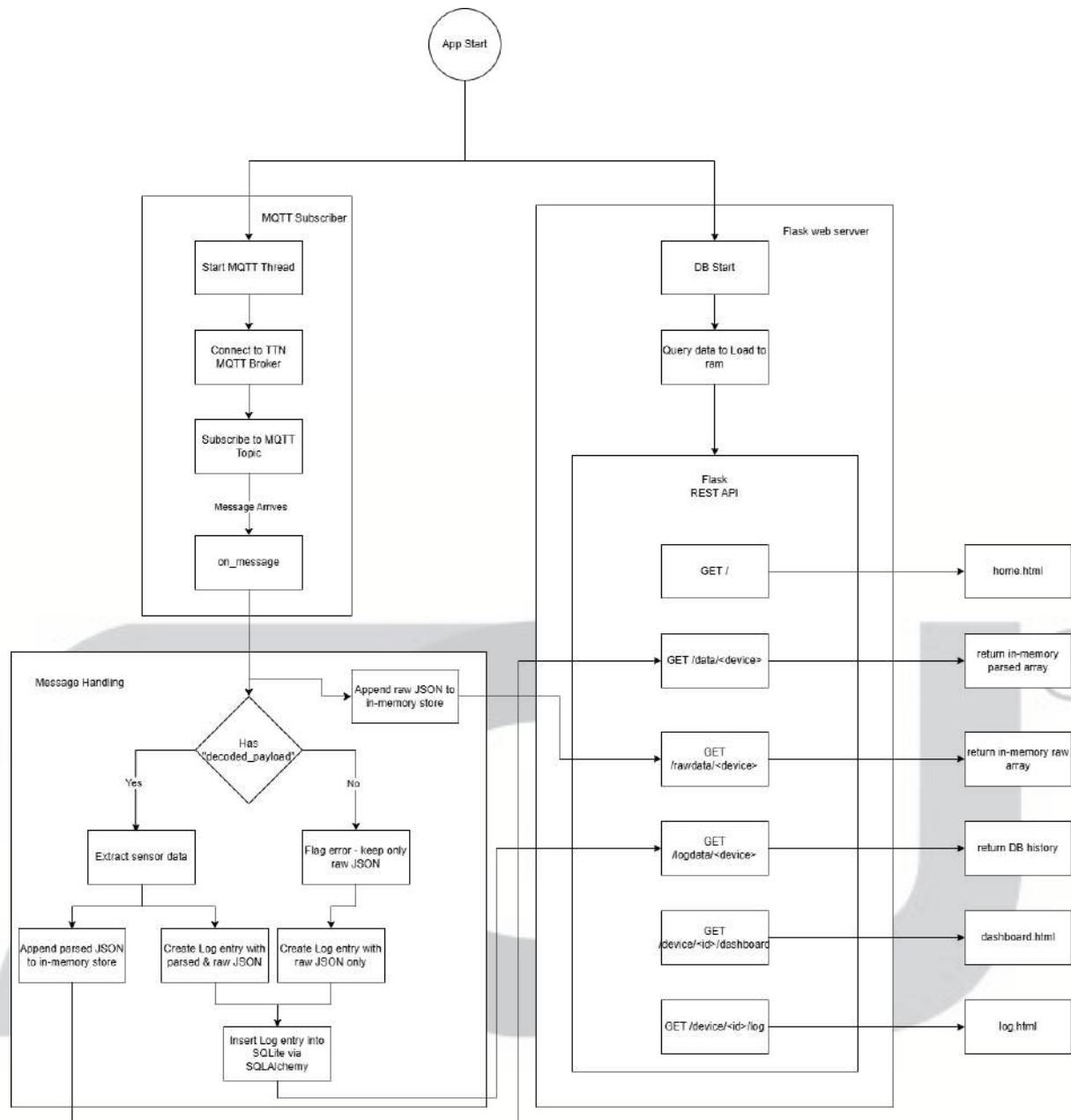


Figure 3.39: Cloud Server Flowchart

The cloud server is implemented in Python using Flask and `paho-mqtt`, with SQLAlchemy for persistence. It runs as two cooperating threads:

MQTT Subscriber

- Connects to TTN's MQTT broker at `au1.cloud.thethings.network:1883`, authenticates via application credentials, and subscribes to the uplink topic `v3/{app-id}@ttn/devices/{device-id}/up`.
- On each `on_message()`, decodes payload JSON, extracts sensor fields (`ph`, `salinity`, `waterTemperature`, `dissolvedOxygen`, `orp`, `humidity`, `batteryVoltage`) and meta-

data (RSSI, received_at), and appends to an in-memory queue for real-time dashboarding.

Data Logger & Web Service

- Persists every message to an SQLite Log table, writing both the parsed floats and raw JSON string.
- Exposes REST endpoints:
 - /data/{device} for live readings (JSON array of last N parsed samples).
 - /logdata/{device} for historical entries (JSON from DB).
- Serves three HTML templates:
 - / – Home page listing available devices.
 - /device/<id>/dashboard – Real-time dashboard with Chart.js for trend lines and status badges.
 - /device/<id>/log – Interactive log table with DataTables.js.

3.5.3 Firmware System

This section provides a high-level overview of the firmware architecture, illustrating both the peripheral interconnections and the control-flow logic implemented on the ESP32.

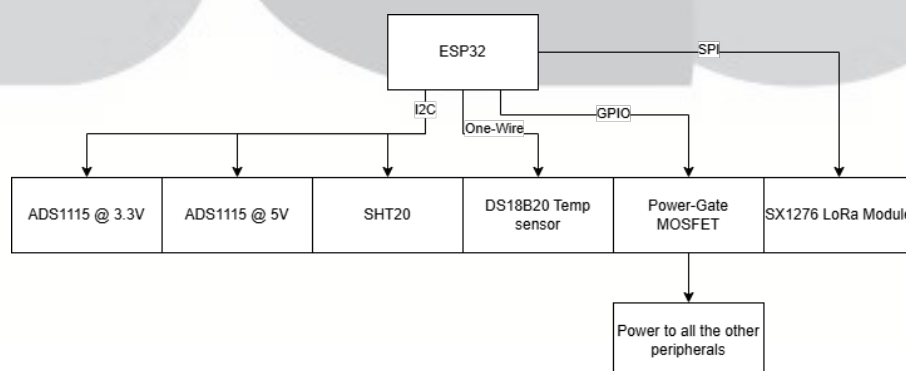


Figure 3.40: Firmware system overview: ESP32 peripheral interfaces and power management.

Figure 3.40 depicts the ESP32 microcontroller at the center of the system and its connections to key peripherals, including the ADS1115 ADCs, SHT20 humidity/temperature sensor, DS18B20 water-temperature probe, power-gate MOSFET, and SX1276 LoRa transceiver.

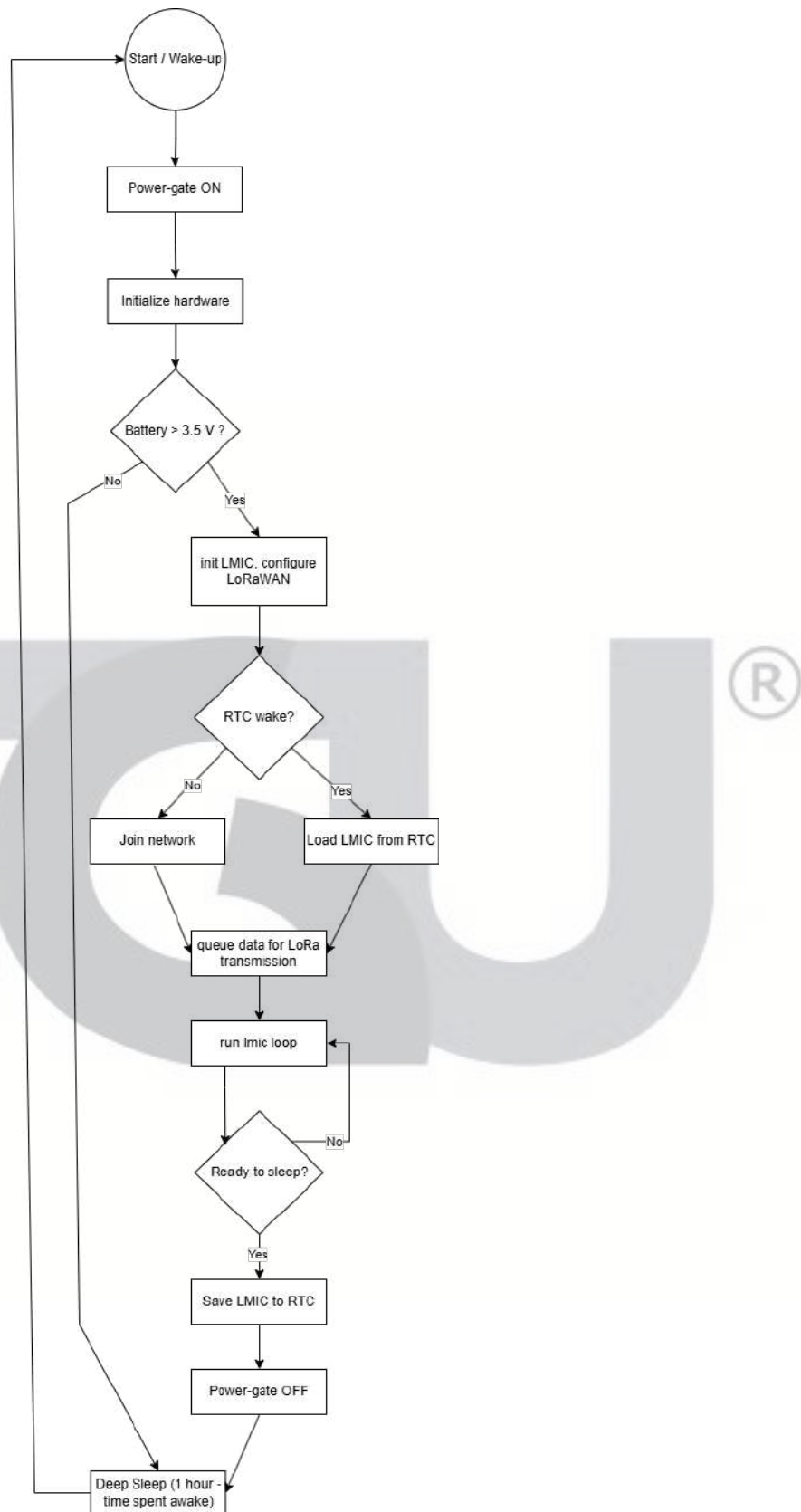


Figure 3.41: Firmware control-flow chart

Figure 3.41 presents the high-level algorithmic flow, detailing the sequence of initialization,

sensor acquisition, LoRaWAN transmission, state preservation in RTC memory, and entry into deep sleep.

Pseudocode Outline:

```
// Called once at startup or RTC wake
setup() {
    initialize_hardware();
    if (read_battery_voltage() < 3.5) {
        go_deep_sleep(TX_INTERVAL);
    }
    os_init();
    LMIC_reset();
    if (woke_from_rtc()) {
        load_LMIC_from_RTC();
    }
    do_send();
}

// Main loop
loop() {
    os_runloop_once(); // Handle LoRaWAN events
    if (no_pending_LMIC_jobs() && ready_to_sleep()) {
        save_LMIC_to_RTC();
        power_gate_off();
        go_deep_sleep(TX_INTERVAL);
    }
    if (timeout_exceeded()) {
        go_deep_sleep(short_retry_interval);
    }
}
```

This architecture cleanly separates:

- **Hardware initialization and power management** (I²C, SPI, GPIO control).
- **LoRaWAN networking** via the MCCILMIC library, including join, uplink, and downlink handling.
- **Deep-sleep scheduling** with RTC context storage for seamless session continuity.
- **Sensor data acquisition** and calibration, abstracted through the `SensorManager` class.

The firmware is organized around four primary objectives:

1. **Low-Level Hardware Control** — I²C sensor reads, GPIO power gating, ADC configuration.
2. **LoRaWAN Stack Integration** — MCCI LMIC library setup, join procedure, uplink/-downlink handling.
3. **Power Management and Deep Sleep** — RTC timer wake-ups, LMIC state preservation in RTC memory.
4. **Sensor Calibration and Compensation** — Offset/gain correction, Single point calibration, two point calibration, temperature compensation.

Hardware drivers

ADS1115 ADC The ADS1115 is a precision, low-power, 16-bit analog-to-digital converter featuring four single-ended (or two differential) inputs and an onboard programmable-gain amplifier (PGA) selectable from ± 6.144 V down to ± 0.256 V via its configuration register . In our firmware, two ADS1115 modules are instantiated:

- `ads3V3_{0x48}` at I²C address `0x48`, powered from 3.3 V, serving the pH, dissolved oxygen, salinity, and ORP sensors.
- `ads5V_{0x49}` at I²C address `0x49`, powered from 5 V, for battery voltage monitoring and ORP calibration.

Low-level interaction proceeds as follows (via the Rob Tillaart ADS1X15 library) :

1. `ads.begin()`: Initializes the I²C interface (`Wire.begin()`) and confirms device presence.
2. `ads.setGain(0)`: Selects a full-scale range of ± 6.144 V (gain=0) to accommodate sensor outputs within our expected 0 V to 5 V range.
3. `ads.readADC(channel)`: Performs a blocking, single-shot conversion on the specified channel:
 - The library writes the 16-bit configuration word (OS=1 to start, MUX selects channel, PGA set by `setGain()`, MODE=single-shot, DR=128 SPS).
 - It polls the OS (or ALERT/RDY) bit until conversion completes.
 - It reads the conversion register via I²C, returning a signed 16-bit raw code.
4. `ads.toVoltage(raw)`: Converts the raw code to volts by scaling against the PGA setting.

[50]

This approach abstracts the low-level register manipulations while ensuring precise, single-shot measurements. The chosen gain and data rate balance noise performance with conversion latency, yielding stable readings for all analog sensors.

SHT20 Temperature & Humidity Sensor The SHT20 is a fully calibrated digital temperature and humidity sensor based on Sensirion's CMOSens® technology, providing linearized measurements over a standard I²C interface at address `0x40` and operating from 2.1 V to 3.6 V. It communicates via two simple commands—`0xF3` to trigger a temperature measurement and `0xF5` for humidity—after which the sensor requires up to 85 ms to complete the conversion and makes three bytes available: two data bytes and a CRC checksum byte .[47]

In our firmware, low-level control of the SHT20 is performed using the Rob Tillaart SHT2x Arduino library. The sequence is as follows:

1. `Wire.begin()` ; initializes the I²C bus at up to 400 kHz.
2. `SHT2x sht` ; constructs the sensor object, using the fixed I²C address (`0x40`).
3. `sht.begin()` ; (optional) configures any custom SDA/SCL pins prior to communication.
4. `sht.readTemperature()` ; writes `0xF3` to the sensor, delays the conversion interval (85 ms), reads the two result bytes and CRC, verifies the CRC-8 checksum (polynomial `0x31`), and applies the linearization formula per the datasheet to return a floating-point temperature in °C .
5. `sht.readHumidity()` ; similarly issues `0xF5`, waits, reads the raw code plus CRC, verifies integrity, and converts to relative humidity (%) via the prescribed transfer function .

[51]

This library abstracts the bit-level register operations, CRC calculations, and conversion formulas, offering a simple, blocking API for accurate environmental sensing.

DS18B20 Water-Temperature Sensor The DS18B20 is a factory-calibrated, 9- to 12-bit temperature sensor communicating over the Maxim 1-Wire bus, which requires only a single data line (DQ) plus ground and an optional pull-up resistor. It can operate in “parasite power” mode—deriving its supply from the data line—or from an external 3.0 V to 5.5 V V_{DD} . [52]

Low-level control is implemented via Paul Stoffregen's OneWire library (v2.3.8) and Miles Burton's DallasTemperature library (v4.0.4) . The sequence is:

1. `OneWire ow(DATA_PIN)` ; Instantiates the bus on the chosen GPIO pin, using open-drain and an external 4.7 k Ω pull-up.

2. `DallasTemperature sensors(&ow)`; Wraps the 1-Wire bus with high-level temperature functions.
3. `sensors.begin()`; Sends a reset/presence sequence and discovers all DS18B20 devices by their unique 64-bit ROM codes.
4. `sensors.setResolution(12)`; Writes to the sensor's configuration register (0x4E) to select 12-bit resolution (0.0625 °C LSB).
5. `sensors.requestTemperatures()`; Issues the "Convert T" command (0x44) to all devices (Skip ROM, 0xCC), waits the required time (up to 750 ms at 9-bit, shorter at higher resolution), then returns.
6. `sensors.getTempCByIndex(0)`; Reads the scratchpad (0xBE command) of the first device: retrieves two bytes for temperature, two bytes for alarm thresholds, one configuration byte, and one CRC byte; verifies CRC-8 (polynomial 0x31) converts the 16-bit signed raw value to °C via

$$T = \frac{\text{MSB} \ll 8 \mid \text{LSB}}{16}.$$

[53], [54]

This architecture leverages the mature `OneWire` and `DallasTemperature` libraries to abstract bus timing, CRC checking, and register-level commands, providing reliable, synchronous temperature readings with minimal firmware complexity.

LoRaWAN Stack Integration

MCCI LoRaWAN LMIC Library The MCCI LoRaWAN LMIC library is a user-friendly Arduino port of the Semtech LMIC (LoraMAC-in-C) stack, providing a comprehensive Class A and Class B LoRaWAN® implementation supporting EU-868, US-915, AU-921, AS-923, and IN-866 regional plans . It abstracts low-level MAC commands into a simple API, handling timing, link-check, ADR, and join management for LoRaWAN® networks. On initialization, the library configures SPI, the SX1276 radio, and region-specific parameters by invoking `os_init()` and `LMIC_reset()` before scheduling the first transmission job.

OTAA Join Process Over-the-Air Activation (OTAA) is the preferred join method for most LoRaWAN® deployments, requiring three root keys: the 8-byte `AppEUI`, the 8-byte `DevEUI`, and the 16-byte `AppKey`. During join, the end device constructs a `JoinRequest` MAC command, encapsulating `DevEUI`, `AppEUI`, and a device-generated `DevNonce`, encrypted with the `AppKey`, and transmits at the network's default data rate. Upon reception, the network server returns a `JoinAccept` message encrypted with the `AppKey`; the device decrypts this payload

to retrieve the session keys (NwkSKey, AppSKey) and channel configuration (CFList) for subsequent uplinks. In our implementation, the keys are stored in `config.h` and supplied via `os_getArtEui()`, `os_getDevEui()`, and `os_getDevKey()` callbacks.

AS-923 Regional Plan Our deployment uses the AS-923 frequency plan, which defines uplink channels in the 920 MHz to 923 MHz unlicensed band with a default maximum EIRP of 16 dBm and duty-cycle or Listen-Before-Talk (LBT) regulations as per Regional Parameters v1.0.2. OTAA JoinReq messages initially use two common channel frequencies (e.g., 923.2 MHz and 923.4 MHz) and receive the full channel set (up to 16 channels) via the CFList field in the JoinAccept. Data rates SF12–SF7 (DR0–DR5) on a 125 kHz bandwidth are supported, with optional DR6 for SF7 on 250 kHz. The LMIC library automatically applies these defaults when `CFG_as923` is enabled in `lmic_project_config.h`.

LMIC Initialization and Uplink Transmission At startup, the LMIC runtime and radio hardware are configured with:

```
os_init();  
LMIC_reset();
```

Here, `os_init()` initializes the LMIC scheduler, SPI interface, and radio driver, while `LMIC_reset()` clears any previous session state, forcing an OTAA join as if the device were new. After reading and serializing sensor data into `payload`, an uplink is requested with:

```
LMIC_setTxData2(1, payload, sizeof(SensorData), 0);
```

This enqueues the packet on application port 1 as an unconfirmed frame. The LMIC state machine automatically selects a channel, enforces duty-cycle limits, and programs the radio for transmission.

Event Processing and Deep Sleep During normal operation, the `loop()` function repeatedly calls:

```
os_runloop_once();
```

which services radio interrupts, MAC timers, and triggers the `onEvent(ev_t ev)` callback for events such as `EV_TXCOMPLETE`. Upon completion of an uplink and no further MAC jobs pending, the firmware invokes:

```
SaveLMICtoRTC(TX_INTERVAL);  
esp_sleep_enable_timer_wakeup(TX_INTERVAL * 1e6 - micros() + CORRECTION_MICROS);  
esp_deep_sleep_start();
```

Jack Gruber [55] demonstrated that, on ESP32, execution restarts at `setup()` after deep sleep, losing all RAM state. To avoid re-joining OTAA on every wakeup—which wastes power and burns join quotas—the complete LMIC context is copied into RTC slow-memory via:

```
RTC_DATA_ATTR lmic_t RTC_LMIC;
void SaveLMICToRTC(int dt) {
    RTC_LMIC = LMIC;
    // adjust band availabilities by deep-sleep duration...
}
```

and restored in `setup()` when waking from a timer source:

```
if (esp_sleep_get_wakeup_cause() == ESP_SLEEP_WAKEUP_TIMER) {
    LoadLMICFromRTC();
}
```

[55]

This preserves session keys, duty-cycle timing, and downlink windows across deep sleep cycles, enabling seamless continuation of LoRaWAN communication without repeated OTAA joins.

Power Management and Deep Sleep

To minimize power consumption between transmissions while preserving the LoRaWAN[®] MAC context, the firmware leverages the ESP32's RTC slow memory. The sequence is as follows:

1. **Prepare for hibernation:** Once the final uplink and downlink windows complete (detected in the `EV_TXCOMPLETE` event), the entire LMIC state—including session parameters, frame counters, duty-cycle timings, and band availability—is copied into RTC slow memory:

```
RTC_LMIC = LMIC;
```

This ensures that, upon wakeup, the device can resume its MAC state without performing a new OTAA join.

2. **Rationale for RTC vs. EEPROM:** Storing context in EEPROM on the ESP32 actually writes to flash memory, which has a limited write-cycle lifespan (typically $\sim 10^4$ – 10^5 cycles) and higher energy cost per write. In contrast, RTC slow memory retains its contents autonomously during deep sleep and supports repeated updates without flash-wear concerns.
3. **RTC memory capacity:** The ESP32 provides 8 kB of RTC slow memory (addressable via `RTC_DATA_ATTR`), which is more than sufficient to store the LMIC context structure (typically 2 kB or less), while leaving room for additional low-power variables.

4. **Power down peripherals:** Prior to sleep, the GPIO driving the power-gate MOSFET is reconfigured to INPUT, disconnecting all non-essential peripherals (ADCs, sensors, radio module, boost converter) and reducing quiescent current to microampere levels.
5. **Configure wakeup source:** A timer wakeup is configured to trigger after the next transmission interval, corrected for the time already spent in the current wake cycle. The expression:

```
esp_sleep_enable_timer_wakeup((TX_INTERVAL * 1000000ULL) - micros());
```

compensates for execution overhead, ensuring the next cycle maintains a consistent period. Here, `micros()` returns the elapsed microseconds since boot.

6. **Enter deep sleep:** The call

```
esp_deep_sleep_start();
```

places the ESP32 into its lowest-power state—shutting off most RAM and digital peripherals—while retaining only the RTC domain and timer logic.

7. **Restore on wakeup:** Upon RTC timer wakeup, execution resumes at `setup()`. The code checks:

```
if (esp_sleep_get_wakeup_cause() == ESP_SLEEP_WAKEUP_TIMER)
```

and, if true, restores the LMIC context:

```
LMIC = RTC_LMIC;
```

This re-establishes the MAC session parameters, frame counters, and duty-cycle state without requiring a new OTAA join.

By leveraging the ESP32's RTC slow memory for context retention and its timer wakeup capability, this approach achieves multi-day autonomous operation with minimal average current draw, avoids unnecessary rejoining of the network, and preserves flash endurance.

Sensor Calibration and Compensation

Effective sensor calibration and compensation are critical to obtaining accurate and reliable measurements in water quality monitoring systems. Calibration aligns a sensor's raw output (usually voltage) with known reference standards, correcting both slope and offset. Two common approaches are:

- **Single-point calibration:** A reference of known value is used to adjust the slope only, assuming negligible offset. This is efficient when zero offset is small and the measurement range is narrow.
- **Two-point calibration:** Two reference points are measured, enabling calculation of both slope and intercept via:

$$\text{slope} = \frac{V_2 - V_1}{E_2 - E_1}, \quad \text{intercept} = E_1 - \text{slope} \cdot V_1,$$

where E_i are expected values and V_i are measured voltages.

Environmental factors—especially temperature—can shift sensor response. Compensation algorithms adjust readings back to calibration conditions, for example using the Nernst equation for electrochemical sensors or empirical temperature coefficients for conductivity-type sensors.

pH Sensor (PHSensor) The pH sensor is calibrated using a two-point method in buffer solutions of pH 6.00 and pH 9.14, bracketing the target measurement range (7.5–8.5). During `PHSensor::calibrate()`, the calibration temperature is recorded, and the raw voltages V_{low}, V_{high} in each buffer are sampled. The driver invokes:

```
calculateTwoPointCalibration(lowBuffer, lowVoltage,
                             highBuffer, highVoltage);
```

storing slope and intercept in EEPROM. On read, the Nernst-based compensation adjusts the slope by the ratio of Kelvin temperatures:

$$m_{comp} = m_{cal} \frac{T + 273.15}{T_{cal} + 273.15},$$

and computes:

$$\text{pH} = V \cdot m_{comp} + b.$$

Electrical Conductivity (SalinitySensor) Salinity (conductivity) is calibrated with a single-point method at 12.88 mS/cm, reflecting the range of interest. In `SalinitySensor::calibrate()`, the expected conductivity E and measured voltage V yield:

$$\text{slope} = \frac{E}{V}, \quad b = 0.$$

To convert from mS/cm to practical salinity (ppt), a factor of 0.5 is applied:

$$\text{salinity (ppt)} = 0.5 \times 12.88 = 6.44 \text{ ppt}.$$

Temperature compensation uses an empirical coefficient ($1.9\%/^{\circ}\text{C}$):

$$\text{comp} = 1 + 0.019 (T - T_{cal}), \quad \text{Salinity} = \frac{\text{raw } E}{\text{comp}}.$$

Dissolved Oxygen (DOSensor) The DO sensor employs single-point calibration at ambient saturation (skipping zero point for efficiency). The driver in `DOSensor::calibrate()` captures the air-equilibrium voltage V_{air} and expected DO E_{air} to compute:

$$m = \frac{E_{air}}{V_{air}}, \quad b = 0.$$

During measurement, raw DO is corrected by the ratio of oxygen saturation solubility functions:

$$DO = (V \cdot m + b) \times \frac{S(T_{cal})}{S(T)},$$

where $S(T)$ is computed via an empirical polynomial.

Oxidation–Reduction Potential (ORPSensor) ORP calibration uses two-point buffers (e.g., 240 mV and 470 mV). In `ORPSensor::calibrate()`, measured voltages V_{low} , V_{high} and expected ORP values yield slope and intercept via the two-point formula. Temperature compensation is minimal for ORP probes, so reads apply:

$$ORP = V \cdot m + b.$$

3.6 Testing and Validation Protocols

To ensure that the developed system meets the objectives outlined in this thesis, a comprehensive validation strategy was adopted. This section presents the methodologies used to evaluate the system across the following key performance indicators (KPIs):

1. **Sensor Data Acquisition:** Accurate and periodic measurement of water quality parameters.
2. **Communication Performance:** Reliable data transmission over LoRa with an effective range of up to 1 km.
3. **Energy Autonomy:** Full operational sustainability via solar power without external intervention.
4. **Environmental Robustness:** Physical durability and operational stability under realistic shrimp farming conditions.

Each of the four primary objectives is addressed through specific tests and data collection protocols:

3.6.1 Sensor Data Acquisition Validation

To verify that the system reliably measures and logs water quality parameters—pH, dissolved oxygen (DO), oxidation-reduction potential (ORP), salinity, and temperature—long-duration tests are performed. The system is left operational for a continuous period of one week, with a cloud server listening for LoRaWAN transmission used to record sensor readings at regular intervals. Consistency and plausibility of the data over time are evaluated to ensure sensor stability and firmware reliability.

3.6.2 LoRa Communication Range Test

To confirm that the system meets the minimum transmission range requirement of 1 km, a line-of-sight range test is performed between the deployed sensor node and a fixed LoRaWAN gateway. At various distances from the gateway, signal parameters such as Received Signal Strength Indicator (RSSI) and Signal-to-Noise Ratio (SNR) are recorded from uplink messages. Packet delivery rate and radio link quality are used to evaluate communication reliability, in line with expected LoRaWAN performance under the AS923 frequency plan [4], [6].

3.6.3 Power Sustainability Evaluation

The node is designed to operate indefinitely using only solar power. To validate this, three complementary tests are conducted:

1. **Power Consumption Profiling:** Current consumption in both active and deep sleep modes is measured using a digital multimeter. The average energy usage per cycle is then estimated by integrating the current profile over a full operational cycle.
2. **Battery Capacity Characterization:** The lithium-ion battery is charged and discharged using a constant current load to empirically determine its usable capacity.
3. **One-Week Energy Autonomy Test:** The complete system is deployed in an outdoor setting with solar exposure and monitored for one week. Battery voltage is logged at regular intervals to verify that energy harvested by the solar panel is sufficient to replenish energy consumed, ensuring net-zero energy balance.

3.6.4 Field Durability and Deployment Robustness

To assess mechanical and environmental robustness, the sensor node is deployed in a realistic shrimp farming environment. Exposure to water spray, high humidity, direct sunlight, and

potential mechanical disturbances are observed over several days. Any structural deformation, ingress, or system instability is documented. Additionally, operational logs are reviewed to ensure continued sensor measurements and data transmission throughout the deployment period.

