

Development of an ESP32-Based LoRa Telemetry System for Real-Time Environmental Monitoring in Seaweed Cultivation

Alia Amini Azhar¹, Sakinah Putri Azria Azis², Mudarris^{3*}

^{1,2}Computer Engineering Study Program, Department of Informatics and Computer Engineering,
Faculty of Engineering, Universitas Negeri Makassar, Makassar, Indonesia

³Electrical Engineering Program, Department of Electrical Engineering,
Faculty of Engineering, Universitas Negeri Makassar, Makassar, Indonesia

*Corresponding author : mudarris@unm.ac.id

ABSTRACT

Remote environmental observation requires a reliable path from sensor acquisition to an accessible monitoring interface. This article presents an ESP32 telemetry prototype developed as a foundation for future seaweed cultivation monitoring. The system integrates a DHT22 sensor, a light-dependent resistor module, and a capacitive medium-moisture sensor with a transmitter, receiver, Wi-Fi connection, and Firebase monitoring platform. The original project specifies LoRa in its design but identifies nRF24L01 in its implementation discussion; consequently, the available measurements are treated as prototype telemetry results rather than verified LoRa performance. Reported comparison readings were 30.60 °C against 30.20 °C and 85.40% relative humidity against 85.00%, corresponding to absolute differences of 0.40 °C and 0.40 percentage points. Five transmission records produced a calculated mean latency of 195 ms, whereas the report separately states an average dashboard delay of 2.037 s and upload synchronization of 95.45%. These figures describe different observation boundaries and cannot be directly combined. The prototype demonstrates a documented multisensor monitoring workflow at laboratory scale, but neither marine water quality nor long-range communication has been established. Additional analysis identifies requirements for sensor calibration, packet traceability, timing verification, and field validation before deployment in seaweed cultivation.

Keywords: ESP32; environmental monitoring; telemetry; seaweed cultivation; sensor validation

1. INTRODUCTION

Seaweed cultivation provides the application context for this project, while the engineering task is to acquire environmental observations and deliver them to a remote interface. A functioning monitoring chain can reduce dependence on reading instruments directly at the sensor location. However, the value of that chain depends on what is measured, how readings are transmitted, and whether the displayed information can be traced to the original observation.

Monitoring is most useful when the measurement can be connected to a practical observation task. A remote display can show whether a node continues reporting, whether a value is changing, and when the latest observation was received. These functions support inspection of the prototype even before its sensors are suitable for marine deployment. They also expose failures that a single demonstration reading can conceal: stale values, incomplete uploads, and disagreement between a displayed unit and the quantity actually measured.

The engineering challenge is therefore broader than connecting several components. Sensor selection, data representation, transmission, and presentation must form a consistent chain. A well-designed interface should preserve units, timestamps, and channel identity so that the user can distinguish an environmental change from a communication or conversion problem. This project provides an opportunity to examine that chain using the documented prototype and to identify the additional evidence required for its intended seaweed-cultivation application.

1.1 Monitoring requirements and measurement relevance

The project report describes a two-node arrangement intended for environmental monitoring around seaweed cultivation. This is a useful starting point for learning distributed acquisition, radio transmission, and cloud visualization. Nevertheless, environmental context and biological measurement must be distinguished. Air temperature and relative humidity describe conditions around a DHT22 sensor; they do not establish the temperature or chemistry of the cultivation water. Similarly, an LDR reading represents an electrical response to incident light, while a capacitive probe responds to its surrounding medium. Neither output becomes a calibrated aquatic parameter simply because it appears on a dashboard.

The present article therefore frames the work as a prototype toward seaweed monitoring. It retains the reported measurements while narrowing conclusions to what those measurements can support. This interpretation avoids presenting a laboratory demonstration as a validated offshore installation. The available report does not document seaweed species, immersion depth, seawater reference measurements, cultivation outcomes, or a prolonged coastal trial. These omissions define the scope of the evidence rather than invalidate the educational prototype.

1.2 Communication architecture and the evidence gap

A remote sensor node and an internet gateway solve different connectivity problems. The local radio link transports observations between the nodes; the gateway then forwards received observations to the monitoring service. Successful reception does not guarantee that a cloud write succeeds, and a successful cloud write does not establish that the measurement itself is accurate. These stages should be evaluated separately with a common packet identifier.

The source introduces LoRa but later attributes behavior and firmware adjustments to nRF24L01. Nordic identifies the nRF24 family as 2.4 GHz transceivers [1], while Semtech describes LoRa and LoRaWAN as distinct physical-layer and networking concepts [2]. These names are not interchangeable. Because the original firmware and an unambiguous bill of materials are unavailable, this article does not resolve the discrepancy by assumption. Radio-specific claims remain conditional on hardware verification.

1.3 Objective and contribution

The objective is to organize the reported prototype into a reproducible engineering account and examine the limits of its validation. The contribution comprises a clear acquisition-to-dashboard architecture, faithful presentation of the numerical observations, separation of sensor differences from calibration accuracy, and separation of local latency from dashboard responsiveness. The extended discussion proposes concrete tests needed to move from a demonstration to an application-ready system. These proposed tests are future work and are not represented as completed experiments.

1.4 Related research and implications for this prototype

Augustin et al. [3] examine LoRa physical- and data-link-layer behavior through field testing and simulation. Their work provides a basis for treating communication performance as an experimental question rather than inferring it from the technology name. For the present prototype, this means that radio configuration, packet structure, and test conditions should accompany any reported delivery or latency value. The available ten-metre demonstration cannot establish the coverage of a future LoRa deployment, particularly while the source identifies two different radio technologies.

Georgantas et al. [4] present a low-cost water-quality monitoring system based on a LoRa network, incorporating water-temperature, pH, and total-dissolved-solids sensing. This illustrates an important design distinction: a water-monitoring application requires channels that directly address the water variables of interest. The present DHT22, LDR, and medium-moisture channels support acquisition and telemetry learning, but their outputs cannot replace those aquatic measurements. The comparison concerns sensor selection and system architecture, not a numerical ranking of the two systems.

Soon et al. [5] integrate LoRaWAN sensing and an LSTM autoencoder in outdoor microalgae culture. Their setup includes pH, water temperature, electrical conductivity, and oxidation–reduction potential, connecting the observations to a defined cultivation experiment. Microalgae cultivation is not equivalent to seaweed farming, so its biological relationships and model performance should not be transferred to this project. Its relevance is methodological: meaningful biological interpretation

requires a defined culture, appropriate water measurements, and observations spanning the behavior being investigated.

1.5 Research gap and scope of comparison

Together, these studies identify three connected requirements: documented communication conditions, application-relevant measurement channels, and validation within a defined environment [3]–[5]. The present work addresses an earlier stage in that sequence. It organizes the available prototype evidence and identifies what must be documented before the system can support a cultivation claim. This contribution is practical and descriptive; it does not propose a new radio protocol or claim to outperform established water-monitoring systems.

Cross-study latency comparisons would be misleading without common timing boundaries. A packet-level delay, gateway response, and complete dashboard update may include different operations. Likewise, a laboratory sensor check cannot be compared directly with a field instrument evaluated over a wider range. Consequently, the discussion uses prior research to establish design and validation requirements while retaining the project's reported numbers within their original scope. Future benchmarking should specify a shared payload, sampling interval, observation duration, and definition of successful delivery.

A further gap concerns the relationship between raw observations and visual indicators. A gauge may appear intuitive while obscuring how an electrical reading became a percentage or another physical unit. In this work, the dashboard and the tabulated readings are treated as different forms of evidence: the image documents interface design, while the tables document specific reported comparisons. Keeping these roles separate prevents graphical polish from being mistaken for measurement validation.

The expanded introduction motivates a staged development strategy. First establish that each channel measures a clearly defined quantity; then verify delivery and storage of that observation; finally evaluate whether the information supports a cultivation decision. This order keeps later analytics dependent on demonstrably usable inputs. It also provides a practical basis for deciding which improvements require new instruments, which require firmware changes, and which require a longer field experiment.

2. RESEARCH METHODOLOGY

2.1 Prototype development and source of evidence

The original project follows a prototyping approach: problem identification, literature review, hardware and software design, assembly, functional testing, correction, and reporting. This article reorganizes those activities into a research narrative using the supplied report as the experimental record. No new hardware experiment was performed during manuscript preparation. Recalculated quantities are explicitly identified, and proposed validation procedures are kept separate from observed results.

The available evidence consists of component descriptions, prototype photographs, a serial-monitor image, four sensor comparison rows, five transmission timestamp pairs, and summary performance statements. Full firmware, radio configuration exports, calibration certificates, and a complete timestamped dataset are not provided. Accordingly, calculations are descriptive and restricted to the listed observations. No statistical population claim or causal attribution is inferred from the small sample.

2.2 Hardware and measurement channels



Figure 1. DHT22, LDR, and capacitive moisture probe, left to right.

Table 1. Components and their roles in the reported prototype

Component	Documented role	Interpretation
ESP32 nodes	Acquisition and receiving gateway	Processes sensor and network data
DHT22	Air temperature and humidity	Ambient measurements
LDR module	Analog light response	Raw ADC counts
Capacitive probe	Medium-moisture response	Raw ADC counts
External radio	Wireless node-to-node transfer	LoRa/nRF24L01 identity unresolved
Wi-Fi and Firebase	Forwarding and remote display	Separate from local radio delivery
Arduino IDE	Firmware development	Program versions not supplied

The DHT22 is connected through a digital interface, whereas the LDR and capacitive probe connect to analog inputs. Espressif distinguishes raw analogRead output from calibrated millivolt conversion and documents a default 12-bit return range of 0–4095 [6]. Even calibrated voltage would require a separate sensor transfer function before becoming illuminance or moisture content. This manuscript retains the original analog values as counts.

2.3 Functional sequence

At the transmitter, the firmware periodically obtains the sensor values and assembles a payload. The radio sends that payload to the receiver, which decodes the values and forwards them over Wi-Fi to the monitoring platform. The source discusses sequential acquisition, so simultaneous sampling is not assumed. For reproducibility, a future implementation should record the sample time, packet sequence, channel validity, receiver time, and upload outcome. A shared identifier distinguishes retries from new observations.

2.4 Evaluation definitions and calculations

For a reported comparison pair, the absolute difference is defined as $\Delta = |x - r|$, where x is the system reading and r is the reference value. The relative difference is calculated as $e = 100|x - r|/|r|$ when the denominator is nonzero. These calculations describe the supplied pairs; they do not establish a calibration curve or total measurement uncertainty. Temperature differences are primarily interpreted in degrees Celsius because a percentage based on the Celsius zero point is not a general accuracy specification.

For packet i , the timestamp difference is $L_i = t_{RX,i} - t_{TX,i}$. The sample mean is the sum of the listed delays divided by their count. This calculation assumes that the reported transmit and receive timestamps are comparable. Since clock synchronization and timestamp placement are undocumented, the values are treated as reported latency observations. Packet delivery ratio should count unique received packets divided by unique transmitted packets within the same defined test interval.

Dashboard latency has a broader boundary than local radio timing. It may include sampling, firmware waiting, packet transfer, gateway processing, internet upload, and interface refresh. The report gives a separate end-to-end summary, but does not provide matched logs that connect it to the five local timestamp pairs. These datasets are therefore analyzed separately. The same restriction applies to the reported cloud synchronization percentage, whose denominator and observation period are absent.

3. RESULTS AND DISCUSSION

3.1 Physical implementation

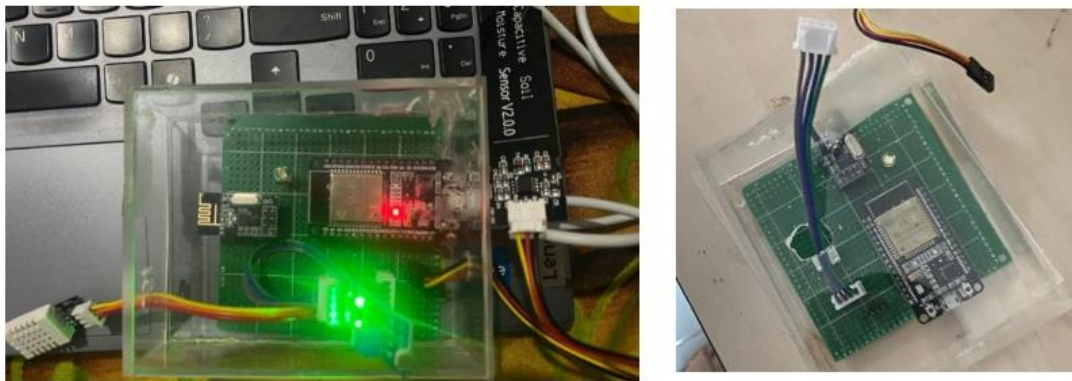


Figure 2. Prototype hardware photographs reproduced from the project report.

Figure 2 documents the assembled prototype and its associated sensor connections. The source describes separate transmitting and receiving sides, with the receiver acting as the internet gateway. The photographs support the presence of a physical assembly but do not prove enclosure sealing, corrosion resistance, or long-term marine suitability. No environmental protection rating or durability test is reported. Those properties should be tested before describing the enclosure as suitable for seawater exposure.

The implementation provides a practical platform for checking whether a sensor change can propagate to a receiving terminal. Its research value depends on preserving the evidence chain: hardware identity, firmware configuration, original observations, and display output should describe the same test. Resolving the radio discrepancy is the first requirement for a repeatable implementation report.

3.2 Dashboard implementation and interpretation

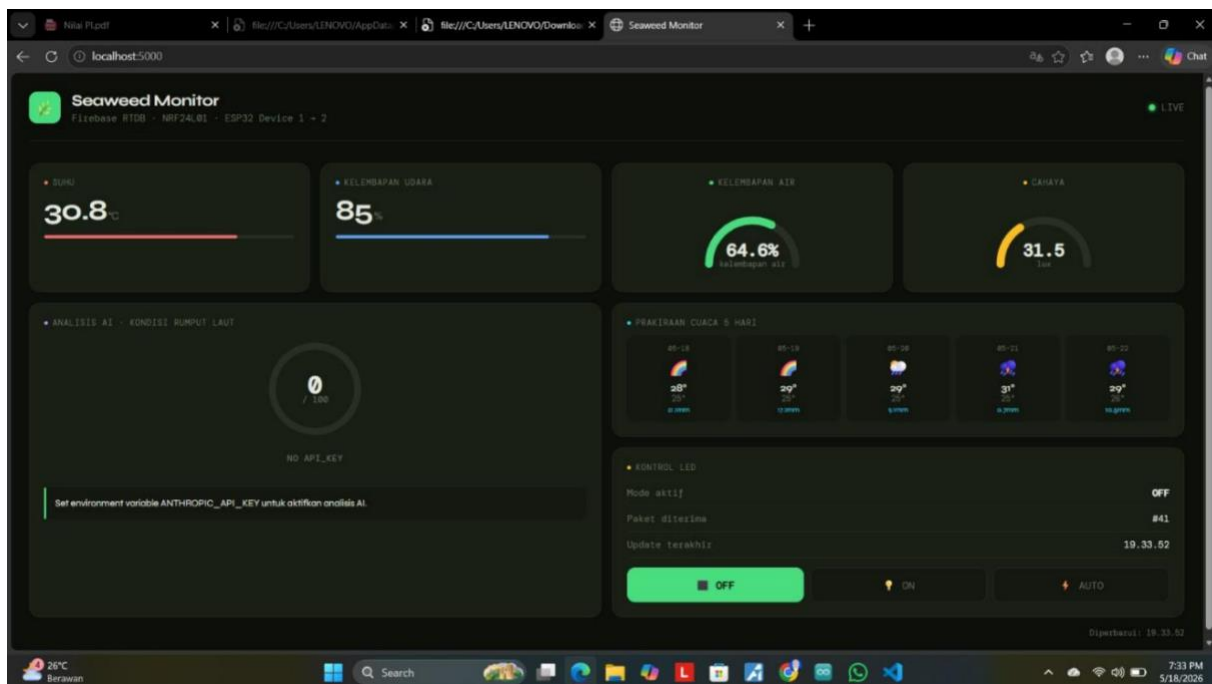


Figure 3. Original Seaweed Monitor dashboard screenshot reproduced from the project report. Original interface labels are retained.

Figure 3 shows the browser-based Seaweed Monitor interface. Its banner identifies Firebase RTDB, NRF24L01, and an ESP32 device path. This provides additional documentary evidence that

nRF24L01 was associated with the displayed prototype, although a screenshot does not establish the complete hardware configuration of every reported test. The image is preserved as supplied so that its visible labels and status indicators remain inspectable.

The top row contains air-temperature and humidity displays, a moisture gauge, and a light indicator. The Indonesian labels “Suhu,” “Kelembaban udara,” “Kelembaban air,” and “Cahaya” correspond to temperature, air humidity, moisture, and light. The snapshot shows 30.8 °C, 85%, a moisture indication of 64.6%, and a light indication of 31.5 lux. These are displayed snapshot values, not additional calibrated experimental observations. The report does not document the conversion that maps the analog channel counts to the latter two indicators.

The lower interface contains an AI-analysis area, a five-day weather panel, and LED controls. The AI area displays “NO API_KEY”; therefore, this image does not demonstrate successful AI analysis. Likewise, the weather tiles and OFF, ON, and AUTO buttons establish that these interface elements exist, but do not validate forecast accuracy or automatic control. The supplied project contains no documented actuator test establishing their effectiveness.

A received-packet count and last-update time provide useful context for interpreting freshness. However, the displayed LIVE status alone cannot establish uninterrupted delivery or a successful database write for every sample. Future interface validation should compare each displayed record with the transmitter log, receiver log, and confirmed database record. Such a check would reveal whether the dashboard shows current observations, previously cached values, or converted quantities whose calibration still needs to be documented.

3.3 Sensor comparison results

Table 2. Reported sensor readings and recalculated pairwise differences

Parameter	Reference	System	Absolute difference	Relative difference
Air temperature	30.20 °C	30.60 °C	0.40 °C	1.32%
Relative humidity	85.00% RH	85.40% RH	0.40 pp	0.47%
LDR output	2600 counts	2605 counts	5 counts	0.19%
Medium probe	3270 counts	3278 counts	8 counts	0.24%

Table 2 preserves the source values and recalculates their differences using the stated formula. The humidity difference is expressed in percentage points (pp), avoiding confusion with the relative percentage difference. The analog rows are comparisons in ADC space. Without a traceable reference instrument and transfer function, a small count difference does not demonstrate accuracy in lux, water content, or another environmental unit.

The comparison includes only one listed reference-system pair for each channel. It cannot establish repeatability across time, response over a measurement range, or performance under changing humidity and supply conditions. The report refers to counter values 77–82, but does not provide a complete numerical sensor series for those records. Additional points have therefore not been reconstructed or invented. The original status label of valid is interpreted as the project’s functional assessment rather than a metrological certification.

3.4 Calibration and signal conditioning

A stronger temperature and humidity evaluation would expose the sensor and a documented reference instrument to several stable conditions and retain repeated paired readings. Their locations and stabilization time should be recorded, because spatial differences can otherwise appear as sensor error. Mean bias, dispersion, and maximum absolute difference could then be reported for the observed range. Sensor replacement should trigger a new check instead of inheriting calibration from another unit.

For the analog channels, validation should begin with the electrical chain. Record the configured ADC resolution, attenuation, supply voltage, and wiring arrangement. Then compare raw readings with a known input or independently measured quantity. Light calibration requires controlled geometry

and a suitable reference; medium-moisture calibration requires a defined material and a defensible reference procedure. A count-to-percentage conversion based only on assumed endpoints would not resolve these requirements.

The report states that a moving average was introduced to reduce analog fluctuations, but does not supply its window length or before-and-after records. The effect cannot be quantified. A future comparison should preserve both raw and filtered streams and use the same input changes. Smoothing should be evaluated against its added response delay, since a stable-looking curve can conceal a short disturbance. Filter parameters must accompany any resulting claim of improved signal quality.

3.5 Local transmission observations

Table 3. Transmission timestamps and delays reproduced from the report

Packet	Transmit time	Receive time	Delay	Outcome
1	19:35:10.150	19:35:10.330	180 ms	Success
2	19:35:20.150	19:35:20.355	205 ms	Success
3	19:35:30.150	19:35:30.342	192 ms	Success
4	19:35:40.150	19:35:40.360	210 ms	Success
5	19:35:50.150	19:35:50.338	188 ms	Success

The five delays total 975 ms, yielding a calculated mean of 195 ms and a range of 180–210 ms. The transmit timestamps are spaced ten seconds apart. That interval describes these listed records and should not be treated as the duration of radio transmission or assumed to govern every experiment in the report. All five rows show successful reception, which is evidence of successful delivery for this small listed sample.

The report separately states 100% local packet delivery at an open-space distance of 10 m, referencing counters 77–82. This six-counter excerpt and the five-row timing table are not shown to be the same test. The article therefore preserves both descriptions without merging their denominators. The absence of a missing counter in a short excerpt does not establish sustained packet delivery over long periods or at distances beyond the tested setting.

3.6 Dashboard responsiveness and upload completion

The source reports a mean dashboard delay of 2.037 s and synchronization of 95.45%. These are retained as reported summary values because the raw observations needed to reproduce them are missing. The dashboard figure is approximately an order of magnitude larger than the listed local delays, but the difference cannot be allocated to a particular processing stage without matched timestamps. In particular, subtracting the two means would not establish the internet-upload delay.

The discussion attributes a roughly two-second wait to radio buffering and timeout settings. Such a delay should be investigated as implementation behavior rather than an unavoidable property of the radio technology. Instrumentation should identify when acquisition completes, when the radio send operation begins and ends, when the receiver parses the payload, and when the cloud confirms storage. These boundaries would distinguish firmware scheduling from retransmission and network waiting.

A displayed value should also include its age and connection state. Firebase documents connectivity handling and synchronization behavior for web clients [7], but a local interface update alone is insufficient evidence of a completed remote write. For this prototype, upload acknowledgements and sequence identifiers should be stored alongside observations. Missing or delayed records should remain distinguishable from periods when the environment was simply unchanged.

3.7 Radio identity and reproducibility

The original design and conclusion name LoRa, whereas the detailed discussion names nRF24L01, a 2.4 GHz operating band, and a 250 kbps configuration. The documented 250 kbps capability applies to nRF24L01+ [1]; the source does not establish the exact module revision. This is a second reason to verify the physical part and firmware rather than silently replace one radio name with another. No long-range LoRa outcome is claimed in this article.

A repeatable test record should include module markings, antenna type, supply arrangement, radio library version, channel, payload size, acknowledgement policy, and retry settings. If the system is rebuilt with LoRa, its operating parameters and results should be documented as a distinct experiment. Comparison between the alternatives would require matched packet content and a defined deployment scenario. The current observations do not rank one technology above the other.

3.8 Limits of application to seaweed cultivation

The monitored channels do not directly measure salinity, pH, dissolved oxygen, or submerged temperature. Consequently, the observations cannot establish seawater suitability or explain seaweed growth. An application study should begin by identifying the target species, site, and management question, then select instruments that measure the relevant quantities. Biological thresholds should come from an appropriate source or a validated local protocol, not from the laboratory sensor values in this report.

Coastal deployment would also expose the system to conditions absent from the documented test. A proposed validation should address enclosure integrity, connector protection, mounting stability, sensor fouling, power availability, and maintenance access. Field trials should record distance and antenna placement alongside communication outcomes. These are design and test recommendations; no durability, energy autonomy, or offshore range result is available from the existing project.

3.9 Practical validation plan

Table 4. Proposed next-stage validation and the evidence it should produce

Priority	Test	Required evidence
1	Verify hardware and firmware	Radio model, configuration, reproducible build
2	Repeat sensor comparison	Paired readings over a defined range
3	Trace each packet	Unique IDs, stage timestamps, acknowledgements
4	Extend communication trials	Distance, duration, packet totals, retry counts
5	Test a relevant aquatic setup	Suitable probes and reference observations
6	Evaluate sustained operation	Power logs, faults, maintenance and data gaps

This sequence prioritizes traceability before adding features. Automatic control and predictive alerts should follow measurement and communication validation, since an inaccurate or stale input can trigger an inappropriate response. The present contribution remains monitoring: no actuator experiment, controlled comparison with manual inspection, or measured improvement in cultivation yield is reported.

4. CONCLUSION AND RECOMMENDATIONS

The project documents an ESP32 multisensor telemetry prototype linking acquisition, wireless reception, and a monitoring platform. The reported sensor pairs differ by 0.40 °C for air temperature, 0.40 percentage points for relative humidity, five ADC counts for the LDR, and eight ADC counts for the medium probe. Five local transmission records yield a calculated mean delay of 195 ms. The separate 2.037 s dashboard delay and 95.45% upload synchronization remain source-reported summaries rather than independently reproduced measurements.

These observations support a limited functional demonstration. They do not establish calibrated water-quality monitoring, sustained long-range performance, or improvements in seaweed

production. The inconsistent radio descriptions prevent attribution of the available results to verified LoRa hardware. Resolving that identity, documenting sensor references, and preserving complete packet logs are the immediate priorities. Further development should introduce application-relevant aquatic probes and evaluate the complete system under a defined field protocol.

5. ACKNOWLEDGEMENTS

This article is based on the Group 3 final project report for the Telemetry Systems course in the Computer Engineering Study Program, Universitas Negeri Makassar. The supplied report is acknowledged as the source of the prototype photographs and experimental observations.

CONFLICT OF INTEREST

The source report does not provide a conflict-of-interest declaration. This statement requires confirmation by the authors before submission.

REFERENCES

- [1] Nordic Semiconductor, "nRF24L01+ Product Specification," version 1.0, 2008. Available: https://docs.nordicsemi.com/r/bundle/pdf_ps_nrf24l01p/. Accessed: Sep. 14, 2026.
- [2] Semtech Corporation, "LoRa and LoRaWAN." Available: <https://www.semtech.com/uploads/technology/LoRa/lora-and-lorawan.pdf>. Accessed: Sep. 14, 2026.
- [3] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, art. 1466, 2016, doi: 10.3390/s16091466.
- [4] I. Georgantas et al., "Integrated Low-Cost Water Quality Monitoring System Based on LoRa Network," *Electronics*, vol. 14, no. 5, art. 857, 2025, doi: 10.3390/electronics14050857.
- [5] C. F. Soon et al., "Smart sensing and anomaly detection for microalgae culture based on LoRaWAN sensors and LSTM autoencoder," *Aquaculture International*, vol. 33, art. 438, 2025, doi: 10.1007/s10499-025-02104-7.
- [6] Espressif Systems, "ADC," Arduino-ESP32 documentation. Available: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/adc.html>. Accessed: Sep. 14, 2026.
- [7] Google, "Enabling Offline Capabilities in JavaScript," Firebase Realtime Database documentation. Available: <https://firebase.google.com/docs/database/web/offline-capabilities>. Accessed: Sep. 14, 2026.