2025 6th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)
26 – 28 November, 2025, Ruse, Bulgaria

Low-Power Motion Event Detection in Maritime Environments

Mohamad Omran
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
ORCID: 0009-0003-5599-9907

Tim Brockmann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
ORCID: 0000-0002-2955-749X

Benjamin Rother
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
ORCID: 0000-0002-9456-0900

Frank Golatowski
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
ORCID: 0000-0003-0848-7784

Christian Haubelt
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
ORCID: 0000-0002-1568-5423

***Abstract*— Efficient motion monitoring on cargo ships is essential for ensuring cargo safety and enabling predictive maintenance. This paper presents a low-power, event-driven sensing approach designed for maritime environments with limited energy availability and constrained communication infrastructure. The proposed node integrates a Bosch BMI160 inertial sensor with a Heltec Wi-Fi LoRa 32 board and operates over LoRaWAN using a hybrid transmission model that combines periodic updates with event-triggered messages based on two motion thresholds. Light-motion events are logged locally for later retrieval, while high-impact events are transmitted on detection. Laboratory and shipboard trials indicate reliable detection with low energy usage. The results support the feasibility of event-driven LoRaWAN sensing for long-term maritime monitoring and lay the groundwork for future improvements such as adaptive thresholds and lightweight machine-learning-based analytics.**

Keywords*— event-driven sensing, LoRaWAN, maritime IoT, low-power design, Hybrid Wireless Communication.

I. Introduction

Modern cargo ships are becoming increasingly connected as the maritime industry embraces digitalization and automation. A key driver of this transformation is the growing need to monitor cargo conditions in real-time to ensure safety, reduce risk, and improve operational efficiency. While technologies such as satellite communication and cloud-based analytics are enabling large-scale ship-to-shore data exchange, many onboard systems still rely on manual processes or energy-inefficient sensor networks [1].

Low-power wireless sensor networks (WSNs) are essential for enabling real-time monitoring in remote or hard-to-reach ship areas. However, continuous data transmission from these sensors can significantly drain battery-powered devices, especially in locations with restricted physical access. Reducing energy consumption is therefore a critical requirement for long-term sensor deployment in maritime environments.

Some industrial solutions, such as the Bosch Connected Industrial Sensor Solution (CISS), have shown the potential of real-time condition monitoring through multi-sensor data fusion. However, such systems are not open-source, nor are they optimized for long-range wireless communication, which limits their applicability in large and metal-dense maritime environments [2].

The CargoAssist project, funded by the German Federal Ministry for Economic Affairs and Energy (BMWE), introduces a standalone, modular IoT framework specifically designed for maritime cargo monitoring. The system combines multiple wireless technologies, LoRa for long-range, low-power transmission; Wi-Fi Mesh for high bandwidth intra-ship networking; and BLE for local configuration, to create a flexible, resilient communication architecture [3].

This paper presents a prototype implementation focusing on motion event detection using the BMI160 inertial sensor from Bosch Sensortec. Motion and vibration are critical indicators of possible cargo instability or mechanical faults, especially in relation to phenomena like cargo liquefaction or engine imbalance [4]. Instead of streaming continuous data, our approach leverages the BMI160's internal interrupt features to operate in event-driven mode. The sensor wakes up and transmits data only when specific motion patterns are detected, drastically reducing energy usage while maintaining responsiveness.

Strategic placement of motion sensors across the ship improves monitoring. In cargo holds, they can detect unusual vibrations or cargo shifts indicating instability or liquefaction risks, enabling preventive actions, such as reducing speed or securing cargo before the situation escalates. On the deck or structural frame, they capture vessel dynamics in rough seas, providing real-time feedback on pitch, roll, and yaw to support route planning and ship behavior models.

Motion sensors can also be placed in the engine room to detect deviations in mechanical behavior. A change in vibration signature, compared to learned baselines, can reveal early signs of malfunction, misalignment, or wear. By applying machine learning techniques, such as one-class

979-8-3315-0119-8/25/$31.00 ©2025 IEEE

SVMs (Support Vector Machines), the system can gradually learn normal vibration patterns and detect deviations, enabling predictive maintenance strategies. These applications are integral to the vision of CargoAssist, where smart, distributed sensing allows the ship to self-diagnose and support operational decisions autonomously.

To support energy efficiency, solar panels can be integrated into the sensor units placed on the ship's surface, while in enclosed areas such as the engine room, thermoelectric or vibration-based harvesters can provide complementary power.

Initial results from this implementation (part of the broader CargoAssist system) demonstrate a significant reduction in power consumption while capturing all motion events without loss. These early findings contribute to the broader discourse on reliable, self-sustained maritime IoT systems.

II. BACKGROUND AND KEY TECHNOLOGIES

A. LoRa

LoRa (Long Range) is a wireless communication technology designed to transmit small amounts of data over long distances with minimal energy consumption. It uses Chirp Spread Spectrum (CSS) modulation, where data is encoded in slowly sweeping frequencies called chirps. This modulation scheme makes the signal robust against noise and interference, allowing receivers to detect messages even when the signal is below the noise floor.

Operating in the sub-GHz ISM bands (such as 868 MHz in Europe), LoRa is particularly well-suited for environments with physical obstructions and metallic structures, such as the interior of cargo ships, where traditional wireless signals may suffer from severe attenuation.

One of LoRa's key strengths is its high receiver sensitivity, which enables long-range communication without requiring high transmission power. This feature makes it ideal for battery-operated sensor nodes that need to remain active for months or years without maintenance.

The trade-off, however, is a limited data rate. As the communication range increases, the symbol duration also increases, reducing throughput. This makes LoRa more appropriate for low-frequency status updates and event-triggered data rather than high-speed or real-time communication.

B. LoRaWAN

LoRaWAN defines the communication protocol and system architecture for low-power, wide-area networks. It enables long-range wireless communication between distributed sensor nodes and central gateways, using unlicensed radio bands and a star-of-stars topology [5].

Devices typically join the network using Over-The-Air Activation (OTAA), where the end device initiates a `Join-request` and receives a `Join-accept` message from the network server. This process establishes session-specific encryption keys and authenticates the device, ensuring secure communication.

To illustrate the overall topology and data flow of a typical LoRaWAN system, we refer to a representative architecture based on the open-source ChirpStack platform (see Fig. 1). ChirpStack is widely used in research and industry due to its modular design, local deployability, and full control over the network infrastructure, which makes it suitable for environments with strict security requirements, as data remain within a private infrastructure. Other LoRaWAN server implementations, such as The Things Stack and Loriot, also exist.

In this architecture, LoRa Nodes transmit their data via LoRa to the LoRa Gateway, which runs a Packet Forwarder and Gateway Bridge. The Gateway Bridge converts the LoRa frames into `MQTT` messages and publishes them to specific `MQTT` topics on an internal `MQTT` broker. The Network Server subscribes to these topics, manages device sessions, validates frames, and handles adaptive data rate. The Network Server and Application Server communicate internally using `gRPC`, a high-performance remote procedure call framework based on `HTTP/2`. The Application Server then processes the payloads and provides device data to applications through APIs (e.g., `HTTP`, `MQTT`) and a web interface. The exact communication flow may differ slightly depending on the ChirpStack version.

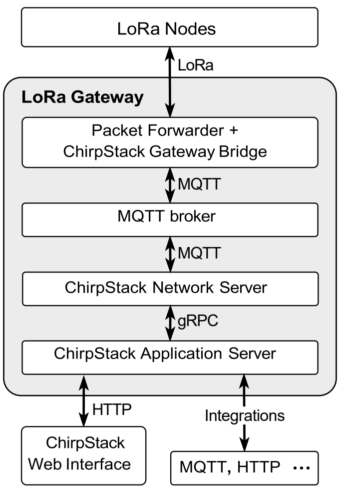

Fig. 1. LoRaWAN Architecture based on ChirpStack.

C. LoRaWAN Device Classes and Transmission Behavior

LoRaWAN defines three device classes: A, B, and C. These classes determine how and when end devices can receive messages from the network. Each class provides a trade-off between energy consumption and downlink availability.

Class A: is the most energy-efficient mode and is supported by all LoRaWAN devices. In this class, the device controls when communication happens. It initiates the process by sending an uplink message and then opens two short receive windows (RX1 and RX2) for possible downlink. Between transmissions, the device remains in deep sleep to save power. Because of its low energy footprint, Class A is used in the CargoAssist system for battery-powered sensor nodes.

Class B: introduces additional receive windows that are scheduled using synchronization beacons sent by the network. This allows the network to know when a device is listening, enabling more predictable downlink delivery. However, this requires extra energy for periodic listening.

Class C: devices keep their receive window open almost all the time, except during transmission. This provides the lowest latency for downlink messages but also consumes significantly more energy.

Fig. 2 illustrates a typical Class A communication exchange. The device begins by performing a `Join-request` using Over-The-Air Activation (OTAA). The network responds with a *`Join-accept`*, completing the secure registration and allowing future communication. After joining, the node can send application data, such as sensor readings, using uplink messages.

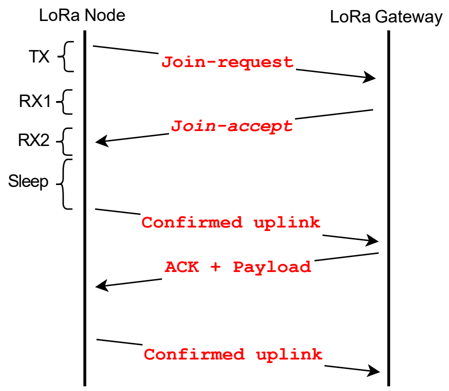

Fig. 2. LoRaWAN Class A Uplink-Downlink Exchange.

If a confirmed uplink is used, the device expects an acknowledgment (ACK) from the network. This response may also contain application payload (e.g., configuration data). The response can arrive during one of two receive windows:

- **RX1**: Opens after a fixed delay (default 1 second) using a frequency tied to the uplink channel and a downlink data rate derived from the uplink.

- **RX2**: Opens later (default 2 seconds after transmission) and uses a predefined frequency and lower data rate. This improves the reliability of downlink delivery, especially over longer distances or noisy channels.

If a valid response is received in RX1, RX2 is skipped to save power. Fig. 3 shows the complete transmission cycle of a Class A device. After transmission and the RX windows, the device returns to sleep mode until it is triggered again, either by an internal timer (for periodic updates) or an external interrupt (e.g., motion detection).

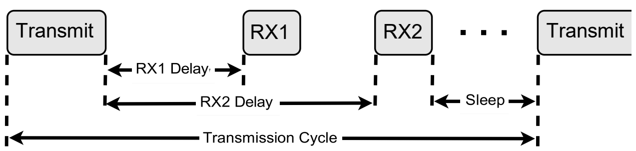

Fig. 3. LoRa Node Class A Transmission Cycle

This mechanism provides a balance between communication reliability and energy efficiency. It fits well with CargoAssist's use case, where sensor nodes need to operate autonomously for long periods while still ensuring that critical data is transmitted reliably.

D. Event-Driven IoT

The Internet of Things (IoT) involves sensors and devices that collect data from the physical world and transmit it for processing. In maritime environments like ships, these devices are often battery-powered and placed in areas where access is limited. This makes both energy efficiency and data reliability critical design concerns.

A common method in many IoT systems is periodic data transmission, where sensors send updates at fixed time intervals. While simple to implement, this method has two major limitations. First, it consumes energy even when no relevant changes occur. Second, and more importantly for dynamic environments, it can miss short, time-sensitive events. Sensors like motion detectors or light sensors may observe brief changes that go unreported if the *sampling interval* is too slow.

To address these issues, CargoAssist applies different transmission strategies depending on the sensor type. For sensors that monitor slowly changing values, such as temperature or humidity, periodic transmission is sufficient. However, for sensors that respond to fast events, such as motion, an event-driven approach is more appropriate.

In this paper, we focus on the *BMI160* motion sensor, which is used to detect movement patterns onboard cargo ships. The sensor is configured to operate in an interrupt-based mode: it stays in sleep mode most of the time and only wakes up to transmit when motion is detected. This event-driven sensing method ensures that critical motion events are captured without missing short occurrences, while also maintaining low energy consumption for long-term operation.

III. Related Work

LoRaWAN is widely used in industrial and maritime monitoring due to its long-range and low-power features. Most implementations rely on periodic transmissions, where sensors send data at fixed intervals regardless of actual changes. While simple, this approach can waste energy and miss short-duration events, especially in dynamic shipboard environments.

A. Event-Driven Communication

Recent work (e.g. [6]) shows that event-driven communication significantly reduces energy usage compared to periodic sampling, particularly when monitoring sparse or unpredictable signals. This model is well-suited for motion detection, where events are brief and require fast transmission. SamurAI [7] demonstrates an event-driven IoT node with embedded AI for on-device decision-making. While effective in some edge computing scenarios, local inference often increases power consumption. In contrast, the CargoAssist system adopts a hybrid approach: only minimal on-node processing is performed, such as interrupt-based event detection, while more complex tasks, such as adjusting thresholds or detecting motion patterns, are handled at the gateway or server side, where machine learning techniques can be applied without affecting node energy consumption.

B. Motion Sensing and LoRaWAN Integration

Early prototypes in the CargoAssist project used the GY-521 module based on the MPU6050 sensor (InvenSense/TDK) for motion detection. While this sensor was easy to integrate and delivered acceptable motion readings, its energy consumption was a major drawback. According to practical measurements reported by Wolles Elektronikkiste [8], the MPU6050 typically draws around 5.1mA during full operation, which makes it less suitable for low-power applications. In our laboratory setup, under the specific constraints of our application, the GY-521 module consumed an average of 6.45 mA. This level of power usage is too high for long-term operation in battery-powered LoRaWAN nodes.

To improve energy efficiency, we replaced the MPU6050 with the BMI160, a more advanced 6-axis IMU. Compared to *MPU6050*'s 14-bit resolution, the BMI160 provides 16- bit accelerometer resolution and improved noise performance, along with additional features such as gesture recognition and motion interrupts. Unlike the MPU6050, which has only one interrupt pin, the BMI160 features two programmable interrupt outputs. This allows the system to distinguish between different types of motion events and enables more adaptive behavior in the sensing logic.

Crucially, the BMI160 supports low-power operation modes, including sleep and suspend states. According to the datasheet, its typical current consumption in normal mode is around 950 µA, about 7 times lower than the GY-521 module.

C. Sensor Data Transmission Techniques

To the best of our knowledge, no published system implements fully event-driven motion sensing over LoRaWAN. Existing efforts, such as the RAKwireless motion detector with battery monitoring and the open-source LoRa-PIR-Motion-Detector [9] [10], demonstrate that motion events can be transmitted via LoRa but lack energy optimization and are not suitable for long-term use. Other studies have also pointed out fundamental limitations of LoRaWAN for event-triggered communication, showing that such transmissions may suffer from latency and packet loss in time-critical scenarios [11]. CargoAssist addresses these shortcomings by combining low- power motion detection with the BMI160 sensor and a hybrid communication approach: real-time events are sent via LoRaWAN, while all sensor data is also logged locally and can later be retrieved over a Wi-Fi mesh backbone. This ensures both on-detection responsiveness and reliable access to historical measurements, making the system more robust for shipboard environments.

IV. System Design and Proposed Method

A. Overview of the CargoAssist

A brief overview of CargoAssist is provided to frame our focus on enhancing motion-sensor data transmission using event-driven communication. As illustrated in Fig. 4, Smart Sensors can be deployed anywhere on the vessel (not only in the cargo hold). They use LoRaWAN for long-range, low-power communication and can also switch to Wi-Fi when higher data rates are needed, although the development of the Wi-Fi environment is not the focus of this paper and will be addressed later in the project. In the full system, LoRaWAN is used for all message types, not only event messages, covering periodic telemetry, configuration, and acknowledgments. All LoRa traffic converges to a single LoRa gateway mounted on the bridge, which terminates the LoRaWAN link and runs ChirpStack with lightweight services for device onboarding, payload decoding, and `MQTT/REST` forwarding. Nearby Wi-Fi gateways collect high-rate sensor traffic and interconnect over an IEEE 802.11s mesh for ship-wide coverage, while BLE is used during commissioning to securely add new Smart Sensors. This hybrid design delivers energy-efficient reach for events and reliable high-throughput paths when needed.

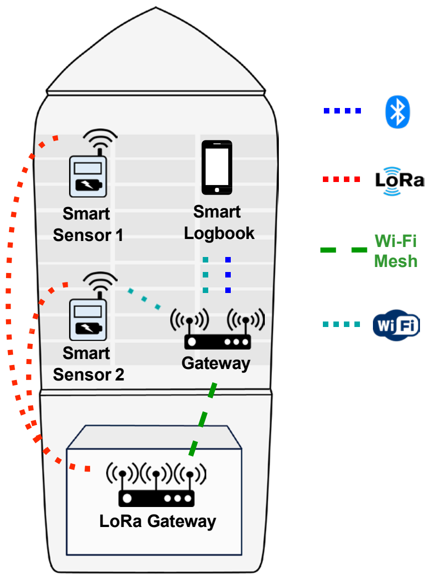

Fig. 4. CargoAssist Communication Diagram

B. Motion Event Detection with BMI160

The motion detection mechanism in CargoAssist is based on the Bosch BMI160, a low-power 6-axis IMU, integrated into a node built around the Heltec Wi-Fi LoRa 32 V3 board. This development board combines an ESP32-S3 microcontroller with integrated Wi-Fi, Bluetooth, and an SX1262 LoRa transceiver, and also includes an onboard OLED display and battery management circuitry, making it suitable for low-power IoT applications. After a successful join to the LoRaWAN network, the Heltec node enters a deep sleep mode, reducing its current consumption to about 850 µA. In this state, the microcontroller and most circuits are inactive, thereby maximizing battery lifetime.

To trigger communication, it relies on either a timer (for periodic telemetry) or motion-based interrupts from the BMI160 sensor. BMI160 provides two interrupt outputs (`INT1` and `INT2`), which are used in our system to differentiate between standard motion events and critical *high-impact* events.

`INT1` is configured for light motion detection. When triggered, it wakes up the microcontroller, which then records a burst of acceleration data to the SD card. Periodically, the node wakes up to transmit a summary of the last motion cycle, typically the three highest acceleration values, over LoRa.

`INT2`, on the other hand, is configured with a stricter threshold for *high-impact* motion. When such a threshold is exceeded (e.g., a sudden shock or cargo shift), the node wakes up and sends a confirmed uplink message with the strongest motion value recorded during the event. This behavior aims to transmit critical motion data at the earliest possible time, while less important events are stored for later analysis.

The benefit of this two-level threshold system is shown in Fig. 5. Events crossing only the lower threshold are queued for periodic transmission, while those exceeding the upper threshold (e.g., th3) trigger transmission on detection. This hybrid logic minimizes power usage while preserving responsiveness to important physical events.

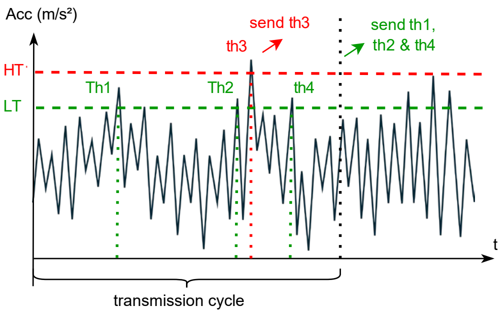

Fig. 5. Hybrid event/periodic logic: Low Threshold (LT): events with their timestamps are buffered and dispatched at cycle end; High Threshold (HT): events with their timestamps are sent on detection.

V. Proof of Concept

The current prototype was physically installed on board the cargo ship WESER STAHL to evaluate the LoRaWAN communication performance. Initial tests confirmed reliable connectivity across various locations, including the forward deck and the cargo hold, reaching the LoRa gateway installed on the ship's bridge. However, the full motion detection logic, including event-triggered transmission, has so far only been tested in a laboratory setting.

In the lab, artificial vibration scenarios were simulated to validate the system's behavior under various motion patterns. The design includes two parallel transmission logics (periodic and event-based) each optimized for different use cases.

The periodic transmission logic, as previously described in Section IV-B, uses `INT1` to detect light motion events. These events are logged to an SD card, and the node transmits a summary (the three highest values) after each motion cycle. This ensures that non-critical vibrations are captured with minimal energy cost.

The event-driven logic handles critical situations where acceleration exceeds the higher threshold (HT). In such cases, the node attempts to transmit the value immediately. However, due to LoRaWAN Time-on-Air (ToA) and regional duty-cycle regulations (e.g., 1% in the EU868 band), immediate transmission is not always permitted [5]. After each uplink, the stack must observe a mandatory off-time, during which no further transmissions are allowed. For a 1% duty cycle this means that the node must remain silent for roughly 99 times the previous airtime. Since the ToA increases with the spreading factor (SF) and payload size, the required waiting period also becomes longer under low data rates.

In a typical LoRaWAN implementation, if a transmission is attempted during this off-time, the stack defers the attempt; naive polling while waiting can keep the node awake and waste energy. To avoid this, a transmission *suppression window* was introduced. After any successful uplink, an *adaptive* delay is enforced during which the node refrains from further transmission attempts. If new HT events occur within this window, they are stored in memory. Once the window expires and transmission is again permitted, the node wakes up briefly and sends the top three values that were recorded during that period.

Fig. 6 illustrates this behavior. The use of a suppression window ensures the system complies with LoRaWAN limitations while maximizing energy efficiency. It also provides a structured way to prioritize data and reduce message redundancy under bursty motion conditions.

The LoRaWAN payload size is another limiting factor. At SF12, the maximum payload is typically around 50 bytes in the EU868 band (DR0, SF12/125 kHz), which is sufficient to include three motion values with timestamps and additional metadata defined by the CargoAssist protocol. It should be noted that the maximum payload size is region-specific, as defined by the LoRa Alliance specifications, and can vary across different regulatory bands. Among the included metadata is a latency-tracking field, which measures the delay between the actual sensing event and the reception of the corresponding uplink message. This provides valuable insight into the system's responsiveness and the real-world behavior of the wireless link.

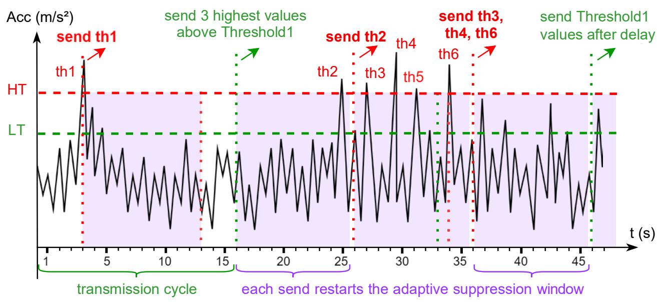

Fig. 6. Suppression-window prioritisation (LoRaWAN constraints): buffer events while restricted; at window end, send the top three ≥ High Threshold (HT) if present, else ≥ Low Threshold (LT).

A. Power Consumption Analysis

The power consumption behavior of the CargoAssist node was evaluated using an Otii Arc profiler, capturing current draw across typical operating states of the system.

The device operates in multiple power states triggered by motion events and transmission timing. TABLE I summarizes the measured current consumption obtained by the Otii Arc. At wake-up, a short power-on pulse of about ~ 650 mA occurs for roughly 92 µs as regulators and peripherals initialize.

During deep sleep, only the RTC domain and interrupt logic remain active, reducing consumption to ~ 857 µA. It should be noted that this value corresponds to the entire node, including the microcontroller, the BMI160 sensor in standby mode, and the SD-card reader. LoRaWAN uplink transmissions exhibit the highest node-level *average current* of ~ 200 mA, whereas the downlink receive windows draw ~ 81 mA. The transmission airtime varies with data rate and payload size. When the receiver was intentionally disabled on the network side, we consistently observed short listen windows of about ~ 57 ms (RX1) and ~ 178 ms (RX2). This behavior is expected: in the absence of a preamble, the stack closes each window after a symbol-count timeout (e.g., 8–16 symbols). Because RX2 usually uses a slower data rate (e.g., SF12/125 kHz), its symbol time is longer, so the same symbol-count timeout translates into a longer duration in milliseconds than RX1. Conversely, when an actual downlink is present, the radio must remain in RX long enough to capture the entire packet airtime (preamble + payload at the configured DR). Therefore, the effective window lasts much longer scaling with the downlink's data rate and payload often hundreds of milliseconds and even longer at low DRs.

These results are in agreement with the Heltec WiFi LoRa 32 datasheet, which reports typical TX currents of 200–230 mA at 14–22 dBm and RX currents of about 90 mA under nominal conditions [12]. The small deviations are expected, since our measurements capture the consumption of the complete node (ESP32-S3, regulators, and attached peripherals such as SD-card reader and BMI160 sensor), rather than the radio transceiver alone.

During Motion Processing, the node remains active to sample and pre-process BMI160 vibration data, extract and timestamp the three highest acceleration values of the current motion cycle, and apply the previously described event-driven transmission logic.

TABLE I. CURRENT CONSUMPTION PER OPERATING STATE

Operating State	Power & Timing Profile	
	Avg. Current	*Typical Duration*
power-on peak	~ 650 mA	92 µs
Startup + Sensor Initialization	~ 77 mA	1–3 s
Motion Processing	~ 68 mA	Variable
LoRaWAN uplink	~ 201 mA	60–350 ms
LoRaWAN downlink window 1	~ 81 mA	57–200 ms
LoRaWAN downlink window 2	~ 81 mA	178–350 ms
Deep Sleep	~ 857 µA	Minutes to hours

[a] Values measured at 3.3 V supply

B. Motion Behavior and Activity Profiles

Fig. 7 displays current consumption traces for representative system behaviors:

- Startup and LoRaWAN Join: Initial power peaks correspond to system boot and the OTAA join procedure.
- Scheduled Transmission: Periodic data sending (e.g., every 3 minutes) of the top three motion values exceeding the lower threshold (LT).
- Light motion (≥LT): Motion interrupt triggers a short wake-up to log data locally without transmission.
- High-impact motion (≥HT): A strong vibration triggers an on-detection uplink transmission, followed by a buffering phase, and later transmission of the top three recorded values after the delay window.

C. Real-World Test Results

To simulate long-term usage, the device was deployed outdoors for 24 hours, with the BMI160 sensor suspended via a flexible 1.5-meter tether to generate natural wind-induced vibrations. In both scenarios, the transmission interval was set to 3 minutes, and frequent *light-motion* events occurred. Without any external energy harvesting, the complete node drew an *average current* of approximately 4.5 mA, which represents the baseline consumption for estimating battery lifetime. When connected to a small 88×142 mm solar panel under partially cloudy weather, the measured net *average current* was reduced to 2.72 mA.

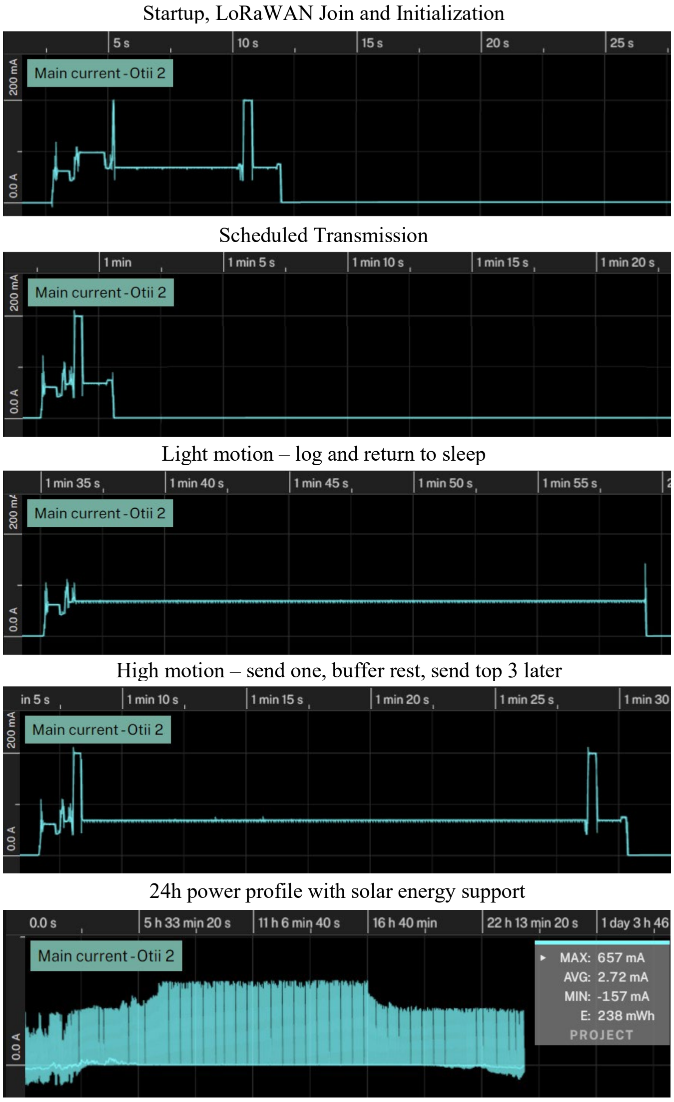

Fig. 7. Current consumption in typical operation scenarios.

This demonstrates that solar assistance can effectively improve autonomy, while the baseline consumption remains the key figure for battery-only scenarios. The node is enclosed in a 12×17.5×6 cm 3D-printed housing with a transparent cover that allows sufficient light to reach the solar cell, while all electronic components are securely mounted inside. The enclosure provides space for up to nine lithium batteries, which can be installed in parallel with a PCM/BMS providing protection and cell balancing for extended operation if no additional sensors occupy the compartment.

VI. Future Work

Future improvements will focus on enhancing adaptability and energy efficiency. As part of ongoing development, we are investigating dynamic Low Threshold (LT) adaptation. High Threshold (HT) is also adjusted accordingly, remaining slightly higher than Low Threshold (LT) to preserve the distinction between normal and critical motion. This enhancement aims to increase the system's sensitivity during quiet periods (e.g., no events or interrupts detected for an extended time) by lowering the threshold, and reducing sensitivity when frequent events occur by raising it. Fig. 8 illustrates this adaptive logic. Such a mechanism could further improve energy efficiency by minimizing unnecessary wake-ups during turbulence and enhancing detection performance during stable conditions.

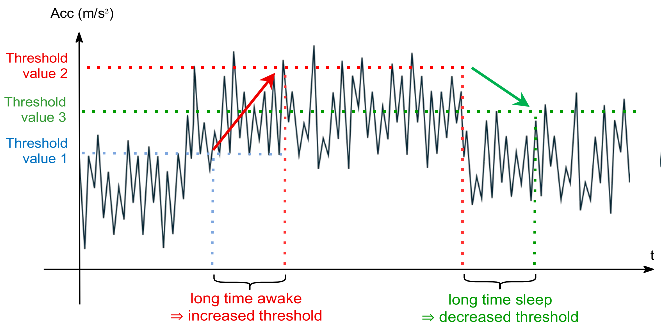

Fig. 8. Adaptive threshold logic.

VII. Conclusion

This paper presented a low-power motion detection system for maritime environments, providing reliable connectivity across the full length of a cargo ship. The approach leverages a hybrid event- and timer-based communication model to optimize energy usage while ensuring timely detection of critical motion events. By using two configurable thresholds, the system differentiates between minor and major vibrations and transmits data accordingly.

As proof of concept, we assumed a worst-case scenario by using a low motion threshold to emulate near-continuous activity on a cargo ship and enforcing a 3-minute reporting interval. Under these demanding conditions, the *average current* was ~ 4.5 mA on battery only and ~ 2.72 mA with light solar assistance. Consequently, on battery only, a single 2500 mAh cell sustains ≈ 23 days (2500 mAh/4.5 mA), while with light solar assistance the lifetime extends to ≈ 38 days (both exclud- ing self-discharge and temperature effects). This demonstrates strong potential for long-term autonomous deployment.

Future work will concentrate on the implementation of the dynamic threshold algorithm, followed by comprehensive validation of the complete system under real shipboard conditions. Additionally, machine learning techniques (such as lightweight decision tree classifiers or on-device anomaly detection models) could be trained on motion data collected during different ship states (e.g., docked, cruising, or in rough seas) to automatically identify significant vibration patterns. This would allow the system to refine threshold selection based on context, further enhancing both responsiveness and data relevance.

Acknowledgment

This work has been achieved in the project CargoAssist, which is funded by the maritime research program, an initiative of the German Federal Ministry for Economic Affairs and Energy (BMWE) under the funding code 03SX601B.

References

[1] L. Wang and H.-H. Hsu, "IoT technology in maritime logistics management: exploration of data analysis methods," *Discover Internet of Things, DOI: 10.1007/s43926-025-00167-9,* 2025.

[2] Bosch Connected Devices and Solutions GmbH, "Connected Industrial Sensor Solution (CISS) Datasheet," [Online]. Available: https://docs.rs-online.com/1f77/0900766b8172a0d5.pdf, 2018.

[3] M. Omran, T. Brockmann, F. Golatowski and C. Haubelt, "CargoAssist: Hybrid Communication Approach for Cargo Monitoring and Anomaly Detection," in *MareSec25*, DOI: 10.5281/zenodo.17120013, 2025.

[4] „Bulk Cargo Liquefaction," UK P&I Club, 2019. [Online]. Available: https://www.dnv.com/maritime/publications/bulk-cargo-liquifaction/.

[5] L. Alliance, „LoRaWAN® Link Layer Specification v1.0.4," [Online]. Available: https://lora-alliance.org/wp-content/uploads/2021/11/LoRaWAN-Link-Layer-Specification-v1.0.4.pdf, Fremont, CA, USA, 2020.

[6] C. Willuweit, C. Bockelmann und A. Dekorsy, „Energy and Bandwidth Efficiency of Event-Based Communication," DOI: 10.48550/arXiv.2303.01783, 2023.

[7] I. Miro-Panades, B. Tain, J.-F. Christmann, D. Coriat, R. Lemaire und C. Jany, „SamurAI: a Versatile IoT Node with Event-Driven Wake-Up and Embedded ML Acceleration," DOI: 10.48550/arXiv.2304.13726, 2023.

[8] Wolles Elektronikkiste, „MPU6050 – Accelerometer and Gyroscope," 2025. [Online]. Available: https://wolles-elektronikkiste.de/en/mpu6050-accelerometer-and-gyroscope.

[9] RAKwireless, „RAKwireless Learn Platform, "LoRaWAN-based Motion Detector with Battery Monitoring on WisBlock"," 2025. [Online]. Available: https://learn.rakwireless.com/hc/en-us/articles/33170102638615-lorawan-based-motion-detector-with-battery-monitoring-on-wisblock.

[10] Aizukanne, „GitHub Repository, "LoRa-PIR-Motion-Detector"," 2025. [Online]. Available: https://github.com/aizukanne/LoRa-PIR-Motion-Detector.

[11] I. Tomić, L. Bhatia, M. J. Breza und J. A. McCann, „The Limits of LoRaWAN in Event-Triggered Wireless Networked Control Systems," DOI: 10.48550/arXiv.2002.01472, 2020.

[12] Heltec Automation, „WiFi LoRa 32 V3.2 (HTIT-WB32LA) Datasheet," [Online] Available: https://resource.heltec.cn/download/WiFi_LoRa_32_V3/HTIT-WB32LA_V3.2.pdf, 2022.