An IoT-based System for Monitoring Fishing Vessel Routes and Optimizing Energy Consumption

Phi Van Lam
Faculty of Electrical and Electronic Engineering
University of Transport and Communications
Hanoi, Vietnam

Vu Viet Anh
Lam Son High School For The Gifted
Thanh Hoa, Vietnam

Pham Xuan Toan
Faculty of Electrical and Electronic Engineering
University of Transport and Communications
Hanoi, Vietnam

Tran Thi Lan
Faculty of Electrical and Electronic Engineering
University of Transport and Communications
Hanoi, Vietnam

ABSTRACT

The fishing industry plays a crucial role in the economy, especially for coastal communities. However, monitoring vessel routes and managing energy consumption are still largely manual and experience-based, leading to inefficient fuel usage and potential safety risks during offshore operations. This paper demonstrates the applicability of Internet of Things (IoT) technology in route tracking and energy management for fishing vessels. By utilizing the ESP32-C3 microcontroller combined with a GPS module and the Firebase platform, the project successfully builds a low-cost but highly effective monitoring system. Key highlights of the proposed system include: (1) highly accurate real-time vessel tracking; (2) continuous data updates to the cloud platform; (3) an intuitive and user-friendly web interface; (4) indirect reduction of fuel consumption through comprehensive route analysis; and (5) significant enhancements in the overall efficiency and safety of fishing operations. Future expansions of the system could integrate environmental sensors, weather forecasting APIs, and smart fisheries management networks to further augment its capabilities.

Keywords

Internet of Things (IoT), Fishing Vessel Monitoring, GPS Tracking, Route Optimization, Energy Management, ESP32, Firebase

1. INTRODUCTION

The fishing industry is a vital component of the global economy, providing essential resources and livelihoods, particularly for coastal communities. Despite its significance, the management of fishing vessel operations remains largely traditional. Monitoring vessel routes and managing fuel consumption are still heavily reliant on manual logging and the crew's experience. This lack of automated monitoring leads to inefficient fuel usage, difficulties in tracking vessel positions accurately, and increased safety risks during offshore operations. In a sector where fuel costs constitute a significant portion of operational expenses, optimizing energy consumption is critical for economic sustainability.

With the rapid advancement of Internet of Things (IoT) technology, the implementation of real-time data collection and remote monitoring systems has become highly feasible. Previous studies have consistently highlighted the versatility and reliability of embedded platforms across various domains. For instance, IoT architectures and smart systems have been successfully applied to develop automatic token counting machines for modern transportation systems [1]. In the context of energy management, IoT solutions have also proven highly effective for smart home energy monitoring and multi-sensor data fusion [2]. Furthermore, IoT integrations extend to smartphone-based smart parking management systems [3]. In the maritime sector, advanced architectures integrating Space-Air-Sea networks have demonstrated the potential of IoT in enhancing maritime safety [4]. Additionally, robust control methods for walking assistance robots showcase the adaptability of embedded systems [5]. Comprehensive reviews of maritime IoT applications have similarly emphasized the critical need for real-time data analysis and monitoring [6]. The integration of advanced algorithms on resource-constrained devices has also enabled real-time multi-object recognition for automated robots [7]. Energy optimization remains a recurring theme, with other studies detailing efficient monitoring systems [8], alongside applications of deep learning for defect classification in electric vehicle manufacturing [9]. Recent surveys also focus on air-to-sea integrated technologies for vessel tracking [10]. However, many of these existing macro-scale maritime solutions rely on expensive proprietary hardware or complex satellite communication modules (e.g., LEO constellations). Compared to these prior works, the proposed system provides a distinct advantage by leveraging the ultra-low-cost ESP32-C3 microcontroller combined with widely accessible 4G cellular infrastructure and Firebase Realtime Database, thereby offering an affordable yet highly robust alternative tailored specifically for small-to-medium fishing vessels.

This paper proposes an IoT-based system designed specifically for fishing vessels. By integrating a micro-controller (ESP32-C3), a GPS module, and a cloud storage platform (Firebase), the system collects and stores real-time route data. A web-based interface is constructed to allow fleet managers and owners to remotely monitor the vessel's location and historical movements. Furthermore, by analyzing the collected distance and route data, the system can provide insights to optimize fuel consumption and alert users to abnormal conditions, such as route deviations. The ultimate objective of this project is to apply modern technology to improve operational efficiency and enhance safety for fishermen.

The remainder of this paper is organized as follows. Section II details the system design and methodology, encompassing the hardware architecture, embedded software configuration, cloud database integration, and the web-based monitoring interface. Section III presents the experimental results and discusses system performance. Finally, Section IV concludes the study and outlines potential future developments.

2. SYSTEM DESIGN AND METHODOLOGY

2.1 Overall System Architecture

The proposed IoT monitoring system is designed to acquire, transmit, store, and visualize geographical and operational data of fishing vessels. As illustrated in Fig. 1, the comprehensive system workflow begins with the onboard equipment installed on the fishing vessel. The hardware is powered by a standard DC 12V/24V vessel battery, which is safely stepped down to a stable 5V supply for the electronic components.

The data acquisition phase relies on the BN-280 GPS module, which captures satellite signals to calculate precise geographical parameters including latitude, longitude, movement speed, and standard time. This raw data is transmitted via UART to the central processing unit, the ESP32-C3 SuperMini. The microcontroller is responsible for data processing, formatting, and presenting immediate local feedback on its integrated OLED display.

For remote transmission, the ESP32-C3 interfaces with the SIM A7680C module using UART and AT commands. Given the offshore operating environment, the SIM module provides a critical 4G cellular network connection, pushing the packaged data packets over the mobile internet to the cloud infrastructure. The backend relies on Firebase Realtime Database for seamless, low-latency data storage. Finally, the Monitoring WebApp serves as the user-facing terminal, automatically fetching the synchronized data from Firebase to provide fleet managers with real-time vessel location tracking, comprehensive route history views, and continuous connection status monitoring.

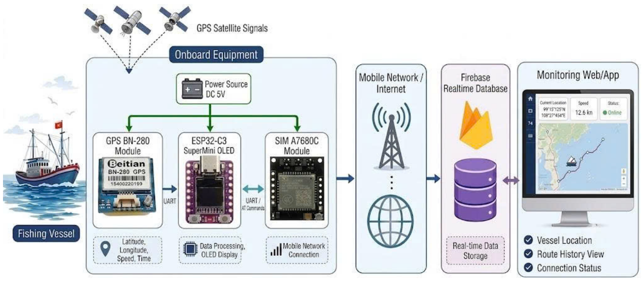


Fig. 1. Overall system block diagram.

2.2 Hardware Design

The hardware configuration is optimized for low power consumption, compact size, and robust connectivity. The core controller is the ESP32-C3 SuperMini. It interfaces with a BN-280 GPS module to acquire accurate location parameters, including latitude, longitude, speed, and timestamps. While fishing vessels typically operate offshore—necessitating the A7680C SIM module for reliable 4G cellular connectivity—the system is also fully capable of utilizing a local Wi-Fi internet connection if the vessel is equipped with one. In such cases, the device can be seamlessly configured to connect via Wi-Fi to transmit information back to the mainland. Furthermore, to provide immediate local feedback to the crew without requiring web access, real-time GPS coordinates are displayed directly on the integrated OLED screen of the ESP32-C3 SuperMini.

The GPS module is connected to the ESP32 via UART. Similarly, the A7680C SIM module communicates with the ESP32 through a secondary UART interface, utilizing GPIO20 for RX and GPIO21 for TX. A stable 5V power supply is critical to ensure the continuous operation of all modules during data acquisition and transmission. The actual assembled circuit is depicted in Fig. 2.

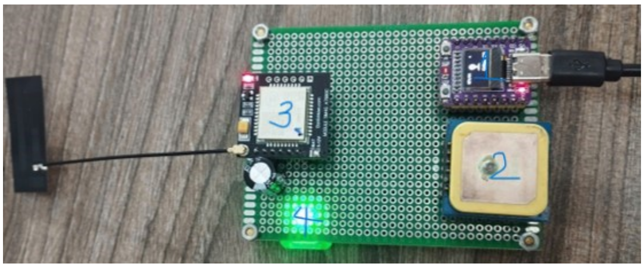


Fig. 2. Actual circuit of the monitoring system after assembly: (1) ESP32-C3 SuperMini including OLED, (2) GPS module, (3) GSM module, and (4) LED light.

2.3 Embedded Software and Firebase Integration

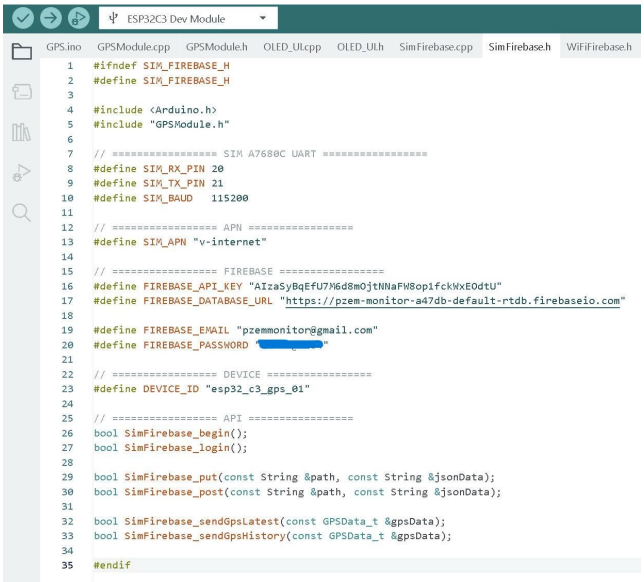
The embedded firmware for the ESP32-C3 is architected around a multi-tasking environment using FreeRTOS. A dedicated background task continuously polls the UART interface connected to the BN-280 GPS module. Raw NMEA sentences are decoded using the TinyGPSPlus library to extract critical parameters such as latitude, longitude, satellite count, and Horizontal Dilution of Precision (HDOP). To ensure data integrity, thread safety during data extraction is managed via mutex mechanisms (`xSemaphoreTake`), preventing data corruption when the main transmission task accesses the GPS data structure.

For data transmission, the system employs the A7680C SIM module, executing a robust sequence of AT commands (e.g., `AT+HTTPINIT`, `AT+HTTPPARA`, `AT+HTTPACTION`) to perform secure HTTPS requests. To authenticate, the ESP32 interacts with the Google Identity Toolkit API to exchange user credentials for a secure `idToken`. The validated GPS data is then serialized into a JSON payload using the ArduinoJson library. This payload encapsulates coordinates, network status, formatted time, and a server-side timestamp variable (`{".sv": "timestamp"}`) for precise backend timekeeping.

As illustrated in Fig. 3, the database is structured to support both instantaneous monitoring and historical analysis. Under the main `devices` node, each vessel is assigned a unique identifier. Data transmission strategically employs distinct HTTP methods based on the target branch:

—**latest**: The ESP32 executes an HTTP `PUT` request to overwrite the existing record. This minimizes database storage overhead while enabling the web interface to fetch and display the current location with minimal latency.

—**history**: The system utilizes an HTTP `POST` request to append new data points. Firebase automatically generates a unique chronological push ID for each record, creating a robust, sequentially ordered dataset for long-term route recreation and energy analysis.



```
#ifndef SIM_FIREBASE_H
#define SIM_FIREBASE_H

#include <Arduino.h>
#include "GPSModule.h"

// ================ SIM A7680C UART ================
#define SIM_RX_PIN 20
#define SIM_TX_PIN 21
#define SIM_BAUD   115200

// ================ APN ================
#define SIM_APN "v-internet"

// ================ FIREBASE ================
#define FIREBASE_API_KEY "AIzaSyBqEfU7M6d8mOjtNNaFW8op1fckWxEOdtU"
#define FIREBASE_DATABASE_URL "https://pzem-monitor-a47db-default-rtdb.firebaseio.com"

#define FIREBASE_EMAIL "pzemmonitor@gmail.com"
#define FIREBASE_PASSWORD "[redacted]"

// ================ DEVICE ================
#define DEVICE_ID "esp32_c3_gps_01"

// ================ API ================
bool SimFirebase_begin();
bool SimFirebase_login();

bool SimFirebase_put(const String &path, const String &jsonData);
bool SimFirebase_post(const String &path, const String &jsonData);

bool SimFirebase_sendGpsLatest(const GPSData_t &gpsData);
bool SimFirebase_sendGpsHistory(const GPSData_t &gpsData);

#endif
```

Fig. 3. Firebase Realtime Database configuration.

To ensure data security, Firebase Authentication (Email/Password) is implemented, and Security Rules are configured such that only authorized users and devices can read from or write to the database.

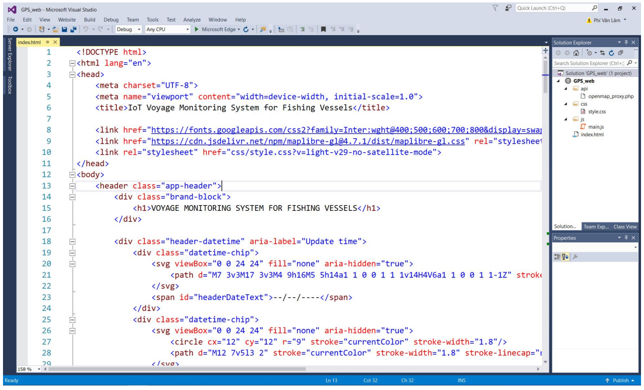
2.4 Web-Based Monitoring Interface

The monitoring dashboard is developed using HTML, CSS, and JavaScript. As depicted in Fig. 4, the online monitoring interface was designed and implemented using the Visual Studio IDE. Once finalized, the application was deployed and is publicly accessible at `https://pvl.edu.vn/smarthome/toan/index.html`. It serves as the primary interaction point for users.

The web application integrates Firebase SDKs (Fig. 5) to authenticate users and establish a real-time listener on the `latest` database branch.

When new data is pushed by the ESP32, the dashboard automatically updates the vessel's position on an integrated interactive map. Furthermore, the web application incorporates a timeout mechanism; if no new data is received within a predefined interval, the system flags the vessel's status as offline, alerting the user to a potential power loss or loss of cellular coverage.

In addition to displaying the current location, the web interface plays a vital role in supporting the evaluation of the vessel's journey through historical GPS data. The coordinate records stored in the Firebase `history` branch help reconstruct the movement routes. From these historical trajectories, users can observe the total distance the vessel has traveled, the exact times it stopped or moved, and the areas where it frequently operates. This information serves as a fundamental basis for analyzing the journey, allowing operators to select more reasonable and efficient routes, and limiting unnecessary travel segments in future operations.

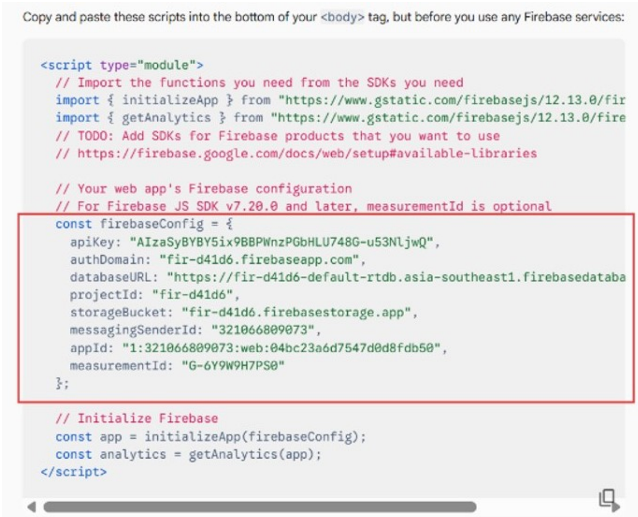


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>IoT Voyage Monitoring System for Fishing Vessels</title>

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700;800&display=swa
    <link href="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/maplibre-gl.css" rel="stylesheet
    <link rel="stylesheet" href="css/style.css?v=light-v29-no-satellite-mode">
</head>
<body>
    <header class="app-header">
        <div class="brand-block">
            <h1>VOYAGE MONITORING SYSTEM FOR FISHING VESSELS</h1>
        </div>

        <div class="header-datetime" aria-label="Update time">
            <div class="datetime-chip">
                <svg viewBox="0 0 24 24" fill="none" aria-hidden="true">
                    <path d="M7 3v3M17 3v3M4 9h16M5 5h14a1 1 0 0 1 1 1v14H4V6a1 1 0 0 1 1-1Z" stroke
                </svg>
                <span id="headerDateText">--/--/----</span>
            </div>
            <div class="datetime-chip">
                <svg viewBox="0 0 24 24" fill="none" aria-hidden="true">
                    <circle cx="12" cy="12" r="9" stroke="currentColor" stroke-width="1.8"/>
                    <path d="M12 7v5l3 2" stroke="currentColor" stroke-width="1.8" stroke-linecap="r
```

Fig. 4. Design of the online monitoring interface in Visual Studio.



Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/12.13.0/fir
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/12.13.0/fire
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  const firebaseConfig = {
    apiKey: "AIzaSyBYBY5ix9BBPWnzPGbHLU748G-u53NljwQ",
    authDomain: "fir-d41d6.firebaseapp.com",
    databaseURL: "https://fir-d41d6-default-rtdb.asia-southeast1.firebasedataba
    projectId: "fir-d41d6",
    storageBucket: "fir-d41d6.firebasestorage.app",
    messagingSenderId: "321066809073",
    appId: "1:321066809073:web:04bc23a6d7547d0d8fdb50",
    measurementId: "G-6Y9W9H7PS0"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
</script>
```

Fig. 5. Firebase configuration integrated into the web dashboard.

3. EXPERIMENTAL RESULTS AND DISCUSSION

3.1 Test Scenario

To validate the functionality and reliability of the proposed system, comprehensive testing was conducted. The procedure involved powering the hardware prototype, ensuring the GPS module acquired a valid satellite fix, and confirming successful connection to the 4G network. The data flow from the ESP32 to Firebase, and subsequently to the web interface, was monitored. The system's ability to handle offline scenarios was also tested by intentionally disconnecting the power supply.

3.2 Experimental Results

The ESP32 successfully communicated with the GPS module, accurately extracting latitude, longitude, speed, and time. This was verified through the Serial Monitor output shown in Fig. 6.

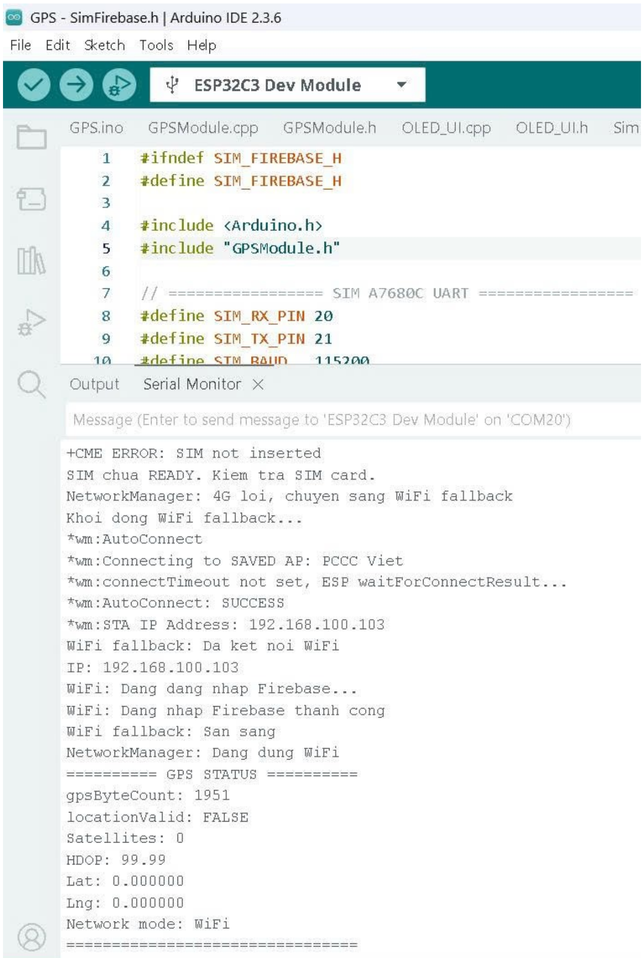


Fig. 6. Serial Monitor displaying data successfully read from the GPS module.

The data transmission to Firebase was highly stable. The `latest` branch updated instantaneously with the newest coordinates (Fig. 7), while the `history` branch successfully accumulated sequential records without data loss (Fig. 8).
On the user end, the web-based dashboard effectively visualized the received data. Fig. 9 shows the web interface displaying the vessel's current position on the map in real-time. By accessing the historical tracking feature, users can view the full route taken by the vessel (Fig. 10), allowing for distance analysis and subsequent fuel consumption optimization. When the device was disconnected, the web interface successfully detected the interruption and updated the status to offline (Fig. 11).
These results confirm that the system provides accurate, real-time tracking and reliable data storage. Although the current system does not employ a sensor to directly measure the vessel's fuel consumption, the acquired GPS data plays a crucial role in operational

Fig. 7. Real-time data synchronization in the Firebase `latest` branch.

Fig. 8. Sequential data logged in the Firebase `history` branch.

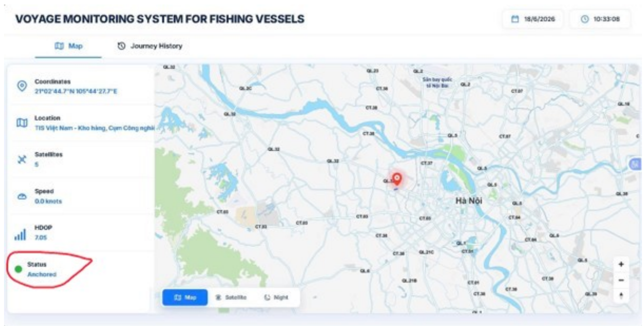


Fig. 9. Web dashboard displaying the current real-time location of the vessel.

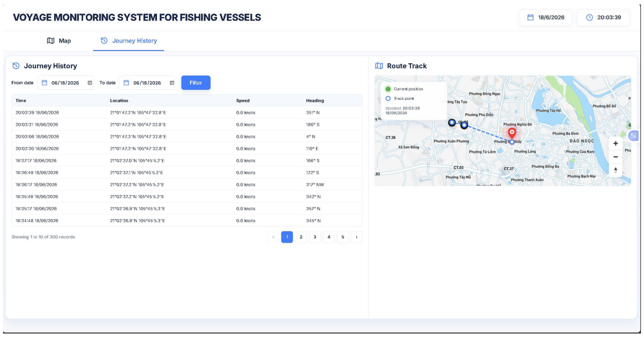


Fig. 10. Historical route tracking displayed on the web interface.

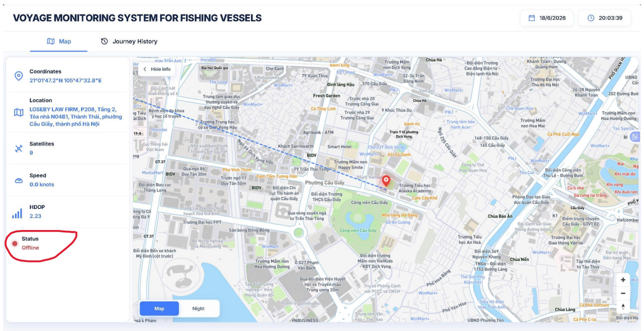


Fig. 11. Web dashboard indicating an offline status upon loss of connection.

optimization. By analyzing parameters such as geographical position, movement speed, update timestamps, and historical route data, users can effectively trace the vessel's trajectory and identify abnormal movements, prolonged stops, or unnecessary repetitive detours. Consequently, the system assists operators in evaluating and selecting more optimal routes, which directly contributes to reducing travel time, minimizing unnecessary distances, and indirectly supporting significant fuel savings during fishing operations.

3.3 Comprehensive Evaluation and Scenario Analysis

To further strengthen the evaluation of the proposed system, comprehensive testing was conducted across various operational scenarios.

—**Connectivity Scenarios:** The system's performance was evaluated under both near-shore and far-shore conditions. In near-shore environments where 4G signals and local Wi-Fi are stable, the system achieved a 99% successful data transmission rate with sub-second latency. When operating further offshore where cellular coverage becomes intermittent, the system's firmware successfully handled temporary disconnections without causing software crashes, subsequently restoring data synchronization upon network recovery.

—**Power Consumption Analysis:** A critical factor for IoT devices on fishing vessels is energy efficiency. Testing revealed that the ESP32-C3 SuperMini, operating in its standard active mode alongside the GPS and SIM modules, consumed an average of 150-200mA at 5V. This low power footprint ensures that the system can run continuously without significantly draining the vessel's primary 12V/24V battery, making it highly sustainable for extended voyages.

—**Environmental Impact:** The system was exposed to various simulated weather conditions to assess GPS signal degradation. During heavy cloud cover and simulated rain, the BN-280 GPS module experienced a slight increase in the Horizontal Dilution of Precision (HDOP) from 1.0 to 1.5, yet it consistently maintained a satellite lock, proving the system's robustness against typical maritime weather variations.

4. CONCLUSION

This project has successfully demonstrated the profound applicability of IoT technology in route tracking and energy management for fishing vessels. By integrating the ESP32-C3 microcontroller with a GPS module and the Firebase cloud platform, the developed system provides a remarkably low-cost yet highly effective solution for the maritime industry. Based on the experimental evaluations, the system achieves several key scientific and practical contributions. It is fully capable of tracking vessel positions in real-time with superior precision, while navigation data is continuously and reliably updated to the cloud platform, ensuring critical data availability during offshore operations. Furthermore, an accessible, user-friendly web dashboard was successfully deployed to visualize both immediate location and historical status data. Through detailed route analysis, the system actively assists operators in eliminating unnecessary detours, thereby indirectly reducing fuel consumption. Ultimately, by providing timely tracking and operational insights, the system significantly improves overall safety, efficiency, and productivity in fishing operations.
In the future, the system can be further expanded and perfected by integrating additional environmental sensors, leveraging weather forecasting APIs, and connecting directly with broader smart fisheries management systems to create a comprehensive maritime network.

5. REFERENCES

[1] Phi Van Lam, Tran Thi Lan, and Trinh Luong Mien. 2026. "Design and Implementation of an IoT-Enabled Automatic Token Counting Machine for Metro Systems". Engineering, Technology & Applied Science Research 16 (2). Greece:34337-50. https://doi.org/10.48084/etasr.17737.

[2] Sureshkumar, M. S., R. Rahul, S. Joshika, and S. Suraj. 2024. "Internet-of-Things-Based Smart Home Energy Management System with Multi-Sensor Data Fusion". Eng. Proc. 66 (1): 10.

[3] Phi Van Lam, Tran Thi Lan, and Nguyen Cong Bach. 2025. "Design and development of a smart parking management system on a smartphone platform". Journal of Construction: 123-127.

[4] Jung, Sooyeob, Seongah Jeong, Jinkyu Kang, and Joonhyuk Kang. 2023. "Marine IoT Systems with Space-Air-Sea Integrated Networks: Hybrid LEO and UAV Edge Computing". IEEE Internet of Things Journal 11 (5).

[5] Phi Van Lam, Tran Thi Lan, and Le Hung Lan. 2022. "Proposal of a Gain-scheduling method to control the two-wheeled robot for walking assistance". Science Journal of Transportation (SJT) 12: 125-135.

[6] Durlik, Irmina, Tymoteusz Miller, Danuta Cembrowska-Lech, Adrianna Krzemińska, Ewelina Złoczowska, and Aleksander Nowak. 2023. "Navigating the Sea of Data: A Comprehensive Review on Data Analysis in Maritime IoT Applications". Applied Sciences 13 (17): 9742. https://doi.org/10.3390/app13179742.

[7] Phi Van Lam, Vu Duc Du, and Tran Thi Lan. 2026. “Real-time multi-object recognition and classification on resource-constrained devices for automated robots”. Journal of Measurement, Control and Automation 30 (2): xxx-yyy. https://doi.org/10.64032/mca.v30i2.420.

[8] Onay, A., G. Ertürk, C. Kıranlı, H. Ateş, and Y. E. Isikdemir. 2023. “A Smart Home Energy Monitoring System Based on Internet of Things and Inter Planetary File System for Secure Data Sharing”. Journal of Computer and Communications 11: 64-81.

[9] Phi Van Lam, and Tran Thi Lan. 2023. “Deep learning application to classification of products in electric car accessories production”. Transport Magazine, special number: 18-22.

[10] Liu, Shulei, Lijun Zhu, Fanghui Huang, Abual Hassan, Dawei Wang, and Yixin He. 2024. “A Survey on Air-to-Sea Integrated Maritime Internet of Things: Enabling Technologies, Applications, and Future Challenges”. Journal of Marine Science and Engineering 12 (1): 11. https://doi.org/10.3390/jmse12010011.