Piston: Journal of Technical Engineering Vol. 9, No. 1, July (2025) 40-53

PISTON

Journal of Technical Engineering:

A Web-Based Navigation Control System for Lake Toba Cleaning Using NodeMCU ESP8266 and Pulse Width Modulation (PWM)

Frengki Simatupang[1,a)], Istas Pratomo Manalu[1], Pandapotan Siagian[1], Marojahan Mula Timbul Sigiro[1], Gerry Italiano Wowiling[1], Eka Stephani Sinambela[1], Ripandy Saragih[1], Grace Yosheva[1], Romaito Silaen[1]

[1]*Computer Technology, Vocational Faculty, Institut Teknologi Del Toba, 22381, Indonesia*

E-mail: [a)] frengki.simatupang@del.ac.id

Received: July 10, 2025 Revision: August 07, 2025 Accepted: August 10, 2025

Abstract: Waste pollution in Lake Toba has become a critical environmental issue, threatening both its natural beauty and ecological sustainability. Manual waste collection methods remain limited in terms of efficiency and operational reach. This study aims to design and evaluate a web-based navigation control system for a floating surface-cleaning device utilizing the NodeMCU ESP8266 microcontroller. The system enables real-time control of direction and motor speed through a web interface, employing Pulse Width Modulation (PWM) for precise speed regulation. A prototype-based engineering approach was adopted, encompassing system design, implementation, and performance testing on land and in water environments. The experimental results indicate that the system successfully responded to all navigation commands (forward, backward, turn, pivot, and stop) with 100% accuracy under a stable local Wi-Fi network. Motor performance in water was found to be approximately 15–20% lower than on land due to fluid resistance. Battery endurance tests showed an operational time of approximately 3 hours on land and 2.1 hours in water at a 60% PWM duty cycle. Overall, the system demonstrates effective and flexible performance and holds promise for further development through the integration of sensors, camera modules, GPS-based autonomous navigation, and LoRa communication.

Keywords: Lake Surface Cleaning, Motor DC, NodeMCU ESP8266, Pulse Width Modulation, Web-Based Control System.

Abstrak: *Pencemaran limbah di Danau Toba telah menjadi masalah lingkungan yang kritis, mengancam keindahan alami dan keberlanjutan ekosistemnya. Metode pengumpulan limbah secara manual masih terbatas dalam hal efisiensi dan jangkauan operasional. Penelitian ini bertujuan untuk merancang dan mengevaluasi sistem kendali navigasi berbasis web untuk perangkat pembersih permukaan air terapung dengan memanfaatkan mikrokontroler NodeMCU ESP8266. Sistem ini memungkinkan pengendalian arah dan kecepatan motor secara real-time melalui antarmuka web, menggunakan teknik Pulse Width Modulation (PWM) untuk pengaturan kecepatan yang presisi. Pendekatan rekayasa berbasis prototipe diterapkan, mencakup perancangan, implementasi, serta pengujian kinerja di darat dan di air. Hasil pengujian menunjukkan bahwa sistem mampu merespons semua perintah navigasi (maju, mundur, berbelok, berputar, dan berhenti) dengan tingkat keberhasilan 100% pada jaringan Wi-Fi lokal yang stabil. Kinerja motor di air tercatat sekitar 15–20% lebih rendah dibandingkan di darat akibat hambatan fluida. Uji ketahanan baterai menunjukkan waktu operasional sekitar 3 jam di darat dan 2,1 jam di air pada siklus kerja PWM 60%. Secara keseluruhan, sistem ini menunjukkan kinerja yang efektif dan fleksibel, serta memiliki potensi untuk dikembangkan lebih lanjut dengan integrasi sensor, modul kamera, navigasi otonom berbasis GPS, dan komunikasi LoRa.*

Kata kunci: *Pembersih Permukaan Danau, Motor DC, NodeMCU ESP8266, Pulse Width Modulation, Sistem Kendali Berbasis Web.*

INTRODUCTION

Lake Toba is the largest volcanic lake in Southeast Asia, with an approximate length of 100 kilometers and a width of 30 kilometers. Located in North Sumatra Province, Indonesia, the lake is not only renowned for its natural beauty and surrounding cultural heritage but also serves as a vital natural resource with significant ecological, social, and economic value [1]. Lake Toba has been designated as one of the priority destinations in Indonesia's Super Priority Tourism Destination (DPSP) development program [2]. However, the rapid growth of tourism and residential development around the lake has presented serious environmental challenges, particularly in terms of pollution. Domestic waste and tourist-generated trash—including plastics, food waste, and other inorganic materials—are frequently found floating on the lake's surface or stranded along its shoreline. This waste not only damages the lake's visual appeal but also pollutes the aquatic ecosystem and threatens the survival of aquatic flora and fauna such as fish and water plants. A report from the Ministry of Environment and Forestry (KLHK) indicates that pollution levels in water bodies, including Lake Toba, continue to rise due to the lack of an effective waste management system [3]. Currently, waste handling at Lake Toba is still carried out manually by local communities or sanitation workers. These efforts are limited in terms of reach, time efficiency, and their dependence on human labor. Therefore, an innovative solution is needed in the form of a waste-cleaning technology that is efficient, environmentally friendly, and does not disrupt the lake's ecological balance—thus supporting the long-term sustainability of Lake Toba's environment and tourism sector.

With the advancement of Internet of Things (IoT) technology and wireless microcontrollers, the control system for the Lake Toba cleaning device can be optimized by combining directional and speed control through a web-based interface [4]. Web-based control offers convenient remote access to the device without requiring special applications or additional hardware—only a standard web browser is needed. One of the devices that supports this approach is the NodeMCU ESP8266 [5]–[8], a WiFi-enabled microcontroller capable of hosting a local server and controlling digital devices.

In this study, a floating waste-cleaning device was designed to enable real-time directional control using two DC motors configured for forward, backward, left, and right navigation. In addition, the system adopts Pulse Width Modulation (PWM) to regulate motor speed according to operational requirements [9]–[11]. Users can send directional commands and select speed levels via a web-based interface accessible from mobile or desktop devices. These commands are then processed by the microcontroller and translated into mechanical actions. The novelty of this system lies in the integration of a local web interface with simultaneous control of both direction and speed of DC motors on a single WiFi-based microcontroller platform. This approach not only provides flexible manual control but also opens opportunities for further development toward automation and cloud-based integration in future implementations.

METHODS

This study adopts a prototype-based system engineering approach, encompassing the processes of planning, requirement analysis, system design, implementation, and testing of a control system for a floating waste-cleaning device. The system includes directional and speed control of DC motors through a web-based interface.

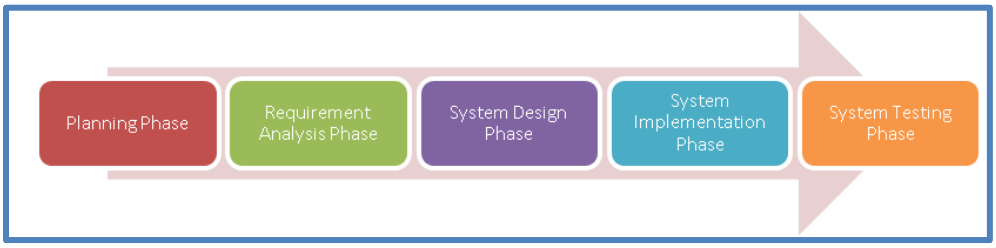

Figure 1. Research Implementation Stages

Figure 1 presents the System Development Life Cycle (SDLC) adopted in the development of a web-based navigation control system for Lake Toba cleaning, utilizing NodeMCU ESP8266 and Pulse Width Modulation (PWM) technology. The diagram outlines five key phases, each representing a crucial step toward building a functional and efficient embedded system that integrates both hardware and software components.

1. **Planning Phase**

   In this initial phase, the scope and objectives of the Lake Toba cleaning system are defined. This includes identifying the environmental challenges, system constraints, required hardware (e.g., NodeMCU, motor drivers, sensors), and expected software functionalities such as web-based control and real-time monitoring.

2. **Requirement Analysis Phase**

   Detailed technical and user requirements are gathered and analyzed. For the hardware aspect, this includes PWM motor control specifications, sensor integration, and power requirements. For the software side, the requirements include responsive web interface design, communication protocols (HTTP), and integration with the ESP8266 microcontroller.

3. **System Design Phase**

   This phase involves the architectural design of both hardware and software components. For the hardware, circuit schematics, pin mappings, and mechanical layout are prepared. The software design includes database schema (if any), user interface mockups, and flowcharts for system logic such as motor direction control using PWM.

4. **System Implementation Phase**

   The actual system is developed in this phase. Hardware components such as the NodeMCU ESP8266, motors, and sensors are assembled and connected. Simultaneously, the software system including the web-based control dashboard, backend code, and PWM control logic is coded and integrated. Firmware is uploaded to the microcontroller and tested for basic functionality.

5. **System Testing Phase**

   In this final phase, the system undergoes extensive testing to ensure all components function as expected. Functional tests are conducted to validate the navigation accuracy, PWM motor control, and responsiveness of the web interface. Integration tests ensure seamless communication between the front-end, back-end, and hardware modules. Any detected bugs or performance issues are resolved before deployment.

System Testing Phase

The system is designed using the NodeMCU ESP8266 as the central controller, which also functions as a web server. The NodeMCU manages the WiFi network connection, receives HTTP requests from the browser, and processes user input to control the logic of the DC motor actuators. Two DC motors are used as the main actuators to propel the device across the water surface, while Pulse Width Modulation (PWM) signals are employed to regulate motor speed.

Figure 2. the architecture of the developed system comprises several key components that function cohesively to enable the web-based control of a mini surface-cleaning vessel. At the heart of the system is the power supply module, which utilizes a DC battery as the primary energy source. This battery provides a stable and continuous DC power supply to all electronic components, including the microcontroller, sensor modules, motor driver, and the main DC motors. The battery's strategic placement within the vessel ensures balanced weight distribution and consistent energy delivery, which is essential for prolonged operational reliability. To enhance environmental awareness and operational safety, an ultrasonic sensor is integrated into the system. This sensor operates by emitting high-frequency sound waves and measuring the time taken for the echoes to return after bouncing off surrounding objects. The resulting distance data is crucial for safe navigation, enabling the vessel to avoid obstacles and providing contextual feedback to the operator. The sensor's output is directly fed into the microcontroller for real-time processing.

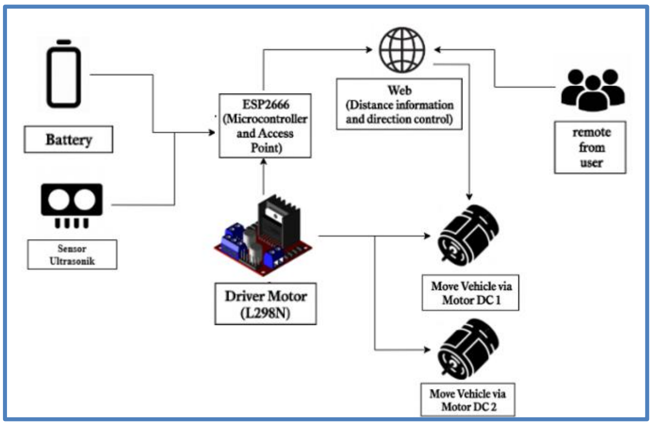

Figure 2. System Architecture

The core control and communication unit is the ESP8266, a Wi-Fi-enabled microcontroller that performs dual functions. As a microcontroller, the ESP8266 orchestrates all system operations, including reading and processing data from the ultrasonic sensor, receiving control commands via the web interface, and translating those commands into motor control signals. As a Wi-Fi module, it is configured as a local Access Point, creating a wireless network that hosts the web interface. This allows users—such as operators using smartphones, tablets, or computers—to directly connect to the vessel without needing an external network. Additionally, the ESP8266 can be configured to connect to an existing internet infrastructure, thereby enabling remote operation from greater distances via the global web.

The L298N motor driver module serves as the power interface between the ESP8266 and the DC motors. Since the ESP8266 cannot supply the current required to drive the motors directly, the L298N receives low-power logic signals from the microcontroller and converts them into high-power control signals suitable for operating the motors. It also supports directional control (forward/reverse) and speed regulation through Pulse Width Modulation (PWM), both of which are essential for precise navigation. The system utilizes two DC motors (Motor DC1 and Motor DC2) as its main actuators. These motors enable propulsion and directional control of the mini vessel. The dual-motor configuration supports a differential drive mechanism, whereby varying the rotational speed or direction of each motor allows the vessel to move forward, backward, turn left, or turn right. Each motor is independently powered and controlled via the L298N driver module [12].

User interaction with the system is facilitated through a web-based interface, which serves as the primary control console. Hosted directly on the ESP8266, this interface is accessible via any standard web browser on a connected device, such as a smartphone, tablet, or computer. Key functionalities of the interface include the display of real-time distance measurements from the ultrasonic sensor and speed feedback from both DC motors, providing the operator with a comprehensive view of the vessel's operational status. The interface also features navigation controls such as buttons or a virtual joystick, allowing the user to issue movement commands—forward, backward, turn left, turn right—which are then transmitted to the ESP8266 for execution. The user, or operator, interacts with the system through this web interface [13]. Whether connected through the ESP8266's local Wi-Fi network or a broader internet connection, the user is able to control and monitor the vessel remotely, ensuring flexibility and ease of use in a variety of deployment environments.

The system workflow begins when the battery powers all electronic components. The ESP8266 microcontroller initializes its internal processes and activates its Access Point (AP) mode, creating a local wireless network. Simultaneously, the ESP8266 begins periodically reading distance data from the ultrasonic sensor, which is essential for monitoring the vessel's surroundings. This distance information is continuously sent to the web-based interface and displayed in real time to the user for situational awareness. In parallel, the user accesses the web interface through their personal device—such as a smartphone or computer—connected to the ESP8266's Wi-Fi network. From the interface, the user issues navigation commands (e.g., forward, backward, turn left, turn right), which are received by the ESP8266. Upon receiving a command, the microcontroller processes the input and generates appropriate control signals that are sent to the L298N motor driver. The motor driver then activates Motor DC1 and Motor DC2 according to the received commands, adjusting their speed and direction based on Pulse Width Modulation (PWM) signals. As a result, the mini vessel moves and maneuvers across the lake in accordance with the user's instructions. This integrated control loop enables real-time remote navigation, providing a reliable and responsive cleaning operation on the water surface.

The proposed system architecture presents a comprehensive framework for web-based DC motor control in a lake surface-cleaning device. By integrating a microcontroller, sensors, actuators, and a web-based interface, the system offers an effective and flexible solution for remote navigation and monitoring. This approach not only facilitates real-time manual control but also establishes a solid foundation for further innovation. Potential future developments include the integration of additional environmental sensors, the implementation of autonomous navigation algorithms, and the enhancement of long-range communication reliability. Further advancements may also involve automated monitoring and control systems, as well as the development of mechanical components for detecting and retrieving floating waste. These enhancements are expected to improve operational efficiency, scalability, and the environmental impact of the system.

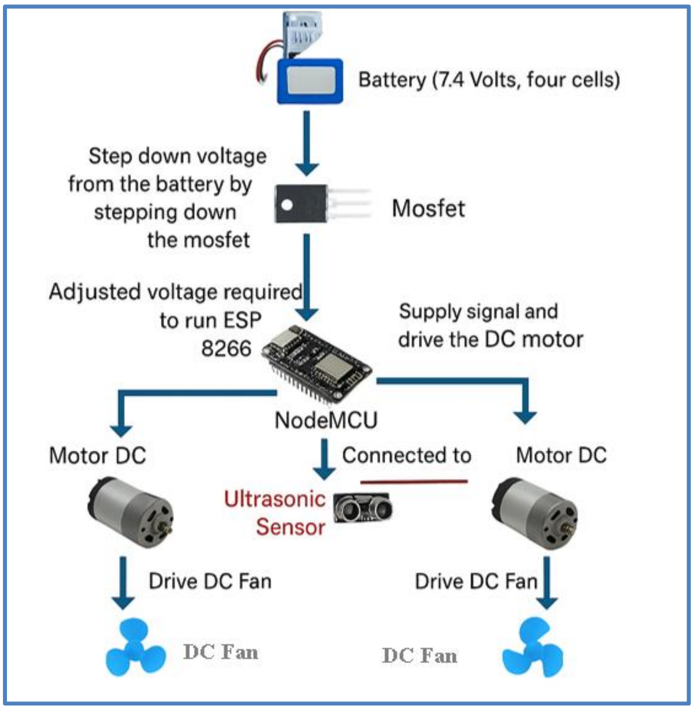

Figure 3. Schematic Design

Figure 3 illustrates the schematic design of the proposed hardware configuration utilized in the development of a web-based navigation control system for Lake Toba cleaning. The system is powered by a 7.4 V lithium battery composed of four cells, which serves as the main power source for the entire setup. To ensure voltage compatibility with the components, particularly the microcontroller, the battery output is routed through a MOSFET-based voltage regulation circuit. This circuit functions to step down the voltage to a level appropriate for the NodeMCU ESP8266 module, thereby protecting the microcontroller from overvoltage and ensuring stable operation. The NodeMCU ESP8266 serves as the core processing and communication unit of the system. It receives regulated voltage from the MOSFET and performs dual roles: executing control logic and establishing wireless communication through its built-in Wi-Fi capability. An ultrasonic sensor is directly interfaced with the NodeMCU to measure the distance between the cleaning vehicle and surrounding obstacles. This information is used for navigation and real-time feedback to the user. The microcontroller also generates control signals for two DC motors, each of which is mechanically connected to a DC fan. These fans act as thrusters, enabling directional movement of the vehicle on the water surface. The control signals from the NodeMCU determine the speed and rotation direction of the motors, allowing for precise maneuverability during cleaning operations. Overall, the schematic demonstrates a compact and efficient integration of power management, sensing, actuation, and wireless control. This hardware configuration enables a lightweight, portable system capable remote-controlled lake cleaning operations in environmentally sensitive areas such as Lake Toba.

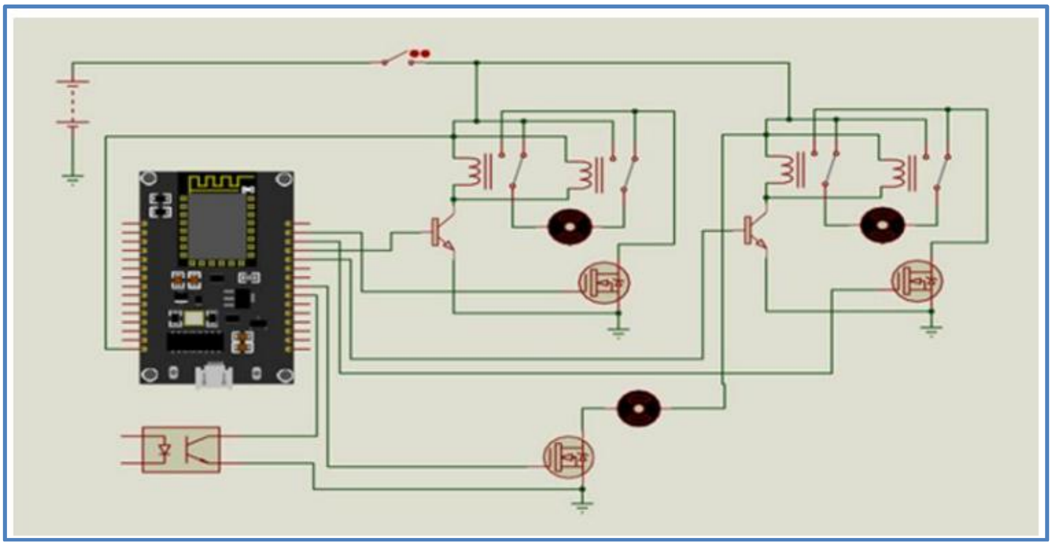

Figure 4. DC Motor Control Circuit Using Node-MCU ESP8266

Figure 4 illustrates the detailed circuit schematic for controlling two DC motors using the NodeMCU ESP8266 microcontroller. The design leverages an H-Bridge configuration that allows full bidirectional control of both motors via digital output pins of the NodeMCU. This motor driver setup is essential for enabling the cleaning vehicle to perform forward, backward, and turning maneuvers. The NodeMCU ESP8266 is positioned as the central control unit, utilizing its GPIO (General Purpose Input/Output) pins to deliver logic-level signals to the motor driver circuitry. Each motor is interfaced through a pair of transistors (likely MOSFETs or BJTs) arranged in an H-Bridge topology. This configuration allows for the control of current direction across the motor terminals, which in turn controls the direction of motor rotation. The coil symbols indicate the presence of relay modules or inductive loads (possibly representing electromagnetic control switches or physical motor windings). Each H-Bridge is responsible for one DC motor and consists of four switching elements that are activated in pairs to reverse the polarity across the motor terminals as needed. Each switching element in the H-Bridge is controlled by the NodeMCU through its output pins. Pull-down resistors or opto-isolators may also be present to protect the microcontroller and ensure reliable switching. The power supply rail (indicated by the uppermost line in the schematic) provides voltage to the motors and driver components, while the bottom rail connects to ground. A flyback diode may be included across each motor to protect the circuit from voltage spikes due to back EMF (electromotive force), although it is not explicitly shown here. The schematic effectively enables real-time motor control through PWM or logical HIGH/LOW signals generated by the NodeMCU. This allows precise navigation of the cleaning vehicle used in the Lake Toba cleaning system, ensuring agile movement and responsiveness to user commands transmitted via the web interface.

Mini Vessel Directional Control Logic

Directional control of the cleaning vessel is achieved by individually activating each of the two DC motors. The control logic follows a set of predefined conditions based on user input received via the web-based interface. Table 1 outlines the motor activation combinations and their corresponding navigation results.

Table 1. Directional Control Logic

No	Web-Based Control Command	DC1 Motor (Left)	DC2 Motor (Right)	Resulting Vessel Movement
1	Forward direction	Forward Rotation (Full Speed)	Forward Rotation (Full Speed)	Moves Straight Forward
2	Reverse direction	Reverse Rotation (Full Speed)	Reverse Rotation (Full Speed)	Moves Straight Backward
3	Right turn direction	Forward Rotation (Full Speed)	Stop / Reverse (Limited Speed)	Turns Right
4	Left turn direction	Stop / Reverse (Limited Speed)	Forward Rotation (Full Speed)	Turns Left
5	Stop	Stop	Stop	Full Stop

The navigation system of the lake-cleaning vessel utilizes a differential propulsion mechanism, driven by two independently controlled DC motors. To achieve precise maneuvering, the system implements a motor control logic embedded in the ESP8266 microcontroller, which translates user commands from the web interface into motor actions. The motors are mounted symmetrically on the left and right sides along the longitudinal axis of the vessel. This configuration enables both translational motion (forward and backward) and rotational motion (turning and pivoting) through variations in the rotational direction and speed of each motor. By modulating these parameters appropriately, the vessel can perform complex movements accurately in response to real-time navigation input.

Linear translational motion [14] is achieved by simultaneously activating both DC motors to rotate in the forward direction, thereby propelling water backward and generating forward thrust. When both motors rotate at the same speed and in the same direction, the thrust force is symmetrical, resulting in a straight forward movement. Similarly, for reverse motion, both motors are commanded to rotate in the reverse direction, pushing water toward the front of the vessel. Maintaining equal rotational speed in this configuration ensures a purely backward translational movement.

Turning Maneuvers is accomplished by creating an imbalance in thrust between the left and right sides of the vessel. For a right turn, the motor on the left side is maintained at full forward speed, while the right motor is either stopped completely or rotated in reverse at a lower speed. This differential thrust causes the vessel to arc to the right. For a left turn, the control logic is the inverse. The right motor continues to rotate forward at full speed, while the left motor is stopped or reversed at a limited speed. These maneuvers allow for smooth, curved turning motions while maintaining stability. Stop (Braking) Motion, To bring the vessel to a complete stop, both DC motors are deactivated simultaneously. This can be accomplished either by cutting off the power supply to the motors via the motor driver, or by applying dynamic braking—a technique that briefly reverses the motor polarity to bring the rotor to a halt more quickly. This mechanism ensures quick and stable halting of the vessel when needed. All motion control logic is embedded in the NodeMCU ESP8266, which interprets user commands received through the web interface and converts them into control instructions for the L298N motor driver module. The driver interprets these digital instructions to determine both the rotation direction and motor speed using Pulse Width Modulation (PWM). The ability to modulate motor speed proportionally enables smooth gradations in turning and pivoting maneuvers, contributing to the vessel's high degree of operational flexibility. Directional commands are implemented using digital HIGH/LOW signals transmitted to the motor driver's input pins, according to the control structure defined in the microcontroller logic. This modular and programmable approach allows for precise, real-time motor control based on user navigation inputs delivered via the web-based control interface.

Speed Control Using PWM

Speed control of the DC motors in this system is implemented using the Pulse Width Modulation (PWM) technique, which is a widely adopted method in embedded systems for regulating motor performance. PWM enables the adjustment of the average power delivered to the motor by modulating the ratio between the signal's "on-time" and "off-time," without altering its base frequency. In other words, PWM manipulates the pulse width of a digital logic signal to control how fast the motor rotates. In this implementation, the NodeMCU ESP8266 generates PWM signals through its digital output pins. These signals are sent to the L298N motor driver module, which then modulates the rotation speed of the left and right DC motors independently. The PWM values are dynamically controlled through a web-based interface utilizing slider elements, allowing users to directly adjust the duty cycle of each motor's signal. The duty cycle ranges from 0% (motor off) to 100% (full speed), enabling precise and intuitive control. This method is not only energy-efficient, but also supports smooth and stable control responses, which is particularly important in dual-motor drive systems where simultaneous direction and speed control is required. Compared to traditional resistive control methods, PWM significantly reduces power consumption while maintaining performance. Several studies have confirmed that PWM is an effective method for controlling motor speed in microcontroller-based systems. It provides stable and responsive speed control, especially in mobile robotic applications such as load-carrying robots and autonomous vehicles. The practical application of PWM in such systems enhances both energy efficiency and motion precision.

The effective voltage delivered to the motor using PWM can be mathematically expressed as follows:

$$V_{effective} = V_{maks} * \frac{Duty\ Cycle\ (\%)}{100} \qquad (1)$$

Where $V_{effective}$ is the average voltage perceived by the motor, V_{max} is the maximum supply voltage, and the duty cycle determines the proportion of the signal in its "on" state. This formula underscores the direct relationship between the PWM signal's duty cycle and the resulting motor speed. The NodeMCU microcontroller generates Pulse Width Modulation (PWM) signals using the analogWrite() function. Speed values are transmitted from the web-based interface and then converted into PWM values—typically within the range of 0 to 1023. These values are subsequently applied to the PWM control channels of the motor driver to regulate the rotational speed of the DC motors [15].

Web Interface

Figure 5 presents the graphical user interface (GUI) of the web-based control system developed for the Lake Cleaning Project. This interface allows users to remotely control the movement and speed of two independent DC motors (left and right) responsible for maneuvering the floating cleaning device.

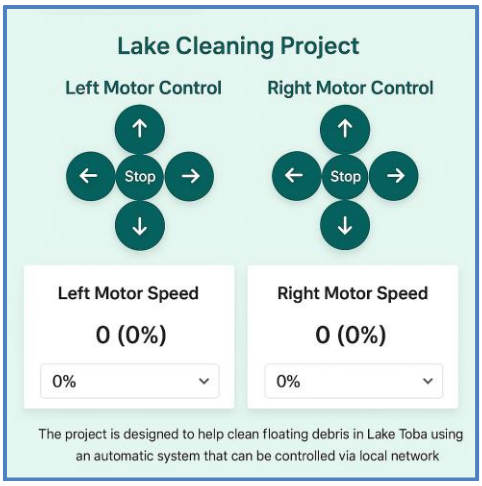

Figure 5. Web Interface Design

The control system divided into two primary sections: Left Motor Control and Right Motor Control. Each section provides directional buttons—forward, backward, left, right, and stop—to allow fine-grained control over the individual motors. This design enables differential drive control, which is essential for accurate navigation, including turning, rotating in place, or moving straight forward/backward. Below the directional controls are motor speed indicators for each motor. The speed displayed both numerically and in percentage, reflecting the pulse-width modulation (PWM) duty cycle sent to each motor. A dropdown selector provided to adjust the speed interactively. The selected value transmitted to the NodeMCU ESP8266 microcontroller through the local network. This web interface is hosted on the microcontroller's local server and is accessible through any web browser within the same network, eliminating the need for cloud-based infrastructure or external internet access. Each control button is linked to a specific URL endpoint (e.g., /forward, /setSpeed?val=60), which, when triggered by the user, sends a corresponding HTTP request to the microcontroller. The intuitive design promotes ease of use, even for non-technical operators. The interface plays a pivotal role in enabling real-time control of the cleaning device, ensuring responsive maneuverability for effectively removing floating debris from Lake Toba. This system demonstrates a practical implementation of IoT-based environmental monitoring and control.

Event-Driven Navigation Control System

The navigation control system for the lake surface waste-cleaning device is designed based on an event-driven architecture, in which each action is triggered by HTTP requests sent from the web interface to the NodeMCU ESP8266 microcontroller. The NodeMCU functions as a local web server, continuously listening for incoming user commands through the network. Upon receiving these requests, the NodeMCU processes the control input in real time to adjust both the direction and speed of the two DC motors—positioned on the left and right sides of the vessel. This architecture enables seamless real-time interaction between the user and the device, ensuring responsive and precise navigation over the surface of the lake.

```
// Pseudocode: Event-Driven Web-Based Motor Control System
Start System
Initialize WiFi Access Point
Initialize Web Server
Initialize PWM Pins for Motor DC1 (Left) and Motor DC2 (Right)
Loop forever:
    Wait for incoming HTTP Request from Web Interface
    If HTTP Request is received:
        Extract the command from URL (e.g., "/forward", "/left", "/stop", etc.)
        Switch (Command):
            Case "/forward":
                Set Motor DC1 to Forward at Full Speed
                Set Motor DC2 to Forward at Full Speed
            Case "/reverse":
                Set Motor DC1 to Reverse at Full Speed
                Set Motor DC2 to Reverse at Full Speed
            Case "/left":
                Set Motor DC1 to Reverse or Stop at Limited Speed
                Set Motor DC2 to Forward at Full Speed
            Case "/right":
                Set Motor DC1 to Forward at Full Speed
                Set Motor DC2 to Reverse or Stop at Limited Speed
            Case "/stop":
                Stop Motor DC1
                Stop Motor DC2
            Default:
                Ignore unknown command
End Loop
```

Each command issued by the user through the web interface—such as /forward, /left, /pivot_right, and others—is captured by the system and mapped to specific motor control actions. Control is achieved by regulating both the rotation direction (forward or reverse) and motor speed using Pulse Width Modulation (PWM) signals. The system follows a structured algorithm, which begins with the initialization of the NodeMCU ESP8266 as both an access point and a web server. Once initialized, the system enters a continuous loop mode, in which it actively listens for incoming HTTP requests. When a command is received, the system executes the corresponding motor action based on the mapped directional logic of the vessel:

a. Forward: Both motors rotate forward at full speed
b. Reverse: Both motors rotate in reverse
c. Turn Right: The left motor rotates forward, while the right motor is either stopped or reverses slowly
d. Turn Left: The right motor rotates forward, while the left motor is stopped or reversed
e. Stop: Both motors are deactivated

The event-driven architecture provides significant advantages in terms of resource efficiency and system responsiveness, as the NodeMCU only executes control logic upon receiving a command. This approach eliminates the need for constant polling or interrupts, resulting in a lightweight system that quickly responds to user input over a Wi-Fi network. The algorithm is implemented directly in Arduino C/C++, utilizing the ESP8266WebServer library. This software design integrates digital logic and PWM values, enabling real-time control over both direction and motor speed. As a result, the cleaning device can be precisely and flexibly controlled from a distance using nothing more than a standard web browser—without the need for additional software or applications.

RESULT AND DISCUSSION

Following the design and implementation process, the system successfully established a functional web-based control scheme for the lake surface waste-cleaning device. The NodeMCU ESP8266 was able to host a web page accessible via any standard browser on laptops or smartphones—without requiring any additional applications. The web interface displayed a set of navigation buttons (Forward, Reverse, Left, Right, and Stop) along with motor speed control, which could be adjusted either through numeric input or by selecting predefined speed levels. The evaluation phase was conducted in two stages: (1) functional testing of the control system, and (2) performance testing under simulated operational conditions. The tests were performed in a calm water pool environment, simulating lake conditions, and in an open outdoor area to evaluate the system's behavior and responsiveness under limited-range wireless control.

Functional Testing of Directional Control

Functional testing of the directional control system was conducted to verify that the web-based control interface could accurately direct the movement of the lake-cleaning device in accordance with user commands. The test covered five primary directional commands: forward, reverse, turn left, turn right, and stop. Each command was transmitted through the web interface, which was connected to the NodeMCU ESP8266. Upon receiving a command, the microcontroller adjusted the rotation of the left and right DC motors accordingly, utilizing Pulse Width Modulation (PWM) to regulate motor speed. The results of the testing demonstrated that the system successfully responded to all control inputs with a 100% success rate under stable local network conditions. The forward command triggered both motors to rotate simultaneously in the forward direction, producing straight forward motion. The reverse command caused both motors to rotate in reverse, allowing the device to move backward. When the left turn command was issued, the right motor continued rotating while the left motor was stopped. Conversely, the right turn command activated the left motor while stopping the right motor. The stop command effectively halted the rotation of both motors simultaneously. These results confirm that the system is capable of executing real-time directional commands with high accuracy and responsiveness, validating the reliability of the web-based motor control architecture.

Table 2. The elemental composition of ST 37

No	Control Command	Left Motor DC1	Right Motor DC2	System Response	Testing Status
1	Forward	Forward	Forward	Device moves straight forward	Success
2	Reverse	Reverse	Reverse	Device moves straight backward	Success
3	Left Turn	Stop	Forward	Device turns left	Success
4	Right Turn	Forward	Stop	Device turns right	Success
5	Stop	Stop	Stop	Device comes to a complete stop	Success
6	Forward	Stop	Stop	Device comes to a complete stop	Failed
7	Forward	Stop	Forward	Device comes to a complete stop	Failed
8	Left Turn	Forward	Forward	Device comes to a complete stop	Failed
9	Right Turn	Stop	Stop	Device moves straight forward	Failed

Table 2 presents the results of the direction control functional testing conducted through the web-based interface of the lake cleaning system. The objective of this test is to verify whether the directional commands issued via the web interface correctly translate into the expected motor behavior and system response. Each test case involves a unique Control Command (e.g., Forward, Left Turn, Stop), along with the corresponding logic applied to the Left Motor (DC1) and Right Motor (DC2). Based on the combination of motor actions, the System Response describes the observed movement of the device, while the Testing Status indicates whether the outcome matches the expected behavior (Success or Failed). Test cases 1 to 5 demonstrate valid and successful responses for all basic movement commands:

1. Both motors moving forward or reverse result in the device moving straight in the respective direction
2. Stopping one motor while the other moves causes the device to turn (left or right)
3. Halting both motors brings the system to a complete stop

Conversely, test cases 6 to 9 expose combinations where the motors receive partial or conflicting inputs:

1. For instance, in test 6, issuing a Forward command while both motors are stopped fails to initiate any movement
2. Test 9 incorrectly executes the Right Turn command with both motors stopped, yet results in forward movement, indicating a possible logic error in motor control or software interpretation

These findings highlight that while standard movement commands are well-implemented, further refinement is necessary for handling edge cases or unexpected motor states to ensure reliable control across all conditions. Directional command testing was conducted by sending control inputs through the web interface and observing the resulting movement of the cleaning device. The results indicated that the system was capable of accurately responding to navigation commands, with an average response time of 0.85 seconds, measured from the moment a button was pressed to the initial movement of the vessel. No significant delays or command conflicts were observed during the testing process. The system maintained stable and consistent behavior throughout all trials. Based on these findings, it can be concluded that the web-based directional control system functions reliably and meets the fundamental navigation control requirements of the Lake Toba surface-cleaning platform.

DC Motor Speed Measurement

The DC motor testing was conducted to evaluate the effect of Pulse Width Modulation (PWM) values on the rotational speed of the motor (in RPM). The PWM values were controlled by the microcontroller and served to regulate the average voltage delivered to the motor. Testing was performed under two conditions: on land and in water, as the motor is intended to operate in aquatic environments. By comparing performance across these two conditions, the efficiency of the motor and the impact of fluid resistance on rotational speed could be analyzed.

Table 3. DC Motor Speed Measurement

No	Duty Cycle PWM (%)	Motor Speed (RPM) in Land Condition	Motor Speed (RPM) in Water Condition
1	0%	0	0
2	30%	1000	850
3	50%	1600	1300
4	60%	1800	1450
5	80%	2000	1600
6	90%	2100	1700
7	100%	2200	1800

Table 3 provides the measured rotational speed (RPM) of the DC motors under different Pulse Width Modulation (PWM) duty cycle values, tested in two different environmental conditions: land and water. The objective of this experiment is to evaluate how the duty cycle of the PWM signal influences the motor speed, and to analyze the impact of environmental resistance—particularly from water—on the motor's performance. The duty cycle was incrementally varied from 0% to 100%, and the corresponding motor speeds were recorded. At 0% duty cycle, the motor remains stationary in both land and water environments, as expected. As the duty cycle increases, the motor speed increases almost linearly in both conditions, indicating proper PWM control.

However, it is observed that the motor speed in water is consistently lower than that in land at every duty cycle level. For example, at 50% PWM, the motor reaches 1600 RPM on land, but only 1300 RPM in water, demonstrating a reduction of approximately 18.75% due to water resistance. Similarly, at full speed (100% PWM), the motor achieves 2200 RPM on land compared to only 1800 RPM in water, revealing a substantial performance gap caused by fluid drag.

At a 0% duty cycle, no PWM signal is transmitted, resulting in no motor rotation. The rotational speed increases significantly as the PWM value rises. Motor performance in water is consistently 15–20% lower due to the drag force exerted by the water on the motor load. The test results indicate that higher PWM values produce higher motor rotational speeds (RPM) [9]. However, the RPM achieved in water is consistently lower than in dry (land) conditions for the same PWM value. This demonstrates that water resistance imposes additional load on the motor, making it work harder and reducing its overall efficiency. These findings are essential for understanding motor workload and power requirements when operating in real-world aquatic environments.

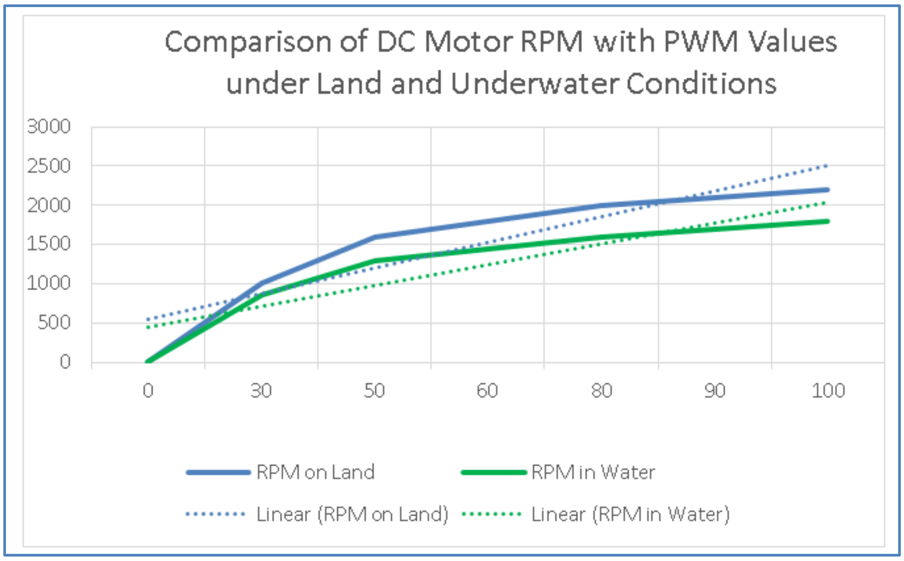

Figure 6. Comparison of DC Motor RPM

The following is a graph comparing DC motor speed (in RPM) against varying PWM Duty Cycle (%) under two conditions:

a. On Land (blue line): shows a consistent increase in RPM, reaching a maximum of approximately 2200 RPM at 100% duty cycle.
b. In Water (green line): RPM is lower due to fluid resistance, with a maximum of around 1800 RPM at 100% duty cycle.

This graph supports the conclusion that operating in water requires a higher PWM compensation to achieve a rotational speed comparable to that in dry conditions.

Speed Testing with PWM

Speed testing was conducted to evaluate the effectiveness of the system in controlling motor rotation speed using Pulse Width Modulation (PWM). PWM was applied to control the speed of the left (DC1) and right (DC2) DC motors proportionally, based on the directional requirements of the system [16]. Speed variations were set at full (100%), limited (approximately 60%), and stop (0%), depending on the control logic activated through the web interface. The testing was carried out under five primary control conditions: forward, reverse, right turn, left turn, and stop, as shown in Table 4. The PWM settings for each directional condition successfully produced movements that matched the expected control logic. In forward and reverse conditions, both motors received a PWM signal of 100%, resulting in stable straight-line motion either forward or backward. In the right turn condition, the left motor operated at 100% PWM, while the right motor was either stopped or received a limited PWM signal (60%), producing a smooth right turn. Conversely, in the left turn condition, the right motor continued at full speed, while the left motor received limited PWM or was stopped. For the stop command, both motors received no PWM signal (0%), causing the system to come to a complete halt.

Table 4. DC Motor Speed Test Results with PWM Based on Direction Control

No	Command Control	PWM Motor DC1	PWM Motor DC2	Outcome of System Movement
1	Forward	100% (Forward)	100% (Forward)	Moves Straight Forward
2	Reverse	100% (Reverse)	100% (Reverse)	Moves Straight Backward
3	Turn Right	100% (Forward)	60% (Reverse)	Turns to the Right
4	Turn Left	60% (Reverse)	100% (Forward)	Turns to the Left
5	Stop	0%	0%	Vessel Comes to a Complete Stop

Observations showed that the implementation of PWM provided effective and precise speed control over the system's directional movement. No voltage spikes or rotational imbalances were observed during the testing process. The delay caused by PWM transitions was also minimal, with an average speed adjustment latency of approximately 80–100 milliseconds, depending on the motor's initial condition. The strengths of this system lie in its simplicity of implementation, accessibility of the web interface, and flexibility in both directional and speed control. The system does not require a cloud server or additional software, making it well-suited for deployment in limited environments with local connectivity. However, some limitations were noted: Limited WiFi range (approximately 15 meters in open environments), System performance depends on the stability of the local WiFi connection, Lack of position feedback or onboard sensors, which means control remains entirely manual and operator-dependent. For future development, the system could be enhanced by integrating a camera module (e.g., ESP32-CAM), distance sensors, and semi-autonomous or GPS-based control systems to expand its operational coverage and improve the device's autonomy.

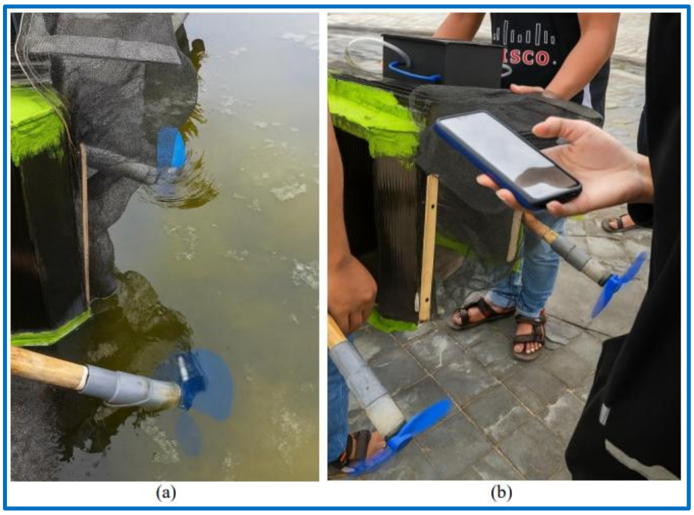

Figure 7. DC Motor Testing (a) Under Land and (b) Under Water Condition

Figure 7 presents the testing of a DC motor equipped with a propeller (fan), serving as the primary component of the propulsion system in the surface-cleaning device prototype. The testing was conducted under two different conditions to comprehensively evaluate the system's performance and stability. In the water-based test (Figure 7a), the DC motor was activated to drive the propeller while partially submerged in water. The aim of this test was to assess the thrust generated and the motor's efficiency under fluid load conditions. Observations focused on the device's stability, the strength of the thrust, and whether splashing or mechanical disturbances occurred within the propulsion system. In the land-based test (Figure 7b), the motor was tested without water load to directly observe the rotational speed of the propeller and to ensure that the power supply and control systems were functioning optimally. Testing on land also allowed for visual inspection of component integrity, including cable connections, motor alignment, and the structural stability of the propeller mount. Both tests were essential for comparing the performance of the DC motor under unloaded (free-air) and loaded (fluid resistance) conditions—critical factors for informing the final design of the propulsion system.

Battery Endurance Testing

Battery endurance testing was conducted to determine how long the system can operate based on the actual energy consumption of the DC motors under various duty cycle conditions and operating environments (on land and in water). For this approach, calculations were performed using the total battery energy (in Wh) as shown in Formula 2, and load power consumption (in Watts) as shown in Formula 3 [17]–[21].

$$Battery\ Energy\ (Wh) = \ Battery\ Voltage\ (V) \times Battery\ Capacity\ (Ah) \tag{2}$$

$$Power\ Consumption\ (W) = \ Voltage\ (V) \times CurrentDraw(A) \tag{3}$$

$$Operating\ Time\ (hours) = \frac{Battery\ Energy\ (Wh)}{Power\ Consumption\ (W)} \tag{4}$$

Using a 14.8V Li-Ion battery with a 2.2Ah capacity, the total available energy is 32.56 Wh. Based on the average current measurements across different test scenarios, the load power was calculated and used to estimate the system's theoretical operating time using Formula 4. Table 5 presents the results of the battery endurance test under two operating conditions: when the motors are running on land and when they are operating in water.

Table 5. DC Motor Speed Test Results with PWM Based on Direction Control

No	Motor Condition	PWM Duty Cycle	Average Current (A)	Load Power (W)	Theoretical Operating Time (Hours)	Actual Operating Time (Hours)
1	On Land	60%	1.0 A	14.8	2.20	3.0 hours
2	In Water	60%	1.3 A	19.24	1.69	2.1 hours
3	On Land	100%	1.5 A	22.2	1.47	2.1 hours
4	In Water	100%	1.9 A	28.12	1.16	1.4 hours

From Table 5, it can be observed that the actual operating time of the system is generally slightly longer than the theoretical calculation. This discrepancy is likely due to several factors: Load variation during testing, where the motor does not continuously operate at full capacity (e.g., during idle periods or brief stops). The efficiency of the motor driver and the power distribution system, which influences the actual current drawn. Tolerance in average current measurements, especially when using manual methods or non-datalogging multimeters. Nonetheless, a linear correlation between load power and operating time is clearly evident: the greater the load power, the shorter the operational duration. The energy and power-based approach offers a reasonably accurate estimation for predicting system runtime when designing and evaluating battery-powered embedded systems. Based on this test, it can be concluded that operating the DC motor at a 60% duty cycle results in more energy-efficient performance compared to full-speed operation (100%). Moreover, operating in water environments imposes greater load and higher current consumption compared to land conditions, thus requiring more careful power management in system design and deployment.

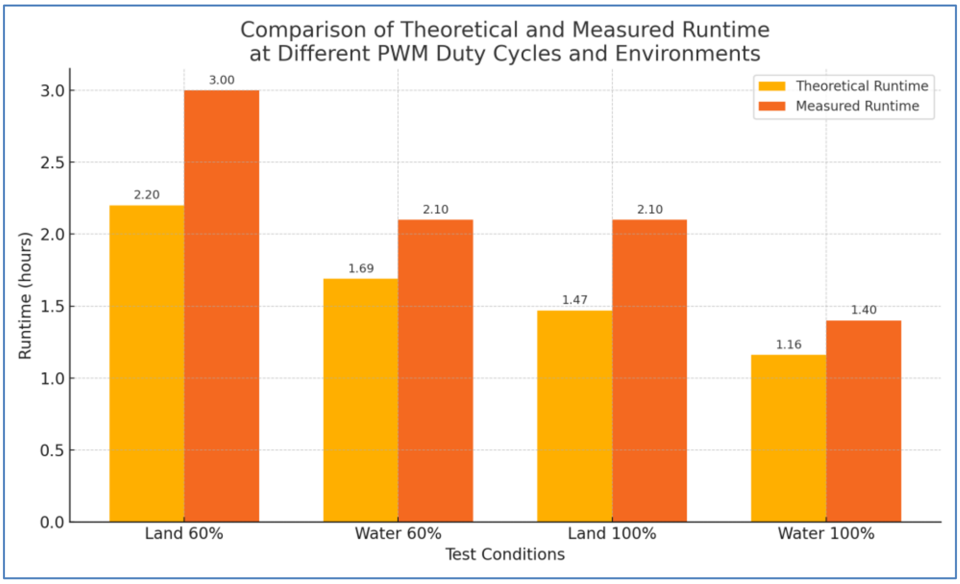

Figure 8. Comparative Analysis of Theoretical and Actual Operating Time in DC Motor System

Figure 8 presents a comparative graph between the theoretical operating time and the actual measured operating time, based on the testing conditions of the DC motor in two environments (land and water) and at two PWM duty cycle levels (60% and 100%). The graph shows a clear trend: operating time decreases as the PWM duty cycle increases, in both land and water environments. Additionally, operation in water results in shorter runtimes compared to land, indicating the presence of additional fluid resistance, which increases motor workload and current consumption. In general, a positive discrepancy is observed between theoretical and actual operating times. For example, at 60% PWM on land, the actual runtime was recorded at 3.0 hours, exceeding the theoretical estimate of 2.2 hours. This difference is likely due to the system not consistently operating under full load (i.e., intermittent operation), as well as higher motor control efficiency during non-continuous duty cycles. The graph further illustrates that the energy model based on Watt-hours provides a reasonably accurate prediction of power performance. However, it does not account for all system variables, such as motor efficiency, driver power losses, and dynamic conditions during operation.

CONCLUSIONS

This study aims to design and test a web-based control system for a surface waste-cleaning device on Lake Toba using the NodeMCU ESP8266, with direction and speed control of DC motors implemented through Pulse Width Modulation (PWM). The results of the implementation and testing show that the system can respond to navigation commands in real-time via a web interface without the need for additional applications. The system successfully controls directional movements (forward, reverse, turning, pivoting, and stopping) with a 100% success rate under a stable local network environment. DC motor testing demonstrated that PWM is effective in precisely controlling motor speed in both land and water conditions. However, the motor's rotational speed in water was observed to be approximately 15–20% lower than on land due to fluid resistance. These findings emphasize the importance of proper PWM tuning to ensure optimal system performance in aquatic environments. Furthermore, battery endurance testing revealed that the system's operating time on land is longer than in water. At a 60% duty cycle, the system could operate for approximately 3 hours on land and 2.1 hours in water. At a

100% duty cycle, the operating time decreased to about 2.1 hours on land and 1.4 hours in water. This confirms that power consumption in water is higher, due to the increased load on the motors caused by water resistance. These conclusions affirm that the developed system meets the basic functional requirements of a web-based navigational cleaning device and provide key insights into motor efficiency and power consumption across varying operational conditions. For future development, the system can be enhanced with the integration of distance sensors, a camera module (such as the ESP32-CAM), and GPS-based autonomous navigation to expand the cleaning coverage area and increase the device's operational autonomy. Additionally, the implementation of LoRa-based communication is suggested to extend control and monitoring capabilities.

REFERENCES

[1] B. Pratikno and P. Sidauruk, "Mempelajari Hubungan Air di Danau Toba dengan Air di Beberapa Sumber Mata Air Dekat Danau Toba," *J. Ilm. Apl. Isot. dan Radiasi*, vol. 11, no. 2, pp. 153–159, 2015, doi: 10.17146/jair.2015.11.2.2798.

[2] M. G. Saragih, E. D. Surya, and M. B, *Pariwisata Super Prioritas Danau Toba*. Medan: CV. Andalan Bintang Ghonim, 2021.

[3] D. J. P. P. dan K. Lingkungan, "Laporan Kinerja: Direktorat Pengendalian Pencemaran Air," Kementerian Lingkungan Hidup dan Kehutanan, Jakarta, 2022.

[4] A. Suwardono, F. E. Prahesti, E. M. Indrawati, and M. A. J. Ashofa, "IoT Based Catfish Farm Monitoring with ESP32 Microcontroller and DS18B20 Sensor," *JST (Jurnal Sains dan Teknol.*, vol. 13, no. 3, pp. 508–516, 2024, doi: 10.23887/jstundiksha.v13i3.85996.

[5] W. S. L. Nasution and P. Nusa, "Implementation of the Internet of Things (IoT) for Remote Light Control Using NodeMCU ESP8266 and Thingspeak Via Website-Based Internet," *J. Comput. Sci. Technol.*, vol. 3, no. 1, pp. 33–39, 2023, doi: 10.54840/jcstech.v3i1.100.

[6] S. Gunawan, A. H. Anshor, and Amali, "Sistem Monitoring dan Kontrol Taman Pintar Berbasis IoT (Internet of Things) dengan NodeMCU ESP8266," *Bull. Comput. Sci. Res.*, vol. 3, no. 4, pp. 283–288, 2023, doi: 10.47065/bulletincsr.v3i4.270.

[7] A. Winarno, Widodo, and M. K. Nurcahyo, "Smart Home Remote Control System Prototype Using Internet of Things (IoT) Based ESP8266 Microcontroller," *BEST J. Appl. Electr. Sci. Technol.*, vol. 5, no. 2, pp. 60–64, 2023, doi: 10.36456/best.vol5.no2.8063.

[8] I. H. Tambunan, B. A. Sihotang, D. Silaen, F. Michael, and K. G. Sitanggang, "Data Communication Between Wheeled Soccer Robots and Referee Box Using ESP8266 and Raspberry Pi 4B," *2023 29th Int. Conf. Telecommun.*, vol. 29th, pp. 1–5, 2023, doi: 10.1109/ICT60153.2023.10374059.

[9] E. Margirahayu, Junaidi, G. A. Pauzi, and S. W. Suciyati, "DC Motor Speed Control System with PWM (Pulse Width Modulation) Technique Based on Arduino For Centrifugation Equipment Application," *J. Energy, Mater. Instrum. Technol.*, vol. 3, no. 3, pp. 99–104, 2022, doi: 10.23960/jemit.v3i3.105.

[10] Pradeep, A. Ratnam, N. Saritha, I. G. Naidu, and G. Swapna, "Pulse Width Modulation Based DC Motor Closed Loop Speed Controller," *Tuijin Jishu/Journal Propuls. Technol.*, vol. 45, no. 02, pp. 4033–4038, 2024, doi: 10.52783/tjjpt.v45.i02.6587.

[11] B. Satria, H. Alam, M. E. Dalimunthe, M. Iqbal, and B. S., "Development of a DC Motor Speed Regulation System Using PWM for Two-Wheeled Electric Vehicles," *Pros. Semin. Nas. Dan Int. Fak. Tek. Dan Ilmu Komput. Univ. Dharmawangsa*, vol. 1, pp. 470–477, 2024.

[12] F. B. Setiawan, Y. Y. C. Wibowo, L. H. Pratomo, and S. Riyadi, "Perancangan Automated Guided Vehicle Menggunakan Penggerak Motor DC dan Motor Servo Berbasis Raspberry Pi 4," *J. Rekayasa Elektr.*, vol. 18, no. 2, pp. 94–101, 2022, doi: 10.17529/jre.v18i2.25863.

[13] A. E. Ramadhan, Djuniadi, and E. Apriaskar, "Sistem Pengaturan Pulse Width Modulation Motor Pada Robot Pembawa Makanan atau Minuman Menggunakan Joystick," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 7, no. 2, pp. 134–143, 2021, doi: 10.15575/telka.v7n2.134-143.

[14] A. M. Ali, C. Shen, and H. A. Hashim, "A Linear MPC with Control Barrier Functions for Differential Drive Robots," *IET Control Theory Appl.*, vol. 18, no. 18, pp. 2693–2704, 2024, doi: 10.1049/cth2.12709.

[15] Q. I. Fatimah, R. Marselino, and Asnil, "Web-Based DC Motor Speed Design and Control," *Motiv. J. Mech. Electr. Ind. Eng.*, vol. 3, no. 3, pp. 101–112, 2021, doi: 10.46574/motivection.v3i3.99.

[16] Yogiswara, H. Y. Riskiawan, S. Anwar, and R. Ardiansyah, "Differential-Drive Wheeled Robot Controller using Pulse- Width modulation in disinfectant sprayer robot," *3rd Int. Conf. Food Agric.*, vol. 3, no. 1, pp. 151–157, 2021.

[17] A. R. Hambley, *Electrical Engineering: Principles & Applications*. New Jersey: Pearson, 2017.

[18] E. Hughes, *Electrical and Electronic Technology*. New Jersey: Prentice Hall, 2012.

[19] M. H. Rashid, N. Kumar, and A. R. Kulkarni, *Power Electronics: Devices, Circuits, and Applications: International Edition*, Fourth Edi. Harlow: Pearson, 2014.

[20] M. H. Rashid, *Power Electronics Handbook*. San Diego: Academic Press, 2001.

[21] A. R. Rey and R. A. Fatekha, "Analysis of Battery Energy Consumption in the Barelang V Base Robot with 4 Brushed DC Motors," in *Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024)*, Atlantis Press, 2024, pp. 5–20. doi: 10.2991/978-94-6463-620-8_2.