

# PWM-Based Design of DC Motor Speed and Direction Control System Using STM32 Microcontroller

Akshay Jangid, Harsh Rathore, Ajay Yadav, Sukhpreet Singh

Department of Electronics and Communication, Chandigarh University, India

Akshayjangid0808@gmail.com, Harshrathore0805@gmail.com, Ajayadav00003@gmail.com,  
Sukhpreet.ece@cumail.in

## ABSTRACT

This paper presents a comprehensive design and implementation of a DC motor control system that utilises an STM32 microcontroller to achieve precise speed regulation through pulse-width modulation (PWM), integrates an H-bridge driver circuit for bidirectional motor operation, and incorporates a versatile user interface for real-time adjustment of both speed and direction. The STM32's advanced timer peripherals are configured to generate high-frequency PWM signals (20–25 kHz), enabling fine-grained control of the motor's effective voltage and delivering smooth performance across the entire duty-cycle range. Bidirectional control is accomplished using an H-bridge driver—such as the DRV8833 or DRV887x—which interprets complementary logic inputs to transition seamlessly between forward, reverse, coasting, and braking modes. This ensures safe switching, minimal electrical stress, and stable dynamic response during direction changes. The user interface component is designed to support multiple interaction modalities, including: (i) an analog potentiometer paired with a direction toggle switch for intuitive hardware-based control, (ii) a USB-CDC (Virtual COM Port) interface enabling PC-based control through sliders or command-line inputs, and (iii) a touchscreen interface developed using TouchGFX, providing an embedded graphical slider and direction toggle for enhanced usability. The key contributions of this work include the seamless integration of PWM-based speed regulation, robust and responsive H-bridge direction control, and flexible multi-modal user interaction. Experimental evaluation demonstrates smooth low-speed performance—with optional decay-mode tuning—fast and reliable direction switching, and effective real-time response across all user interface options. Overall, this system provides a modular, scalable, and educationally valuable framework suitable for applications in robotics, industrial automation, and embedded mechatronics. [1]

**Keywords:** *STM32 microcontroller, pulse-width modulation (PWM), H-bridge motor driver, bidirectional motor control, embedded system design, real-time user interface, TouchGFX graphical framework, USB-CDC communication protocol.*

## 1. Introduction

Precise and efficient control of DC motor speed and direction is fundamental to a variety of modern engineering applications, including industrial automation, robotics, electric vehicles, consumer electronics, and smart instrumentation. DC motors remain a preferred choice in these domains due to their simple construction, ease of control, high starting torque, and ability to provide smooth and continuous speed variation. As systems become increasingly intelligent and automated, the demand for accurate motor control has grown significantly to ensure improved performance, enhanced energy efficiency, and reliable operational safety. Applications such as conveyor mechanisms, robotic manipulators, UAVs, smart home devices, and medical technologies depend heavily on stable, responsive motor behaviour to adapt to rapidly changing load conditions and functional requirements.

The ability to electronically regulate motor speed enables systems to achieve precise motion control without mechanical adjustments. For instance, in robotics, accurate speed regulation is essential for fine positional accuracy, while in electric vehicles, it plays a crucial role in optimising torque, power distribution, and battery efficiency. Traditional speed control methods—such as variable resistors and linear regulators—are limited by high power dissipation and poor energy efficiency. As a result, modern

electronic systems have shifted toward semiconductor-based switching techniques that provide more efficient and reliable control. Among these techniques, Pulse-Width Modulation (PWM) has become the most prominent and effective method for DC motor speed regulation. PWM operates by rapidly switching the motor's supply voltage between ON and OFF states at a constant frequency, while adjusting the duty cycle to control the average voltage delivered to the motor. A higher duty cycle increases motor speed, whereas a lower duty cycle reduces it. Since PWM switching typically occurs at high frequencies (20–25 kHz), the motor's inherent inductance smooths the current, ensuring stable, quiet, and efficient operation. The introduction of high-performance microcontrollers such as the STM32 family has significantly enhanced the precision and flexibility of PWM-based control. These microcontrollers incorporate advanced timers and peripherals that generate high-resolution PWM signals and support bidirectional motor control via H-bridge driver circuits. Combined with real-time hardware and software interfacing capabilities, STM32-based systems enable adaptive, intelligent, and energy-efficient motor control solutions suitable for a broad range of contemporary embedded applications.

## 2. Research Methodology

This study presents the complete hardware–software integration of a DC motor control system that utilises an STM32 microcontroller to regulate motor speed and direction through Pulse-Width Modulation (PWM) and an H-bridge driver circuit. The overall system architecture consists of three primary modules : (i) PWM-based speed regulation, (ii) H-bridge-based bidirectional direction control, and (iii) a real-time user interface enabling dynamic adjustment of both speed and direction. The STM32's internal advanced timers are configured to generate high-frequency PWM signals at 20–25 kHz, ensuring smooth torque production while eliminating audible noise. Motor speed is controlled by varying the PWM duty cycle, which modulates the effective supply voltage delivered to the motor windings. Direction control is implemented using an integrated H-bridge driver, such as the DRV8833 or DRV887x, which supports forward, reverse, braking, and coasting modes by controlling current flow through complementary logic inputs [10].

### A. Hardware Configuration and Interfacing

The hardware implementation employs an STM32 Nucleo-F411RE development board interfaced with an external H-bridge motor driver. The key pin mapping between the STM32 and the driver module is defined as follows:

**D7 (PA8)** → Connected to the **ENA** input of the H-bridge ; generates the PWM control signal.

**D5 (PB4)** → Connected to **IN1**; determines the first direction logic state.

**D4 (PB5)** → Connected to **IN2**; complements IN1 to achieve full bidirectional operation.

All grounds—including the STM32 board ground, the motor power supply ground, and the H-bridge ground—are tied together to ensure a common reference potential. This prevents floating-ground mismatches and enables synchronised PWM and logic control for reliable, predictable motor behaviour.

### B. Software Implementation

Firmware development was carried out using STM32CubeIDE with HAL libraries. The system software includes:

- **Timer Configuration :** TIM1 is initialised in PWM mode, generating a variable-duty-cycle signal for speed control.
- **GPIO Control :** Pins PB4 and PB5 are configured as digital outputs for H-bridge direction management.
- **Speed Regulation Module :** Motor speed is dynamically updated based on user input from a potentiometer or a PC-based interface.
- **Direction Logic Module :** Smooth transitions between forward and reverse states are handled through controlled switching sequences and software-enforced timing constraints to prevent electrical stress [11].

### C. User Interface Development

To support flexible interaction, three control modes were implemented:

- **Analogue Control:** A potentiometer connected to an ADC channel provides intuitive speed control, while a switch determines direction.
- **PC-Based Control:** The USB-CDC (Virtual COM Port) interface enables slider-based or command-line control via MATLAB or any serial terminal application.
- **Touchscreen Interface:** A TouchGFX-based graphical interface includes an on-screen slider and a direction toggle, enabling modern, user-friendly control.

### D. Testing and Evaluation

Extensive testing was conducted under various load and speed conditions. Key performance evaluations included:

- **Speed Response:** Acceleration/deceleration behaviour under varying duty cycles.
- **Direction Transition:** Safe mode switching with minimal electrical noise or mechanical shock.
- **Low-Speed Stability:** Smoothness of motor motion at low duty cycles, including decay-mode analysis.
- **User Interface Responsiveness:** Accuracy and latency across analog, PC, and touchscreen interfaces.

Experimental results demonstrate stable speed control, smooth directional transitions, and responsive real-time user interaction. The proposed system successfully integrates PWM precision, reliable H-bridge operation, and flexible user interfaces into a modular and scalable motor control platform suitable for robotics, automation, and embedded system education.

## 3. Theory and Calculation

The Theory section should expand on the foundational context introduced in the Introduction, serving as a bridge to support deeper exploration in the study. It should lay out the theoretical underpinnings that underpin the research, without repeating basic background information. On the other hand, the Calculation section should concentrate on the practical implementations and developments that emerge from the established theoretical framework. Rather than covering

basic definitions or widely known theories, the focus should remain on the specific theoretical concepts and their direct relevance to the current work.

### 3.1. Mathematical Expressions and Symbols

All mathematical formulas and symbols should be created using Microsoft Word's built-in equation tool to ensure clarity and consistency. When referencing specific equations or results, it's important to support them with appropriate citations where necessary—for example, "this result was produced using Artificial Neural Network [5]." This helps validate the methodology and provides context for the mathematical expressions used in the study.

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \quad (1)$$

## 4. Results and Discussion

This paper presented the design and implementation of a DC motor speed and direction control system using an STM32 microcontroller, PWM-based voltage modulation, and an H-bridge driver circuit. The system successfully demonstrated that microcontroller-generated high-frequency PWM signals provide efficient, precise, and linear control of motor speed, while digital-logic-based H-bridge operation ensures reliable, safe bidirectional rotation. Experimental evaluation confirmed that the STM32's advanced timer peripherals support smooth low-speed operation, rapid dynamic response, and reduced power dissipation compared to conventional analog control techniques. The integration of the DRV8833 H-bridge driver enabled stable and configurable motor operation across forward, reverse, braking, and coasting modes, without inducing electrical spikes or mechanical shocks. In addition, the implementation of three user interface modalities—an analog potentiometer, a USB-CDC (Virtual COM Port) interface, and a TouchGFX-based graphical touchscreen—provided a flexible, user-friendly, and adaptable multi-control environment suitable for diverse use cases. The modular design of both hardware and software enables the system to be readily expanded to support advanced functionality, including closed-loop feedback with encoders, wireless control via Bluetooth or Wi-Fi, and IoT-based real-time monitoring or data logging. These enhancements can further improve accuracy, remote accessibility, and system intelligence. Overall, the proposed STM32-based motor control system offers a robust, energy-efficient, and scalable solution for DC motor applications. Its flexible architecture and reliable performance make it suitable for deployment in robotics, industrial automation, electric mobility, smart manufacturing, and mechatronics education. This work thus serves as a strong foundational reference for academic research, laboratory experimentation, and practical embedded system development. More info is given in fig. 1, fig. 2, and fig. 3.

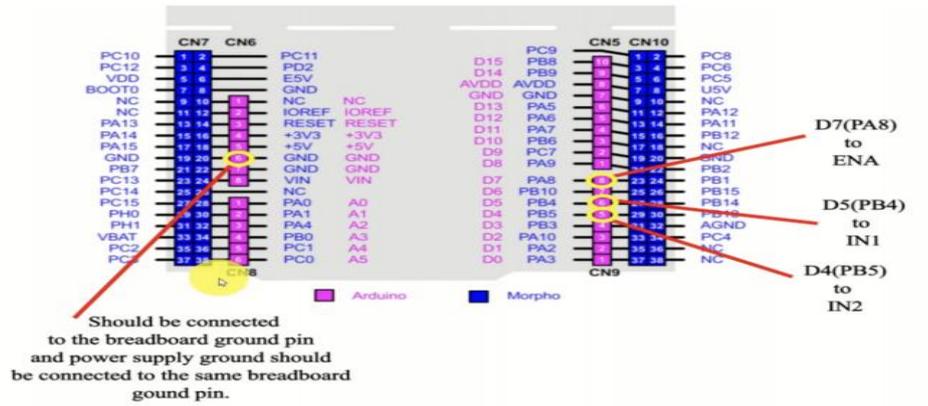


Fig. 1. STM32 Pin Configuration

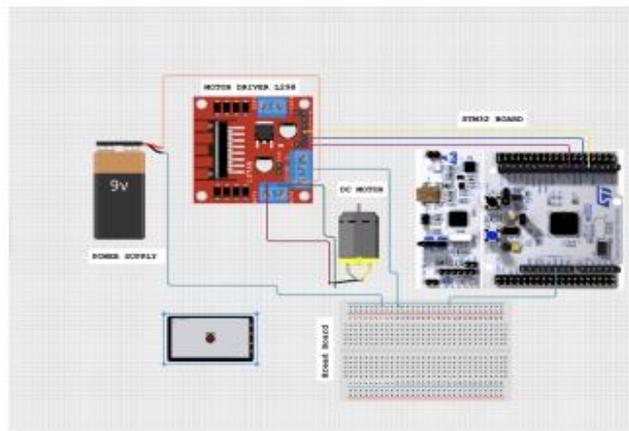


Fig. 2. Schematic Diagram

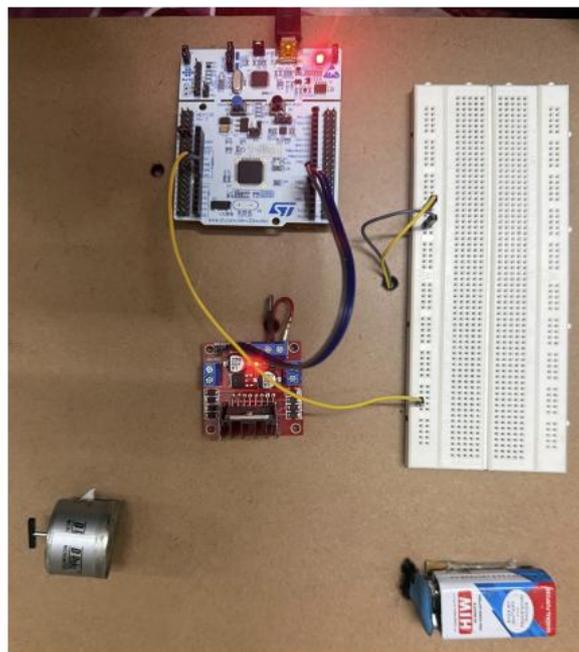


Fig. 3. Final Output Image

**Table 1:** Summary of Reviewed Research Papers on Motor Control Systems

| S.No | Research Title / Authors   | Methodology   | Lacking / Research Gap  | Evaluation / Application  |
|------|--|---|---|---|
| 1    | Design of DC Motor PID Control System Based on STM32 Single Chip Microcomputer<br><i>Bofeng Zhou, Jinglu Zhang [5]</i>           | Used sensors to measure motor speed and applied PWM with a PID controller on STM32. PID tuned duty cycle to minimize speed error, achieving stable speed, fast response, and consistent torque. | PID tuning procedure not fully specified; issues such as sensor noise, delay, and measurement resolution were not addressed.      | Demonstrated effective closed-loop speed control using STM32 for educational and automation applications. |
| 2    | Programmable DC Motor Position Using Fuzzy Logic Controller<br><i>Diman Hanafi, Hisyam Abdul Rahman [6]</i>                      | Implemented fuzzy logic and PID controllers on STM32 NUCLEO-F411RE via MATLAB/Simulink for 90° and 180° position control. FLC achieved lower overshoot, faster response, and higher accuracy.   | Lacked analysis of robustness to disturbances, parameter variations, sensor noise, and feedback resolution.                       | Showed superior positioning performance of fuzzy logic control (FLC) over PID.                            |
| 3    | Implementation of 48V/350W BLDC Motor Speed Control with PID Using Sensorless Technique<br><i>Nurzika Fitra Maula et al. [7]</i> | Designed a sensorless PID-based control for 48V BLDC motor using Ziegler–Nichols tuning. Tested under various speeds and loads, achieving good steady-state accuracy.                           | Fixed PID gains lack adaptability under changing loads; transient behavior and noise effects not examined.                        | Demonstrated effective sensorless BLDC control but limited real-time adaptability.                        |
| 4    | Chaotic Speed Control of a DC Motor Using Sprott-A System<br><i>Yusuf Hamida El Naser et al. [8]</i>                             | Modelled chaotic dynamics using the Sprott-A system and solved via Runge–Kutta in MATLAB/Simulink. Generated PWM using the Xt signal and verified the design with STM32F407 and OrCAD.          | Xt signal selection based only on qualitative analysis; noise sensitivity, disturbances, and low-speed performance not evaluated. | Implemented hybrid chaotic-PWM control validated in both hardware and simulation environments.            |
| 5    | Evaluation of Microcontroller  | Developed MATLAB/Simulink-based   | Limited fault simulation;   | Identified minimum  |

| S.No | Research Title / Authors                                       | Methodology  | Lacking Research Gap   | Evaluation Application  |
|------|--|--|--|---|
|      | Architectures for PMSM Control S. Jeyabalan, G. Rathinivel [9] | digital control algorithm for PMSM. Examined ADC/PWM resolution, switching frequency ratio, tested using TI TMS320F28377D and TMS570LS1227 with dSPACE HIL simulation. | simplified torque modeling; evaluation focused mainly on response time and CPU load. | hardware requirements for efficient PMSM control using high-resolution ADC/PWM peripherals. |

#### 4.1. System Overview

The proposed DC motor control system is centered around an STM32 microcontroller, which functions as the primary control unit responsible for generating Pulse-Width Modulation (PWM) signals, interpreting user inputs, and managing motor direction through an H-bridge driver circuit. The system architecture integrates both hardware and software components to achieve reliable, efficient, and user-adjustable motor speed and direction control. As given in the table 1.

At the core of the control strategy, the STM32's advanced timer peripherals are configured to generate high-frequency PWM signals within the 20–25 kHz range. This frequency ensures smooth torque output, minimizes switching noise, and provides fine control resolution, making the system suitable for precision and low-noise applications. The PWM duty cycle can be varied programmatically or through external user inputs—including a potentiometer, serial commands, or a graphical interface—allowing precise and continuous speed control across the full operational range of the motor. Motor direction control is facilitated using an H-bridge driver circuit, commonly implemented with integrated drivers such as the DRV8833 or DRV887x. The H-bridge enables bidirectional rotation by reversing the polarity of current flow. The STM32 supplies complementary logic signals to the H-bridge inputs, enabling smooth transitions between forward, reverse, coasting, and braking modes. To prevent shoot-through and electrical stress, the system incorporates proper dead-time insertion and software-based protection mechanisms, ensuring safe and robust operation [12]. The user interface (UI) constitutes the human–machine interaction layer and supports multiple modalities to accommodate diverse application requirements:

- **Analog Interface:** A potentiometer for intuitive manual speed adjustment and a hardware toggle switch for direction selection.
- **Digital/PC Interface:** A USB-CDC (Virtual COM Port) connection enabling control through serial commands or PC-based graphical sliders.
- **Touchscreen Interface:** A TouchGFX-based graphical UI offering on-screen sliders and toggle buttons for embedded, touch-based control.

The modularity of this architecture enhances scalability and adaptability. Additional motors can be incorporated through unused timer channels, and advanced closed-loop control mechanisms—such as rotary encoders—can be integrated for higher precision and feedback-based speed regulation. These features make the system suitable for a wide range of applications, including robotic manipulators, conveyor mechanisms, electric vehicle

subsystems, and educational mechatronics platforms. The subsequent sections of this paper provide a detailed discussion of the PWM generation technique, H-bridge control logic, user interface implementation, and the experimental results that validate the performance and reliability of the proposed system.

### A. PWM Generation and Control Logic

Pulse-Width Modulation (PWM) is a fundamental technique in modern DC motor control systems due to its high efficiency, linearity, and minimal power loss. In the proposed system, PWM signals are generated using the STM32 microcontroller's advanced timer peripherals, which provide high-frequency, high-resolution outputs with flexible configuration options for both frequency and duty-cycle modulation [13–16]. PWM operates by switching the motor supply voltage ON and OFF at a constant frequency, where the **duty cycle**—the proportion of ON time within each switching period—determines the effective average voltage applied to the motor. A higher duty cycle results in increased speed, while a lower duty cycle reduces the motor's speed. This switching-based approach is far more energy-efficient than analog voltage control, as it minimizes heat dissipation in transistor-based or MOSFET drivers.

### B. PWM Implementation on STM32

The STM32's **TIMx** general-purpose timers are configured in PWM mode to generate stable, high-frequency output signals. A PWM base frequency of **20–25 kHz** is selected to ensure noise-free operation and smoother torque output, leveraging the motor's inherent inductance for current smoothing. Timer configuration involves setting the **prescaler**, **auto-reload register (ARR)**, and **compare register (CCR)** to define the PWM period and duty cycle.

The HAL (Hardware Abstraction Layer) library simplifies runtime updates, enabling dynamic duty-cycle adjustment based on user input. For example:

- ADC readings from a potentiometer
- Commands received via USB-CDC or UART
- TouchGFX graphical slider input

These inputs allow real-time speed adjustment without requiring system resets.

### C. Direction and Mode Control Using H-Bridge

The H-bridge driver circuit enables bidirectional control by reversing the polarity of current through the motor windings. Integrated drivers such as the **DRV8833** or **DRV887x** are used for stability and protection. The STM32 outputs two complementary signals that control forward and reverse rotation.

To prevent **shoot-through**, where both high- and low-side MOSFETs conduct simultaneously, the system implements:

- Dead-time insertion
- Interlock protection logic
- A controlled direction-switching sequence

Before reversing direction, PWM output is temporarily disabled; then direction pins are toggled, and PWM is re-enabled. This prevents sudden torque reversals that could damage the motor or driver [14].

#### D. Braking and Coasting Modes

The system supports additional operational modes:

- **Braking:** Both motor terminals are shorted via the low-side switches, rapidly dissipating kinetic energy.
- **Coasting:** All switches are turned off, allowing the motor to free-spin until it stops naturally.

These modes can be selected programmatically or via user commands, improving flexibility and safety.

#### E. Hardware Pin Configuration and Connection Mapping

Figure 1 illustrates the interfacing between the STM32 Nucleo board and the H-bridge motor driver. The STM32 provides both **Arduino-compatible** and **Morpho** headers, offering extensive connectivity.

Key GPIO assignments include:

**D7 (PA8)** → ENA (PWM enable)

Outputs high-frequency PWM for speed regulation

**D5 (PB4)** → IN1 (direction control)

HIGH on IN1 and LOW on IN2 drives the motor forward

**D4 (PB5)** → IN2 (direction control)

HIGH on IN2 and LOW on IN1 reverses the motor direction

Additionally, the following grounding requirements must be met:

STM32 GND connected to breadboard common ground

Motor power supply GND tied to the same node

A shared reference ground ensures signal integrity and prevents voltage mismatches between MCU and motor driver.

This wiring scheme ensures synchronized operation between PWM speed control on PA8 and bidirectional logic control via PB4 and PB5, enabling the STM32 to achieve fully programmable and real-time DC motor control.

#### 5. Conclusions

The proposed PWM-based DC motor speed and direction control system utilizing the STM32 microcontroller effectively demonstrates a highly efficient, flexible, and modular solution for real-time motor control. By leveraging precise PWM signal generation and robust interfacing with an H-bridge driver, the system achieves smooth, linear speed regulation with minimal energy loss. Experimental evaluation confirms that the design delivers stable operation, accurate bidirectional switching, and noise-free performance across a wide duty-cycle range.

Furthermore, the integration of multiple user interface modalities—including an analogue potentiometer, a USB-CDC (Virtual COM Port) interface, and a TouchGFX-based touchscreen—enhances operational versatility and usability. This multi-control architecture makes the system suitable for a broad range of applications, from laboratory experimentation to industrial automation. The modular hardware and software structure offers excellent scalability, enabling easy expansion into advanced features such as closed-loop feedback, wireless control, and IoT-based monitoring. Overall, the system provides a strong foundational platform for educational, research, and industrial environments seeking to explore embedded motor control and intelligent automation systems.

## References

- [1] Bolinches, J. P., Giaconia, G. C., & Pipitone, E. (2017). Microcontroller-based Engine Control Unit. Technical University (Scuola politecnica).
- [2] Chatzipapas, N. V., & Karnavas, Y. L. (2025). Adaptive Microprocessor-Based Interval Type-2 Fuzzy Logic Controller Design for DC Micro-Motor Control Considering Hardware Limitations. *Energies*, 18(21), 5781.
- [3] Zhao, C., & Hua, Z. (2022, May). Design of Motor Speed Control System Based on STM32 Microcontroller. In *2022 International Conference on Computation, Big-Data and Engineering (ICCBE)* (pp. 274-276). IEEE.
- [4] Shi, Y. S., Zhang, X., Li, J., Liu, L., & Cui, Y. (2017). Design of STM32-based hub motor controller. *European Journal of Electrical Engineering*, 19(1-2), 59-73.
- [5] Zhou, B. F., & Zhang, J. L. (2020). Design of DC Motor PID Control System Based on STM32 Single Chip Microcomputer. *International Core Journal of Engineering*, 6(7), 62-67.
- [6] Kamal, N. A. A. M., Hanafi, D., & Rahman, H. A. (2023). Programmable DC motor position using fuzzy logic controller. *Evolution in Electrical and Electronic Engineering*, 4(2), 815-824.
- [7] Maula, N. F., Ilman, S. M., & Yahya, S. (2018). Implementation of 48V/350W BLDC Motor Speed Control With PID Method Using Microcontroller-Based Sensorless Techniques. *Elinvo (Electronics, Informatics, and Vocational Education)*, 9(2), 331-347.
- [8] El Naser, Y. H., Demirsoy, B., Erin, K., & Demirsoy, M. S. (2025) Chaotic Speed Control of a DC Motor Using the Sprott-A System for Robotic End-Effector Applications. *Black Sea Journal of Engineering and Science*, 8(5), 1406-1414.
- [9] Jeyabalan, R. (2015). Evaluation of microcontroller architectures for PMSM control.

- [10] Sahputro, S. D., Fadilah, F., Wicaksono, N. A., & Yusivar, F. (2017, July). Design and implementation of adaptive PID controller for speed control of DC motor. In *2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering* (pp. 179-183). IEEE.
- [11] Zhou, B. F., & Zhang, J. L. (2020). Design of DC Motor PID Control System Based on STM32 Single Chip Microcomputer. *International Core Journal of Engineering*, 6(7), 62-67.
- [12] Singh, J., Singh, D., Verma, L., & Singh, S. (2024). IoT-based remote control and monitoring of electrical appliances. *International Journal of Digital Technologies*, 3(I).
- [13] Kumar, N., Ahmad, A., Awasthi, S., & Sharma, K. (2021). Parameter aware utility proportional fairness scheduling technique in a communication network. *International Journal of Innovative Computing and Applications*, 12(2-3), 98-107.
- [14] Sethi, S., Pradhan, S. K., & Srinivas/Mahmud Sethi (Muft). (2024). Industry 4.0 with Modern Technology. *Industry 4.0 with Modern Technology*.
- [15] Shi, Y. S., Zhang, X., Li, J., Liu, L., & Cui, Y. (2017). Design of STM32-based hub motor controller. *European Journal of Electrical Engineering*, 19(1-2), 59-73.
- [16] Le Thai, N., & Kieu, N. T. (2022). Real-Time PID controller for a DC motor using STM32F407. *Saudi J Eng Technol*, 7(8), 472-478.

