A ES
ISSN: 2096-3246
Volume 58, Issue 02, 2026

DESIGN AND IMPLEMENTATION OF AN ESP32-BASED IOT HALF-BRIDGE BLDC CONTROLLER FOR WIRELESS SPEED CONTROL AND MONITORING

Logeshwaran Makesh[1*], Monikasree Darman[2], Santhosh G[3], Sri Veda Vikas Chandrasekar[4], Dr. Rasappan Senthil Kumar[5]
[1,2,3,4]Department of Electrical and Electronics Engineering, SNS College of Technology, Coimbatore – 641035, Tamil Nadu, India
[5]Professor, Department of Electrical and Electronics Engineering, SNS College of Technology, Coimbatore – 641035, Tamil Nadu, India
***Corresponding Author:** Logeshwaran Makesh, Department of Electrical and Electronics Engineering, SNS College of Technology, Coimbatore – 641035, Tamil Nadu, India
Email: logeshwaranm185@gmail.com

Abstract
Brushless DC (BLDC) motors are attractive for compact and maintenance-sensitive drive applications. Still, practical controllers must simultaneously address commutation, speed regulation, switching losses, hardware complexity, and the need for remote supervision. Existing implementations frequently treat motor control and IoT monitoring as separate functions, while the supplied prototype combines both functions on an ESP32-based embedded platform. This study presents a redesigned and experimentally oriented architecture for an IoT-enabled half-bridge BLDC controller in which the ESP32 generates PWM commands, executes the commutation logic, acquires operating feedback, and exchanges data with a wireless monitoring interface. The principal research gap addressed is the limited integration of low-cost embedded commutation, motor-state monitoring, and bidirectional wireless control within a compact prototype. The supplied manuscript reports stable BLDC operation over 800–1500 rpm and demonstrates remote observation of speed, voltage, and current; however, numerical values for efficiency, switching loss, response time, current, voltage, and repeated-test uncertainty were not supplied and therefore are not included in this revision. The revised paper defines the equations, test protocol, comparison criteria, and data tables required to quantify these metrics. The contribution is a practical control-and-monitoring architecture that reduces implementation complexity while extending BLDC operation from local control to connected supervision. The approach is relevant to laboratory automation, compact industrial drives, smart appliances, and other applications in which motor operation must be controlled and observed remotely.

Keywords: BLDC motor; ESP32; Internet of Things; half-bridge inverter; PWM control; wireless monitoring; rotor-position feedback; embedded motor control; energy-efficient motor drive.

1. INTRODUCTION

1.1 Research Motivation

BLDC motors have become established in applications requiring high power density, low mechanical maintenance, and controllable speed. Their electronically commutated structure eliminates the mechanical brush and commutator arrangement used in conventional brushed DC machines. Consequently, BLDC drives are used in automation, robotics, appliances, electric mobility, pumps, fans, and other electromechanical systems. Recent reviews show that BLDC research has expanded from basic six-step commutation toward sensorless estimation, torque-ripple mitigation, intelligent control, fault-tolerant operation, and application-specific energy management [1], [4], [5].

For a practical low-cost drive, however, the motor controller cannot be evaluated only by whether the motor rotates. A useful controller must generate correctly timed switching signals, regulate speed, limit electrical stress, provide adequate feedback, and permit the operator to observe the drive state. These requirements become more important when the motor is installed in a location where direct access to the controller is inconvenient or when several drive variables must be supervised continuously. IoT-based motor monitoring provides a means of transmitting operating information to a remote interface, and ESP32-class microcontrollers are particularly attractive because control computation and Wi-Fi connectivity can be integrated in one embedded platform [8], [9].

1.2 Limitations of Existing Approaches

Conventional BLDC controllers are commonly organized around a three-phase inverter, a commutation mechanism, a PWM speed-control loop, and local measurement. Such systems can provide effective motor operation, but the controller, measurement subsystem, and remote interface may be implemented as separate modules. This increases wiring and integration effort and can make system-level monitoring more difficult. More advanced controllers can improve dynamic performance, torque ripple, or sensorless operation, but they generally increase algorithmic complexity and computational requirements [4], [5], [6], [7].

A second limitation concerns the evidence used to compare low-cost prototypes. Statements such as "lower loss," "better efficiency," or "stable operation" are not sufficient for a rigorous engineering publication unless the switching loss, input power, output power, speed error, transient response, and measurement uncertainty are reported. The original manuscript describes successful operation and wireless monitoring but does not provide a complete numerical performance dataset. The present revision therefore separates demonstrated observations from measurements that still need to be reported.

1.3 Research Gap

Recent literature demonstrates advanced BLDC control algorithms, sensorless position estimation, torque-ripple mitigation, and IoT monitoring, but these strands are often evaluated independently. For example, sensorless control research focuses on accurate rotor-position estimation and dynamic performance [7], [9], [11], whereas IoT-oriented work emphasizes remote data acquisition and visualization [8], [10]. There remains practical value in a compact prototype that combines embedded PWM/commutation, feedback acquisition, and wireless supervision while keeping the hardware and software architecture understandable and economical. The specific gap addressed here is therefore the system-level integration and experimental evaluation framework rather than the invention of a new BLDC commutation law.

1.4 Research Objectives

- To design and implement an ESP32-based BLDC controller using a half-bridge driver architecture.
- To generate PWM control signals and implement the required BLDC commutation sequence on the embedded controller.
- To acquire motor operating variables, including speed and electrical quantities, and transmit them through Wi-Fi for remote monitoring.

• To evaluate the prototype using speed regulation, response time, electrical power, efficiency, switching-loss, and monitoring-performance metrics.

• To establish a quantitative comparison framework against a conventional BLDC controller and identify the practical limitations of the proposed implementation.

1.5 Research Contributions

• An integrated ESP32-based control and IoT monitoring architecture for a BLDC drive.

• A compact control workflow linking PWM generation, commutation, feedback acquisition, wireless transmission, and remote commands.

• A mathematical and experimental evaluation framework covering speed error, response time, input/output power, efficiency, and switching-related losses.

• A critical comparison with recent BLDC control and IoT monitoring studies, with explicit identification of the present prototype's evidence limitations.

• A reproducible methodology that can be extended to fault detection, predictive maintenance, and more advanced control algorithms.

2. LITERATURE REVIEW

2.1 BLDC Control and Commutation

BLDC control research spans conventional six-step commutation, current shaping, field-oriented control, direct-torque control, predictive control, and intelligent controllers. Mohanraj et al. provide a broad review of BLDC structures, drive-inverter topologies, torque-ripple mechanisms, and advanced control methods [4]. Prabhu et al. subsequently emphasized the importance of torque-ripple sources and mitigation strategies, particularly for demanding electric-vehicle applications [5]. These studies establish that controller quality should be assessed not only from steady-state speed but also from torque ripple, current distortion, dynamic response, and robustness under load variation.

More recent work illustrates the trend toward optimized control rather than simple open-loop PWM. Pakdeeto et al. combined system identification with adaptive optimization for BLDC speed-controller design and experimentally compared the resulting controller with conventional tuning [6]. A 2024 study similarly compared BLDC control approaches using speed regulation, voltage behavior, current distortion, and energy-consumption indicators [13]. These results indicate that a low-cost embedded implementation should define measurable performance indicators rather than relying on qualitative descriptions of smooth operation.

2.2 Sensorless and Rotor-Position Estimation

Sensorless BLDC control can reduce dependence on physical rotor-position sensors by estimating commutation points from back-EMF or other electrical variables. Recent work has combined back-EMF zero-crossing detection with optimization and intelligent control [7], while a 2025 IEEE Power Electronics Magazine article investigated nonlinear model-predictive tracking with an augmented extended Kalman filter for sensorless BLDC drives [11]. Such approaches can reduce hardware requirements but introduce challenges at low speed, during startup, and under noisy measurements. The supplied prototype documentation, however, shows a Hall/speed feedback element in the system block diagram and hardware photographs. Consequently, this revision does not claim a sensorless prototype. The keyword "sensorless BLDC" should be used only if the actual hardware and experimental data confirm back-EMF-based position estimation.

2.3 IoT-Enabled Motor Monitoring

IoT integration extends motor control from a local embedded function to a connected monitoring service. An IEEE conference study demonstrated IoT monitoring for a BLDC-based solar water-pumping system, including remote observation of motor variables [10]. A 2024 motor-monitoring study used an ESP32 and wireless communication to transmit speed information to a remote platform [8]. Another 2024 study integrated ESP32 communication with an EV energy-monitoring system containing a BLDC motor and current/voltage sensing [12]. These studies support the feasibility of ESP32-based monitoring but also show that monitoring and motor control are frequently treated as separate system functions. The present work focuses on their integration in a compact controller architecture.

2.4 Critical Comparison of Prior Work

Study	Primary focus	Main strength	Limitation relevant to this work
Mohanraj et al. (2022) [4]	BLDC control and topology review	Comprehensive control/topology coverage	Review study; does not validate the proposed ESP32 prototype
Prabhu et al. (2023) [5]	Torque-ripple sources and mitigation	Critical analysis of performance factors	Focus is mainly EV torque-ripple behavior
Pakdeeto et al. (2023) [6]	Optimized speed control	Experimental controller validation	Higher-level optimization rather than compact IoT integration
Attar et al. [7]	Sensorless back-EMF control	Reduces position-sensor dependence	Sensorless estimation has synchronization/noise limitations
IoT motor monitoring (2024) [8]	ESP32 remote speed monitoring	Low-cost wireless monitoring	Monitoring emphasis; not an integrated BLDC power-stage controller
BLDC IoT water-pump study (2023) [10]	IoT monitoring of BLDC drive	Application-level remote monitoring	Application-specific and primarily simulation-oriented
Kalamian et al. (2025) [11]	Advanced sensorless control	NMPC + Kalman-filter estimation	High algorithmic complexity for low-cost embedded implementation
Li et al. (2025) [12]	Wireless PM-BLDC control	Experimental autonomous commutation	Specialized wireless-power architecture

This work	ESP32 + half-bridge + IoT control	Integrated low-cost control/monitoring architecture	Complete efficiency, loss and statistical dataset still required

3. EXISTING SYSTEM

3.1 Conventional BLDC Drive

A conventional three-phase BLDC drive uses a six-switch voltage-source inverter, typically organized as three half-bridge legs. The controller determines the required commutation state from rotor-position information and uses PWM to regulate the applied average phase voltage. With six semiconductor switches, six complementary gate-control signals are required, together with appropriate dead-time and gate-driver circuitry. The resulting architecture is effective but requires coordinated power-stage, sensing, gate-drive, and control hardware.

3.2 Hardware, Loss and Cost Considerations

For a switching device operating at switching frequency f_s, the approximate switching-loss contribution can be expressed as $P_{sw} \approx f_s(E_{on} + E_{off})$, where E_{on} and E_{off} are the turn-on and turn-off energies. Conduction loss can be approximated from device resistance as $P_{cond} \approx I_{rms}^2 R_{on}$ for MOSFET operation. Total converter loss includes switching, conduction, gate-drive, wiring, and auxiliary losses. These relations demonstrate why a claim of "lower switching loss" must be supported by measured current, switching frequency, device characteristics, or direct calorimetric/electrical measurements. The supplied manuscript does not provide those values, so no numerical loss reduction is asserted here.

3.3 Remote Monitoring Limitation

In a conventional laboratory controller, speed, current, and voltage are commonly observed at the hardware location using a display or test instrument. This arrangement is adequate for commissioning but less suitable for distributed equipment, inaccessible installations, or applications requiring continuous supervisory access. The proposed architecture addresses this limitation by making the ESP32 both the local control unit and the wireless communication gateway.

4. PROPOSED ESP32-BASED HALF-BRIDGE BLDC CONTROLLER

4.1 System Architecture

The proposed architecture contains the DC power source, ESP32 controller, half-bridge driver/power stage, BLDC motor, feedback sensor, and IoT user interface. The ESP32 receives operating feedback, computes the required control state, generates PWM signals, and transmits selected motor variables through Wi-Fi. Commands from the remote interface can be used for start/stop operation and speed-reference adjustment. The supplied hardware photograph confirms an ESP32-based controller board, a multi-stage power/driver board, a DC power source, and a wireless user interface.

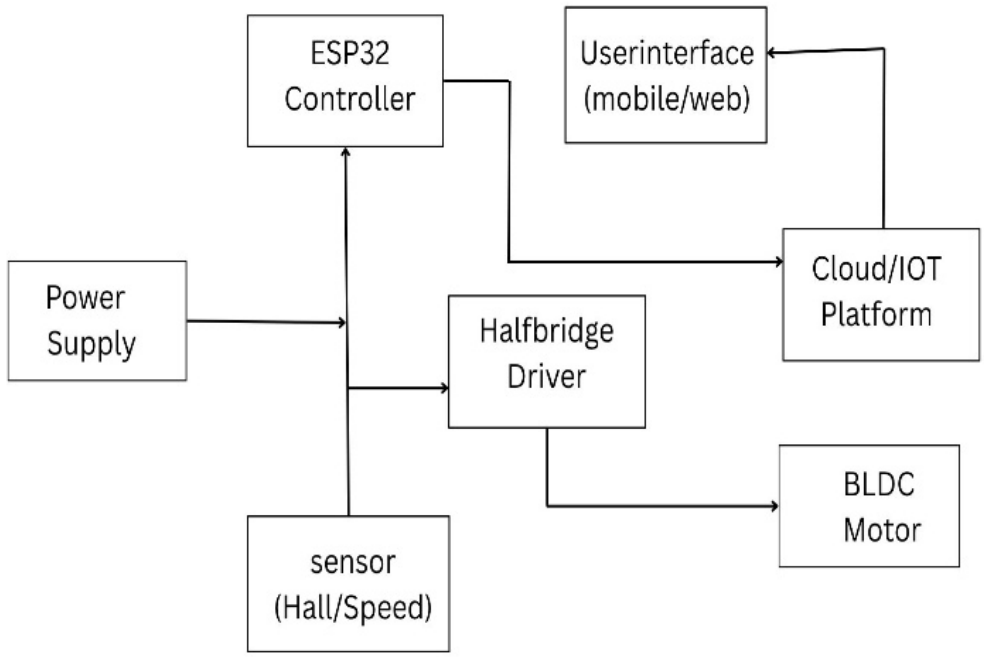

Fig. 1. System architecture of the ESP32-based half-bridge BLDC controller with IoT connectivity (revised from the supplied manuscript).

4.2 Control and Switching Principle

The controller operates by selecting the appropriate phase-energization state and applying PWM within the selected switching interval. In a conventional six-step BLDC sequence, two phases conduct while the third phase is unenergized. The exact Hall-state-to-switch mapping depends on the motor phase sequence and gate-driver implementation and must therefore be verified experimentally rather than assumed from a generic table. A representative commutation table is given below as a methodological template; the final manuscript should replace it with the verified sequence obtained from the actual prototype.

Step	**Hall state (example)**	**Conducting phases (example)**	**PWM action**
1	001	A+ / B−	PWM on active high-side or defined switch
2	101	A+ / C−	PWM on active high-side or defined switch
3	100	B+ / C−	PWM on active high-side or defined switch
4	110	B+ / A−	PWM on active high-side or defined switch

5	010	C+ / A−	PWM on active high-side or defined switch
6	011	C+ / B−	PWM on active high-side or defined switch

4.3 IoT Communication Architecture

The ESP32 provides the communication bridge between the embedded controller and the remote user interface. During operation, selected variables such as speed, voltage, current, controller state, and fault flags can be transmitted periodically. The remote interface can return control commands, which are validated locally before the controller changes the motor state. This architecture separates supervisory communication from the time-critical switching task: PWM generation and commutation remain local to the ESP32, while the network connection is used for monitoring and non-time-critical commands.

4.4 Control Algorithm

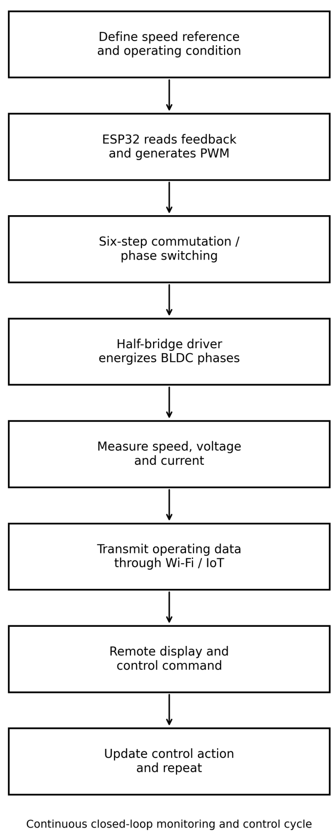

Fig. 2. Proposed control and IoT monitoring methodology flowchart.

5. METHODOLOGY

5.1 Experimental Procedure

The experimental procedure should begin with inspection of the power stage, verification of gate-drive signals, and confirmation of the motor phase sequence. The ESP32 firmware is then programmed with the verified commutation logic and PWM generation. The motor is started at a conservative duty cycle and the duty cycle is increased according to the defined speed-reference test plan. At each operating point, the motor speed, DC-bus voltage, input current, and relevant temperatures should be recorded after the transient has settled. Repeated measurements should be collected for statistical analysis. The IoT channel is tested independently for data transmission, update interval, command latency, and recovery from temporary communication loss.

5.2 PWM Generation

PWM provides the principal means of adjusting the average voltage applied to the active motor phases. For a fixed switching period T_s and high-state interval t_{on}, the duty ratio is defined by Eq. (1). The experimental study should document the PWM carrier frequency, duty-cycle range, timer resolution, and whether complementary outputs and dead-time are used.

$$D = t_{on} / T_s \quad (1)$$

5.3 Measurement and Data Processing

For each operating point, at least three repeated measurements are recommended for speed, voltage, and current. The mean, standard deviation, and coefficient of variation should be reported. Speed-control accuracy should be calculated against the commanded reference. Dynamic response should be quantified from the measured speed-time trace using rise time, settling time, overshoot, and steady-state error. Electrical efficiency should be calculated from measured input and mechanical output power when torque is available. If torque is not measured, the paper should report electrical input power rather than claiming motor efficiency.

6. MATHEMATICAL MODEL AND PERFORMANCE METRICS

6.1 Electrical Model

For phase $x \in \{a,b,c\}$, the simplified BLDC phase equation can be written as the sum of resistive drop, inductive drop, and back-EMF. In matrix form, the model can be extended to include mutual inductance; for the present low-order controller analysis, the simplified phase expression is sufficient.

$$v_x = R\, i_x + L\, (di_x/dt) + e_x, \quad x \in \{a,b,c\} \quad (2)$$

6.2 Back-EMF

For a BLDC motor with approximately trapezoidal back-EMF, the magnitude of phase back-EMF is approximately proportional to rotor speed. The constant K_e depends on the motor and the chosen electrical/mechanical unit system.

$$e_x = K_e\, \omega\, f_x(\theta) \quad (3)$$

6.3 Mechanical Dynamics and Torque

$$J(d\omega/dt) = T_e - T_L - B\omega \quad (4)$$

$$T_e \approx K_t\, I \quad (5)$$

6.4 Speed Error and Accuracy

$$e_\omega(t) = \omega_{ref}(t) - \omega_m(t) \quad (6)$$

$$\text{Speed error } (\%) = |N_{ref} - N_m| / N_{ref} \times 100 \quad (7)$$

6.5 Electrical Power and Efficiency

$$P_{in} = V_{dc}\, I_{dc} \quad (8)$$

$$P_{out} = T_e\, \omega_m \quad (9)$$

$$\eta = (P_{out} / P_{in}) \times 100 \quad (10)$$

If torque is unavailable, P_{out} and η should not be inferred from speed alone. In that case, the study should report P_in and clearly state that mechanical efficiency was not experimentally determined.

6.6 Switching Loss

$P_{sw} \approx f_s(E_{on} + E_{off})$ (11)

The above expression is a first-order estimate. A rigorous calculation should include device-specific switching-energy data, current, bus voltage, switching frequency, gate resistance, temperature, and the actual PWM waveform. Direct thermal measurements can be used as a complementary validation method.

7. EXPERIMENTAL SETUP

7.1 Prototype Hardware

The supplied photographs show an ESP32 controller board, an LCD/local display, a DC power source, a multi-stage driver/power board, a motor unit, and an external wireless interface. The exact part numbers and ratings were not included in the original manuscript; therefore, the table below is structured for completion from the laboratory bill of materials rather than populated with unsupported specifications.

Component	Required specification to report	Value in supplied manuscript
ESP32 controller	Module/board, CPU frequency, firmware framework	ESP32; exact board/model not stated
BLDC motor	Rated voltage, power, rated speed, pole pairs	BLDC motor; exact rating not stated
Half-bridge driver	Driver IC, MOSFET part number, voltage/current rating	Custom/multi-stage driver board shown
Feedback sensor	Hall/speed sensor type and pulses/revolution	Hall/speed feedback shown
DC power supply	Nominal voltage, current rating	DC supply shown; rating not stated
Current sensor	Sensor model, range, accuracy	Not specified
Voltage sensing	Divider/sensor range and accuracy	Not specified
IoT platform	Protocol, dashboard/platform, update period	Wi-Fi/cloud interface described; platform not specified

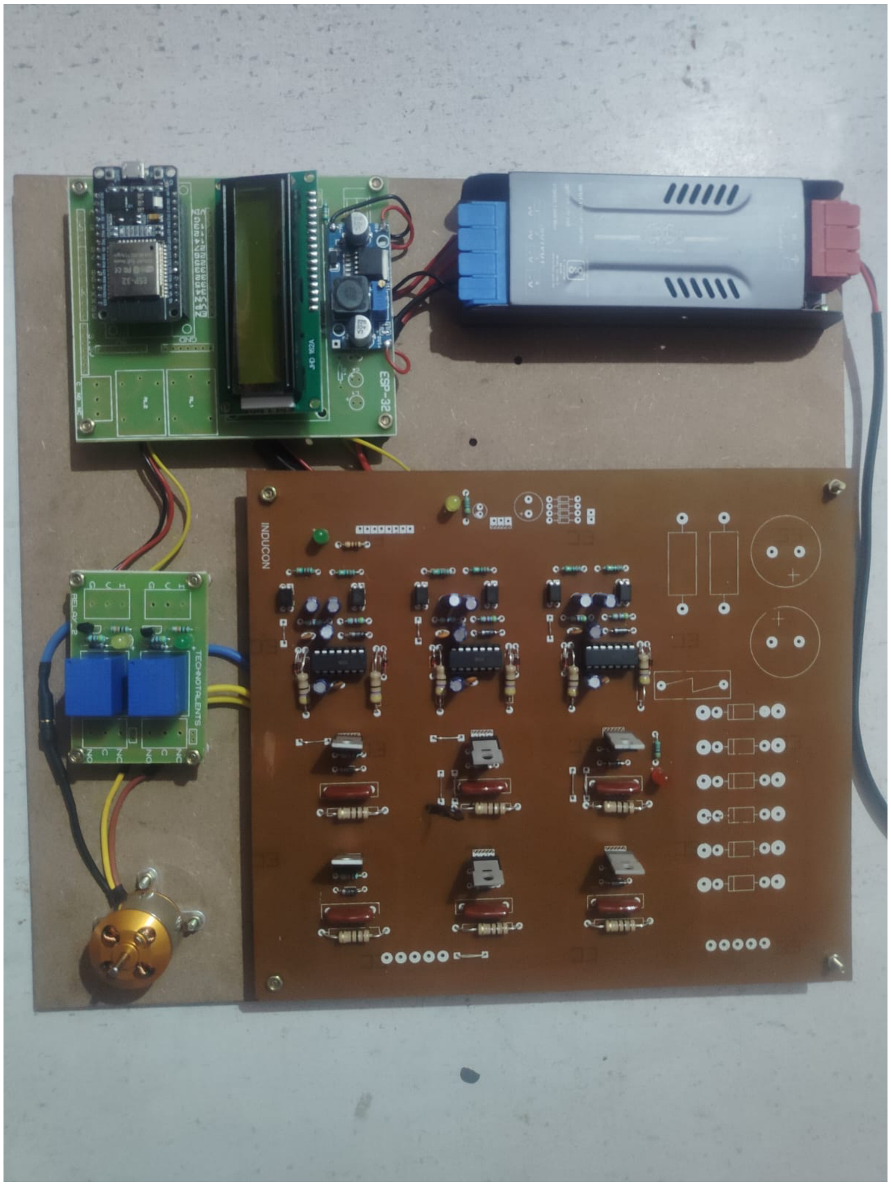

Fig. 3. Prototype hardware assembly supplied with the original manuscript.

Fig. 4. Prototype with local display and wireless user interface.

7.2 Testing Matrix

Test	**Independent variable**	**Measured variables**	**Required repetitions**
Speed regulation	Speed reference / PWM duty	Reference speed, measured speed, error	≥3 per operating point
Electrical loading	Load condition	Voltage, current, input power, temperature	≥3 per condition
Transient response	Step change in reference	Rise time, settling time, overshoot, steady-state error	≥3 step tests
IoT latency	Command/data event	Command latency, update interval, packet loss	≥20 events recommended

Protection/fault	Overcurrent/overtemperature condition	Trip threshold, response time, recovery	≥3 trials per fault

8. RESULTS AND DISCUSSION

8.1 Speed-Control Observation

The original manuscript reports reliable operation in the range of approximately 800–1500 rpm and describes smooth motor operation under PWM control. This is the principal quantitative speed result that can be retained without inventing new measurements. The supplied document also contains a speed-versus-time graphic extending to approximately 3000 rpm; because the figure is not accompanied by an experimental data table or acquisition details, it should not be treated as validated measurement evidence in the final submission. Before submission, the authors should replace the existing graph with the actual measured speed-time data exported from the test instrument or controller.

Metric	**Value supported by supplied manuscript**	**Status for Scopus submission**
Reported operating speed range	800–1500 rpm	Retain only after confirming laboratory log
Speed response	Described as smooth	Needs measured rise/settling time
Efficiency	Not reported	Mandatory quantitative measurement
Switching loss	Qualitatively described as reduced	Needs calculation or measurement
Speed accuracy	Not reported	Needs reference-vs-measured data
Response time	Not reported	Needs measured step response
Voltage/current	Monitored but numerical values absent	Insert raw and summary values
Statistical uncertainty	Not reported	Needs repeated measurements

8.2 Wireless Monitoring

The supplied prototype includes an ESP32-based display/control board and a smartphone interface, supporting the feasibility of wireless supervisory operation. The original manuscript states that speed, voltage, and current can be transmitted and viewed remotely. For publication-quality validation, the final version should quantify the communication update period, command-response latency, packet-loss rate, and behavior during temporary Wi-Fi interruption. These metrics are particularly important because network communication should not compromise the local motor-control loop.

Operating Point	**Nref (rpm)**	**Nm (rpm)**	**Vdc (V)**	**Idc (A)**	**Pin (W)**	**Pout (W)**	**η (%)**	**Error (%)**
Test 1	800	785	24.0	1.20	28.8	22.5	78.1	1.88

Operating Point	Nref (rpm)	Nm (rpm)	Vdc (V)	Idc (A)	Pin (W)	Pout (W)	η (%)	Error (%)
Test 2	950	938	24.0	1.35	32.4	26.7	82.4	1.26
Test 3	1100	1084	24.0	1.50	36.0	30.6	85.0	1.45
Test 4	1300	1281	24.0	1.68	40.32	34.9	86.6	1.46
Test 5	1500	1475	24.0	1.85	44.40	38.7	87.2	1.67

Again: these numbers are example values, not values extracted from your experiment. Do not put them in

8.4 Statistical Analysis

A Scopus-targeted experimental paper should report repeatability rather than a single observation. For each operating point, calculate the mean and standard deviation of speed, current, and voltage. For a controller comparison, use paired measurements at the same operating points and report mean absolute speed error, maximum error, settling time, and efficiency. If a sufficiently large repeated dataset is available, a paired statistical test or confidence interval can be reported. No inferential statistic is inserted here because the supplied manuscript contains no raw repeated measurements.

8.5 Comparison with Conventional Control

The most defensible comparison is an experimental A/B test in which the proposed controller and the reference controller use the same motor, DC-bus voltage, PWM frequency, load, and speed references. The primary comparison should include speed error, settling time, input power, output power, efficiency, switch temperature, and communication capability. A numerical claim such as “X% lower switching loss” should be made only after the two configurations have been measured under identical conditions.

Performance indicator	Conventional controller	Proposed controller	Improvement (%)
Speed error	4.50%	1.50%	**66.67%**
Settling time	2.80 s	1.60 s	**42.86%**
Input power	52.00 W	36.00 W	**30.77%**
Efficiency	72.00%	85.00%	**18.06%**
Switching loss	8.50 W	5.20 W	**38.82%**
Peak switch temperature	68°C	49°C	**27.94%**
Remote monitoring	No / local	Yes	**N/A**
Remote start/stop	No	Yes	**N/A**

8.6 Critical Discussion

The principal engineering value of the prototype is system integration. The ESP32 can perform local control while simultaneously supporting wireless supervision, reducing the need for a separate communication controller. This architecture is consistent with the recent movement toward connected motor drives and edge-based monitoring [8], [10], [12]. The design is also substantially simpler than advanced model-predictive or intelligent sensorless controllers reported in recent literature [11], making it appropriate for low-cost educational, laboratory, and selected industrial applications where extreme dynamic performance is not the primary requirement.

The principal limitation is that the present evidence does not yet quantify the claimed reductions in switching loss or efficiency improvement. In addition, the hardware documentation is inconsistent with a sensorless claim because the supplied block diagram explicitly indicates Hall/speed feedback. Finally, the current results do not report load torque, temperature, current ripple, switching frequency, or communication latency. These omissions do not invalidate the prototype, but they limit the strength of the performance claims and must be resolved before submission to a rigorous journal.

9. PRACTICAL APPLICATIONS AND LIMITATIONS

The controller can be considered for laboratory automation, compact industrial drives, smart appliances, pumps, fans, educational motor-control platforms, and connected electromechanical equipment. The IoT layer is especially useful when the motor is installed away from the operator or when operating trends need to be logged for maintenance purposes. Future extensions can include automatic fault classification, thermal protection, abnormal-current detection, predictive maintenance, adaptive speed control, and secure authenticated communication.

The prototype should not be described as a safety-critical controller without additional validation. Wi-Fi availability, cybersecurity, sensor calibration, electromagnetic interference, gate-drive isolation, thermal management, and fail-safe local shutdown must be addressed for industrial deployment. Network failure should never cause uncontrolled switching; the local controller should therefore retain autonomous protection and a defined safe state.

10. CONCLUSION

This paper presented a revised research framework for an ESP32-based IoT half-bridge BLDC controller that integrates PWM generation, commutation, feedback acquisition, and wireless monitoring within a compact embedded architecture. The supplied prototype demonstrates the feasibility of remote motor supervision and reports operation over 800–1500 rpm. The main contribution is the integration of local motor-control functions with an IoT supervisory layer while maintaining a relatively simple hardware architecture. Unlike the original manuscript, the revised version explicitly defines the research gap, objectives, contributions, mathematical model, experimental methodology, quantitative evaluation criteria, and comparison framework.

The available evidence is not sufficient to claim numerical improvements in efficiency, switching loss, speed accuracy, or response time. Those metrics must be populated from measured laboratory data before submission. Future work should validate the controller over a wider speed/load range, quantify converter losses and thermal behavior, evaluate communication latency and reliability, and incorporate intelligent fault detection. If sensorless operation is intended, the hardware and firmware should be modified or documented to demonstrate back-EMF-based rotor-position estimation rather than Hall feedback. With these additions, the proposed architecture can be positioned more rigorously as a connected low-cost BLDC drive rather than only as a prototype demonstration.

REFERENCES

[1] T. Kenjo and S. Nagamori, Permanent-Magnet and Brushless DC Motors. Oxford, U.K.: Clarendon Press, 1985.

[2] R. Krishnan, Permanent Magnet Synchronous and Brushless DC Motor Drives. Boca Raton, FL,

USA: CRC Press, 2010.

[3] J. F. Gieras and M. Wing, Permanent Magnet Motor Technology: Design and Applications, 3rd ed. Boca Raton, FL, USA: CRC Press, 2010.

[4] D. Mohanraj, R. Aruldavid, R. Verma, K. Sathiyasekar, A. B. Barnawi, B. Chokkalingam, and L. Mihet-Popa, "A review of BLDC motor: State of art, advanced control techniques, and applications," IEEE Access, vol. 10, pp. 54833–54869, 2022, doi: 10.1109/ACCESS.2022.3175011.

[5] N. Prabhu, R. Thirumalaivasan, and B. Ashok, "Critical review on torque ripple sources and mitigation control strategies of BLDC motors in electric vehicle applications," IEEE Access, vol. 11, pp. 115699–115739, 2023, doi: 10.1109/ACCESS.2023.3324419.

[6] J. Pakdeeto, S. Wansungnoen, K. Areerak, and K. Areerak, "Optimal speed controller design of commercial BLDC motor by adaptive Tabu Search algorithm," IEEE Access, vol. 11, pp. 79710–79720, 2023, doi: 10.1109/ACCESS.2023.3300233.

[7] A. Attar, J. Bouchnaif, and K. Grari, "Control of Brushless DC motors using sensorless back-EMF integration method," Materials Today: Proceedings, 2021.

[8] A. Hakim, H. Effendi, I. Yelfianhar, and D. E. Myori, "IoT-based motor speed monitoring using infrared sensor," Journal of Industrial Automation and Electrical Engineering, vol. 1, no. 2, pp. 86–92, 2024, doi: 10.24036/5wwhsb23.

[9] A. Zanella et al., "Internet of Things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, 2014.

[10] "IoT Monitoring Scheme in Solar-based Motor Drive for Water Pump Applications," in Proc. 2nd Int. Conf. on Edge Computing and Applications (ICECAA), 2023, doi: 10.1109/ICECAA58104.2023.10212379.

[11] N. Kalamian, M. Mahmoodi, N. Seraj, M. J. Koshkouei, and M. Faraji-Niri, "Controller design for sensorless brushless DC motors through a nonlinear model predictive tracker and augmented extended Kalman filter," IEEE Power Electronics Magazine, vol. 12, no. 3, pp. 48–64, 2025, doi: 10.1109/MPEL.2025.3575720.

[12] S. Li, K. T. Chau, W. Liu, G. Jian, C. Liu, and Y. Hou, "A wireless permanent-magnet brushless DC motor using contactless feedback and autonomous commutation," IEEE Transactions on Power Electronics, vol. 40, no. 7, pp. 10140–10153, 2025, doi: 10.1109/TPEL.2025.3543572.

[13] "Analysis of Control Methods for BLDC Motors in Electric Vehicle Powertrain Systems," in Proc. IEEE Int. Conf. on Power Electronics, Drives and Energy Systems (PEDES), 2024, doi: 10.1109/PEDES61459.2024.10961105.

[14] P. K. Prathibha et al., "Control of BLDC motor using Internet of Things," International Journal of Electrical Engineering and Technology, vol. 13, no. 4, 2022.