iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** XI **Month of publication:** November 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.85618

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue XI Nov 2026- Available at www.ijraset.com*

# A Review of Intelligent Hybrid Vehicle Control Unit and a Proposed ESP32-Based ECU Framework for an Educational Hybrid Electric Vehicle Trainer

Abhijeet Mandhare[1], Ansh Thakur[2], Soham Mohite[3], Niraj Bhoir[4], Dr. T. Z. Quazi[5]
*Department of Automobile Engineering, Saraswati college Of Engineering, Mumbai University, India*

***Abstract:*** ***An electronic control unit (ECU) in a hybrid electric vehicle (HEV) receives information from sensors, evaluates the vehicle operating condition, and selects suitable control states through an energy-management strategy (EMS). The choice of control method affects decision quality, response, computational demand, and implementation cost. This paper reviews recent HEV energy-management research published from 2022 to 2026, with emphasis on rule-based control, reinforcement learning, deep reinforcement learning, model predictive control, and low-cost hardware validation. The Ten relevant journal studies are compared according to their control approach, input information, validation method, reported contribution, and limitations. The review indicates that learning- and prediction-based methods can improve optimisation under varying driving conditions, but often depend on detailed models, training data, or host-computer resources. Rule-based control remains attractive where transparency, predictable execution, and low implementation complexity are important. Based on these findings, a input–decision–output using an ESP32 microcontroller is presented for an educational HEV trainer. The ECU accepts driver-demand, vehicle-state, battery-state, and safety-related inputs; applies sensor validation and priority-based decision rules; and produces operating-mode and warning outputs for display. It defines a technically grounded architecture and a future validation plan. The main contribution is to connect recent HEV EMS research with a simple, transparent, and potentially low-cost embedded ECU suitable for educational demonstration.***

***Keywords: Hybrid electric vehicle, electronic control unit, hybrid vehicle control unit, energy management strategy, ESP32, rule-based control, sensor inputs, decision logic, embedded system.***

## I. INTRODUCTION

Hybrid electric vehicles combine more than one energy source and require a supervisory controller to decide how the available sources should be used. The energy-management strategy is therefore an important part of the vehicle control system. It receives information such as driver power demand, vehicle speed, battery state of charge (SOC), and operating constraints, and converts this information into control decisions. The quality of these decisions influences energy use, battery operation, drivability, and system reliability [1]-[4]. Research on HEV energy management has moved from fixed rule-based methods towards optimisation and learning-based methods. Rule-based strategies use calibrated thresholds or operating maps. Optimisation-based strategies include dynamic programming, equivalent consumption minimisation, and model predictive control (MPC). Learning-based approaches include Q-learning and deep reinforcement learning (DRL). Each method has a different balance between decision quality, computation, training effort, transparency, and real-time implementation [2]-[7].

Although advanced methods are valuable for research, an educational trainer has a different immediate requirement. Its ECU should read a manageable set of inputs, make understandable decisions, identify invalid or unsafe input conditions, and display the selected operating state. A low-cost microcontroller can be considered for this purpose if its electrical and timing limitations are respected. This paper therefore focuses on the ECU decision layer and its ESP32-based implementation concept.

## II. AIM, OBJECTIVES AND SCOPE

### *A. Aim*

The aim of this review is to examine recent HEV energy-management decision strategies and use the findings to a practical ESP32-based ECU input–output for an educational trainer.

©IJRASET: All Rights are Reserved 

### B. Objective

1) Review recent research on rule-based, optimisation-based, and learning-based HEV energy-management strategies.
2) Compare the information used by these strategies, their decision methods, and their validation approaches.
3) Identify the benefits and limitations of low-cost hardware and hardware-in-the-loop (HIL) implementation.
4) Identify a suitable set of ECU inputs and logical outputs for an educational HEV trainer.
5) Justify the selection of ESP32 for a low-cost prototype while identifying the electrical, computational, and safety limitations that must be addressed.
6) A transparent decision-making sequence, including sensor validation, priority handling, operating-state selection, and fault indication.

### C. Scope

The proposed outputs are ECU-level operating-state commands and display/status signals. The paper does not specify motor-driver circuits, engine actuation, power-electronic design, mechanical transmission, or measured vehicle performance. Any future connection to physical actuators would require a separate safety and interface design.

## III. REVIEW METHODOLOGY

A focused narrative review was adopted. Searches were carried out on scholarly search services and publisher platforms, including ScienceDirect, MDPI, Wiley Online Library, IEEE Xplore records, and publisher-indexed article pages. The search was conducted on 3 Oct 2026. Search phrases included combinations of: “hybrid electric vehicle energy management strategy”, “HEV rule-based control”, “reinforcement learning energy management”, “model predictive control HEV”, “low-cost hardware-in-the-loop HEV”, and “ESP32”.

Studies were considered relevant when they were published between 2022 and 2026 and addressed at least one of the following: HEV energy-management decisions, rule-based or learning-based control, real-time or HIL validation, or low-cost controller implementation. Publisher records were checked for article title, authors, year, journal, article number, and DOI. Ten journal articles were selected for the comparative discussion. The official Espressif ESP32 datasheet was used separately as a hardware specification source [11].

This is a focused narrative review. No claim is made that every paper in the field was retrieved, that screening was independently performed by multiple reviewers.

## IV. OVERVIEW OF HEV ECU AND ENERGY-MANAGEMENT STRATEGIES

### A. ECU and EMS Functions

An ECU is an embedded electronic controller that receives input signals, processes them according to programmed logic, and generates output signals. In an HEV supervisory control context, the EMS is the decision logic that selects the operating state or energy-management action. The terms ECU and EMS are related but not identical: the ECU is the hardware/software platform, while the EMS is the control strategy executed by it.

A typical decision cycle contains five stages: (i) signal acquisition, (ii) signal conditioning and validation, (iii) state estimation or classification, (iv) decision-making, and (v) output update and fault monitoring. This separation is useful for embedded implementation because sensor errors can be detected before they affect the decision logic.

### B. Main classes of Energy-Management Methods

| Method | Basic principle | Strengths | Limitations for a low-cost educational ECU |
|---|---|---|---|
| Rule-based control | Uses fixed thresholds, state machines, or calibrated maps. | Simple, transparent, low computational demand, easy to demonstrate. | Thresholds require calibration; performance may be suboptimal outside the selected conditions. |
| Optimisation-based control | Minimises a defined cost function; examples include DP, ECMS and MPC. | Can handle multiple objectives and constraints; useful benchmark methods. | May require a model, optimisation calculations, or future-condition prediction. |

| | | | |
|---|---|---|---|
| Reinforcement learning | Learns actions from state, action, and reward interactions. | Can adapt to complex or changing operating conditions. | Training, state-space design, validation, and safe deployment are challenging. |
| Deep reinforcement learning | Uses neural networks to represent complex state/action relationships. | Can handle larger or continuous state spaces than basic tabular RL. | Training and inference may require more computation and data; explainability is reduced. |
| Hybrid/hierarchical methods | Combines rules with prediction, optimisation, or learning. | Can combine transparent safety rules with more advanced decisions. | Architecture, tuning, and verification become more complex. |

## V. REVIEW OF SELECTED RESEARCH

### *A. Cheng, Xu and Chen (2022): Q-learning for a hybrid energy-storage system*

Cheng et al. proposed a Q-learning-based energy-management strategy for a hybrid electrical system using a lithium battery and a supercapacitor. The method used demand power and battery SOC as state information and selected the supercapacitor output power as an action. The study used simulation to assess the strategy under changing operating conditions [1]. Its relevance to ECU design is the clear relationship between measured state variables and a selected control action. However, the study is not an ESP32 implementation and its learning method should not be assumed to run directly on a small microcontroller without resource assessment.

### *B. Du et al. (2022): Prioritised deep reinforcement learning*

Du et al. developed a double deep Q-learning framework for a series hybrid electric tracked vehicle. The method combined prioritised experience replay and an adaptive neural-network optimisation method. Simulation results were compared with dynamic programming and an earlier DRL method, with the authors reporting improved training efficiency and energy performance [2]. The work demonstrates the potential of advanced learning for energy-management decisions. Its training process, neural-network structure, and workstation-based computation make it a reference for algorithm development rather than a direct low-cost ECU template.

### *C. Liu et al. (2023): Imitation reinforcement learning*

Liu et al. proposed an imitation-reinforcement-learning approach for HEV energy management. The method incorporated optimal-control information into the learning process and used a state representation involving battery SOC and travel distance. The study evaluated adaptability through simulation under different driving conditions [3]. This work highlights the value of selecting meaningful input variables and using prior control knowledge. Its computational and training requirements remain a consideration for embedded deployment.

### *D. Yang et al. (2023): Reinforcement learning combined with MPC*

Yang et al. combined reinforcement learning with an MPC framework and used a long short-term memory network to predict vehicle speed. The authors reported simulation and HIL validation of fuel-economy and real-time performance [4]. This paper is relevant because it links predicted operating conditions with repeated control decisions and includes HIL testing. However, the presence of a prediction network and MPC optimisation makes the approach more complex than the fixed-rule logic proposed for the first version of an educational ESP32 ECU.

### *E. Wang et al. (2023): Comparison of thirteen DRL methods*

Wang et al. compared thirteen deep reinforcement-learning algorithms for HEV energy management using a common vehicle model. The comparison considered discrete and continuous action spaces, convergence behaviour, training time, and final test performance [5]. The study is valuable because it shows that algorithm selection should be based on more than a single reported fuel-economy value. It also reinforces that learning-based methods require a defined training and evaluation procedure. The paper does not establish that these methods are suitable for direct execution on an ESP32.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*F. Essa et al. (2023): Low-cost HIL controller testing*

Essa et al. investigated an ANN-enhanced MPC controller for HEV speed control using an HIL arrangement. The low-cost hardware included an Arduino Mega 2560 connected to a host computer. The study compared the proposed controller with PI, classical MPC, and ANN configurations [6]. This is an important embedded-control reference because it demonstrates how a relatively inexpensive microcontroller can participate in an HIL test setup. The host computer performed an important part of the test environment; therefore, the paper should not be interpreted as proof that the full neural predictive controller operated independently on the Arduino.

*G. Urooj and Nasir (2024): Review of intelligent HEV energy management*

Urooj and Nasir reviewed HEV energy-management techniques with attention to rule-based and reinforcement-learning approaches. They also identified challenges associated with uncertainty, multi-layer control, instrumentation, user behaviour, and weather-related factors [7]. This review supports the need to treat sensing and input quality as part of the control problem rather than as a separate hardware detail. It also supports the use of modular control layers.

*H. Mittal and Shah (2024): Energy-management technology roadmap*

Mittal and Shah presented a technology roadmap covering current rule-based and optimisation-based strategies and future directions involving machine learning, deep learning, and additional information sources [8]. Their discussion indicates that future EMSs may use richer information, including driver demand, vehicle speed, SOC, traffic, and weather. For an educational prototype, however, the input set should remain limited to signals that can be measured or explicitly simulated and validated.

*I. Tresca et al. (2024): Deep Q-learning EMS*

Tresca et al. developed a deep Q-learning energy-management system for a plug-in HEV. The agent was trained using the WLTC and different reward formulations were assessed against a dynamic-programming benchmark. The authors also evaluated the selected reward formulation under driving scenarios different from the training cycle [9]. This work illustrates the importance of defining input states, control actions, reward functions, and out-of-training evaluation. For the proposed ECU, a deterministic state machine is more straightforward to verify initially; a learned strategy can be considered in later research.

*J. Ben Slimene et al. (2026): Low-cost rule-based implementation*

Ben Slimene et al. reported a rule-based EMS implemented using an Arduino Mega 2560. The study combined MATLAB/Simulink model development with experimental validation on a 1:10 scale prototype and reported control-loop timing and energy-management results [10]. This is particularly relevant to low-cost educational applications because it demonstrates that a calibrated rule-based strategy can be implemented on inexpensive embedded hardware. Nevertheless, the reported values belong to the authors specific prototype and test conditions. They cannot be transferred to an ESP32 design or to the present trainer without independent testing.

## VI. COMPARATIVE ANALYSIS AND DISCUSSION

The selected studies represent complementary research directions. The learning-based studies focus on improving decisions under varying conditions, while the HIL and low-cost hardware studies address practical execution and validation. The review suggests that algorithm performance and implementation feasibility must be considered separately. A method that performs well in simulation may still require additional work before it can be executed reliably on a microcontroller.

| Ref. | Main approach | Input/state information highlighted | Validation reported | Relevance to proposed ECU |
|---|---|---|---|---|
| [1] | Q-learning | Demand power, battery SOC | Simulation | Shows explicit mapping from inputs to action. |
| [2] | Double deep Q-learning | Vehicle operating/state information | Simulation and driving-cycle evaluation | Shows advanced learning potential; high training complexity. |
| [3] | Imitation RL | SOC, travel distance and operating state | Simulation | Shows value of meaningful state selection. |

| | | | | |
|---|---|---|---|---|
| [4] | RL + MPC + LSTM prediction | Vehicle speed and EMS state variables | Simulation and HIL | Shows prediction and real-time validation approach. |
| [5] | 13 DRL algorithms compared | Model state and action variables | Simulation benchmark | Shows need for consistent comparison. |
| [6] | ANN-enhanced MPC | Speed reference, speed error and feedback | Simulation and HIL using Arduino + PC | Supports low-cost HIL, not stand-alone ESP32 proof. |
| [7] | Review of rule-based and RL methods | Vehicle, storage and operating information | Literature review | Highlights instrumentation and uncertainty. |
| [8] | EMS technology roadmap | Driver demand, speed, SOC and future context | Literature review | Supports staged development and limited input set. |
| [9] | Deep Q-learning | Vehicle state, SOC and driving-cycle information | Simulation across driving cycles | Shows training and generalisation considerations. |
| [10] | Rule-based EMS | Driver demand and battery SOC among control inputs | Simulation + 1:10 prototype | Closest low-cost implementation reference. |

A direct numerical ranking is not appropriate because the studies use different vehicle models, driving cycles, objectives, hardware platforms, and validation methods. For example, simulation fuel-consumption results cannot be compared directly with HIL controller response or a scale-prototype result. The more easy comparison is qualitative: what inputs are used, what decision method is applied, how the method is validated, and what computational or instrumentation requirements are reported.

Three findings are particularly relevant. First, driver demand and battery SOC appear repeatedly as important decision variables [1], [8], [10]. Second, advanced learning and prediction methods can improve adaptation but introduce training and computation requirements [2]-[5], [9]. Third, low-cost hardware studies show the value of staged implementation and HIL testing, but the processor and host-computer roles must be stated accurately [6], [10]. These findings support a first-stage ESP32 design based on deterministic, explainable rules with sensor validation and clear output states.

## VII. PROPOSED ESP32-BASED ECU FRAMEWORK

### *A. Why ESP32 is considered*

The ESP32 is considered as a candidate controller because the original ESP32 family integrates a 32-bit processor, GPIO, ADC, PWM, UART, SPI, I²C, timers, and wireless communication interfaces. These peripherals can support analogue sensor acquisition, digital switches, display communication, and optional wireless monitoring [11]. This combination can reduce the number of separate controller modules required for a basic educational prototype.

The selection is a proposal, not a claim that ESP32 is an automotive-qualified production ECU. The exact board and chip variant must be specified. ESP32 GPIO is not 5-V tolerant; analogue inputs must be conditioned so that the pin voltage remains within the permitted range. ADC readings require calibration and filtering, and the ADC performance can be affected by non-linearity and noise.

The classic ESP32 also has pin-function restrictions and ADC2 interactions with Wi-Fi; ADC1 pins are generally preferable for analogue sensing when Wi-Fi is active. External signal conditioning, voltage dividers, protection, stable power regulation, and isolation where required must be designed before connecting trainer signals [11].

ESP32 is therefore suitable for investigation as a low-cost educational controller where the required decision logic is modest. Its suitability must be demonstrated through measurement of input accuracy, loop time, output reliability, fault response, and stability under the intended operating conditions.

*B. Proposed ECU inputs*

| Input variable | Suggested signal/source | Purpose in ECU decision |
|---|---|---|
| Accelerator position | Potentiometer or position sensor with conditioned analogue output | Represents driver power demand; may be converted to low, medium, or high demand. |
| Brake input | Digital brake switch or conditioned analogue signal | Identifies braking request and gives braking logic priority over normal drive-mode selection. |
| Vehicle speed | Hall sensor/pulse input or simulated speed signal | Provides a vehicle operating-state variable and supports stationary/low-speed logic. |
| Battery voltage | Voltage sensor/divider with protection or BMS-reported value | Supports battery monitoring; voltage alone is not a complete SOC measurement. |
| Battery current | Current sensor or BMS-reported value | Can support SOC estimation and charge/discharge indication. |
| Battery SOC | BMS value or estimated value from a defined battery model | Used as a decision constraint; estimation method and accuracy must be stated. |
| Vehicle load | Load sensor or user-selected/simulated trainer input | Represents low or high load demand when a physical load sensor is unavailable. |
| Temperature | Temperature sensor with suitable interface | Supports temperature-limit monitoring and fault/warning logic. |
| Fault/status input | Digital signal or software diagnostic flags | Indicates sensor, communication, or system abnormality. |
| Road condition/gradient (optional) | Predefined scenario input or external estimate | May be included only if the trainer provides a credible measured or simulated value. |

*C. Proposed ECU outputs*

| **Output** | **Meaning** | **Suggested destination** |
|---|---|---|
| Operating-mode code | EV, engine, hybrid, or braking/regen status selected by the decision logic | Display or mode indicator |
| Battery status | SOC band and charge/discharge/normal status | Display |
| Driver-demand status | Low, medium, or high demand classification | Display/serial monitor |
| Fault warning | Sensor invalid, over-temperature, low-SOC warning, or general fault | Warning LED/display/buzzer |
| System-ready status | Indicates that startup checks and input validation have passed | Display/ready indicator |
| Diagnostic data | Raw/filtered inputs, selected state, and fault flags | Serial monitor or optional wireless dashboard |

In this paper, EV, engine, hybrid and outputs are logical operating-state indications for the educational trainer. They are not claims that the ESP32 directly controls an engine, traction motor, inverter, or high-voltage battery. Any future actuator interface would require separate driver electronics, interlocks, and safety validation.

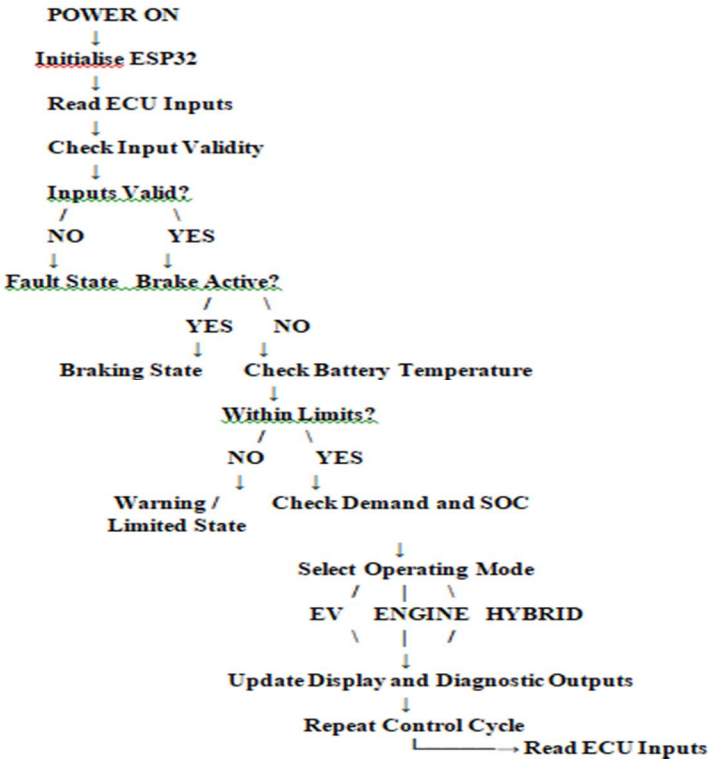
*D. Proposed decision-making logic*

A rule-based finite-state approach is proposed for the first version because its decisions are transparent and can be tested with known input combinations. The ECU should execute the following priority order: (1) startup and sensor checks, (2) fault handling, (3) brake-request handling, (4) battery and temperature constraint checks, and (5) normal operating-mode selection. Safety-related conditions must take priority over normal mode selection.

| Priority | Condition (illustrative) | ECU decision/output |
|---|---|---|
| 1 | Power-up or reset | Initialise peripherals; set outputs to a safe default; run self-check. |
| 2 | Critical sensor invalid, over-temperature, or diagnostic fault | Enter FAULT state; inhibit normal mode selection; display fault code. |
| 3 | Brake input active | Select BRAKING state; indicate regenerative-braking request only when battery/temperature constraints permit; otherwise show braking with recovery unavailable. |
| 4 | Battery SOC below configured lower limit | Set LOW-SOC warning; prevent selection of an SOC-dependent electric mode. |
| 5 | Accelerator demand low and SOC within permitted range | Select EV-mode indication. |
| 6 | Accelerator demand high or load demand high | Select HYBRID-mode indication, subject to valid input and system constraints. |
| 7 | Demand moderate and SOC below preferred band | Select ENGINE-mode indication or charge-support state as defined by the trainer's agreed logic. |
| 8 | No active demand and speed near zero | Select IDLE/STANDBY indication. |
| 9 | No special condition | Maintain the normal state and refresh display/diagnostics. |

The thresholds in this table are placeholders to be calibrated for the trainer. No universal accelerator percentage, speed limit, SOC threshold, or temperature limit is asserted. Final values must be selected from the actual sensor ranges, battery specifications, trainer operating instructions, and supervisor-approved requirements. Hysteresis should be added around thresholds to prevent rapid switching when a measured value fluctuates near a boundary.

*E. ECU flowchart*



## VIII. RESEARCH GAP AND PROPOSED CONTRIBUTION

The reviewed literature contains extensive work on energy-management algorithms, including reinforcement learning, DRL, MPC, and rule-based control. It also includes examples of low-cost hardware and HIL validation. However, the selected studies do not establish a ready-made ESP32 input–decision–output specifically designed for a simple educational HEV trainer. In particular, there is a practical need to connect the choice of ECU inputs, signal validation, priority-based decision logic, and understandable output states in one transparent structure.

The proposed contribution is therefore not a new optimisation algorithm and not a claim of improved vehicle fuel economy. It is a structured, low-cost embedded ECU concept that translates common HEV operating information into explainable status decisions. Its value is educational and implementation-oriented: students can observe how input changes affect the selected state and how fault conditions override normal decisions.

Similar rule-based and embedded control concepts already exist, including the low-cost Arduino implementation reported by Ben Slimene et al. [10]. The distinction proposed here is the ESP32-focused signal architecture and the explicit inclusion of input validation, fault priority, and trainer-oriented output mapping. A stronger novelty claim would require a broader search of ESP32-based automotive and educational control systems.

## IX. VALIDATION PLAN AND PERFORMANCE MEASURES

Because the proposed ESP32 framework has not yet been tested, the following items are recommended as future validation work. Results should be added only after the hardware and software have been implemented and measured.

| Test area | Suggested test | Measure/report |
|---|---|---|
| Input accuracy | Apply known voltages or switch states to each input channel. | ADC error, repeatability, noise, and valid operating range. |
| Signal validation | Disconnect, short, freeze, or vary an input outside its configured range. | Fault detection success and detection time. |
| Decision correctness | Apply a test matrix covering accelerator, brake, SOC, speed, load, and temperature combinations. | Correct state selection compared with the approved decision table. |
| Priority handling | Apply brake and fault conditions while normal mode conditions are also true. | Confirmation that safety/fault priority overrides normal decisions. |
| Timing | Measure repeated control-cycle execution under normal and communication-load conditions. | Average, maximum, and worst-case loop time. |
| Stability | Apply inputs close to decision thresholds. | Unwanted state switching and effectiveness of hysteresis. |
| Reliability | Run repeated long-duration operation and test watchdog recovery. | Resets, missed updates, fault recovery, and stable operation. |
| Cost and power | Record actual board, interface, protection, and power-supply costs. | Total prototype cost and supply requirements. |

Testing should first be performed with low-voltage signals. Any connection to a battery system or vehicle actuator must be reviewed separately for voltage compatibility, current limits, isolation, transient protection, and safe failure behaviour.

## X. LIMITATIONS AND FUTURE WORK

This review is limited by its focused narrative search and the use of ten selected journal articles. The ESP32 framework is conceptual and has not been experimentally evaluated. The final input set and thresholds also depend on the sensors and operating specifications available on the actual trainer.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Future work should implement the input interface and decision logic on the selected ESP32 board, calibrate the sensors, define SOC estimation or BMS communication, and test the flowchart using a structured test matrix. Further work may compare a simple rule-based controller with a more advanced method on the same input set. HIL testing can be introduced before any actuator-level integration. Measured timing, input error, fault response, and cost should then be reported.

## XI. CONCLUSION

This paper reviewed ten recent studies on HEV energy-management strategies and low-cost controller implementation. The literature shows that rule-based methods offer transparency and low computational demand, while MPC and learning-based methods provide more advanced optimisation and adaptability at the cost of greater modelling, training, or computational requirements. HIL and low-cost hardware studies demonstrate the importance of testing controller behaviour under repeatable conditions. Based on this comparison, an ESP32-based ECU framework is proposed for an educational HEV trainer. The framework uses driver-demand, vehicle-state, battery-state, and safety-related inputs; validates signals; applies priority-based decision rules; and generates operating-mode, battery-status, and fault-warning outputs. ESP32 is a low-cost platform for this limited supervisory and demonstration function, provided that its voltage limits, ADC behaviour, pin restrictions, signal conditioning, and timing are addressed. The proposed framework is a design concept rather than a validated result.

## REFERENCES

[1] Y. Cheng, G. Xu, and Q. Chen, “Research on energy management strategy of electric vehicle hybrid system based on reinforcement learning,” Electronics, vol. 11, no. 13, Art. no. 1933, 2022. doi: 10.3390/electronics11131933. Research on Energy Management Strategy of Electric Vehicle Hybrid System Based on Reinforcement Learning

[2] G. Du, Y. Zou, X. Zhang, L. Guo, and N. Guo, “Real-time predictive energy management of plug-in hybrid electric vehicles for coordination of fuel economy and battery degradation”. doi: 10.1016/j.energy.2021.122523. Real-time predictive energy management of plug-in hybrid electric vehicles for coordination of fuel economy and battery degradation - ScienceDirect

[3] Y. Liu, Y. Wu, X. Wang, L. Li, Y. Zhang, and Z. Chen, “Energy management for hybrid electric vehicles based on imitation reinforcement learning,” Energy, vol. 263, Art. no. 125890, 2023. doi: 10.1016/j.energy.2022.125890. Energy management for hybrid electric vehicles based on imitation reinforcement learning - ScienceDirect

[4] N. Yang, S. Ruan, L. Han, H. Liu, L. Guo, and C. Xiang, “Reinforcement learning-based real-time intelligent energy management for hybrid electric vehicles in a model predictive control framework,” Energy, vol. 270, Art. no. 126971, 2023. doi: 10.1016/j.energy.2023.126971. Reinforcement learning-based real-time intelligent energy management for hybrid electric vehicles in a model predictive control framework - ScienceDirect

[5] H. Wang, Y. Ye, J. Zhang, and B. Xu, “A comparative study of 13 deep reinforcement learning based energy management methods for a hybrid electric vehicle,” Energy, vol. 266, Art. no. 126497, 2023. doi: 10.1016/j.energy.2022.126497. A comparative study of 13 deep reinforcement learning based energy management methods for a hybrid electric vehicle - ScienceDirect

[6] M. E.-S. M. Essa, J. V. W. Lotfy, M. E. K. Abd-Elwahed, K. Rabie, B. M. ElHalawany, and M. Elsisi, “Low-cost hardware in the loop for intelligent neural predictive control of hybrid electric vehicle,” Electronics, vol. 12, no. 4, Art. no. 971, 2023. doi: 10.3390/electronics12040971. View article

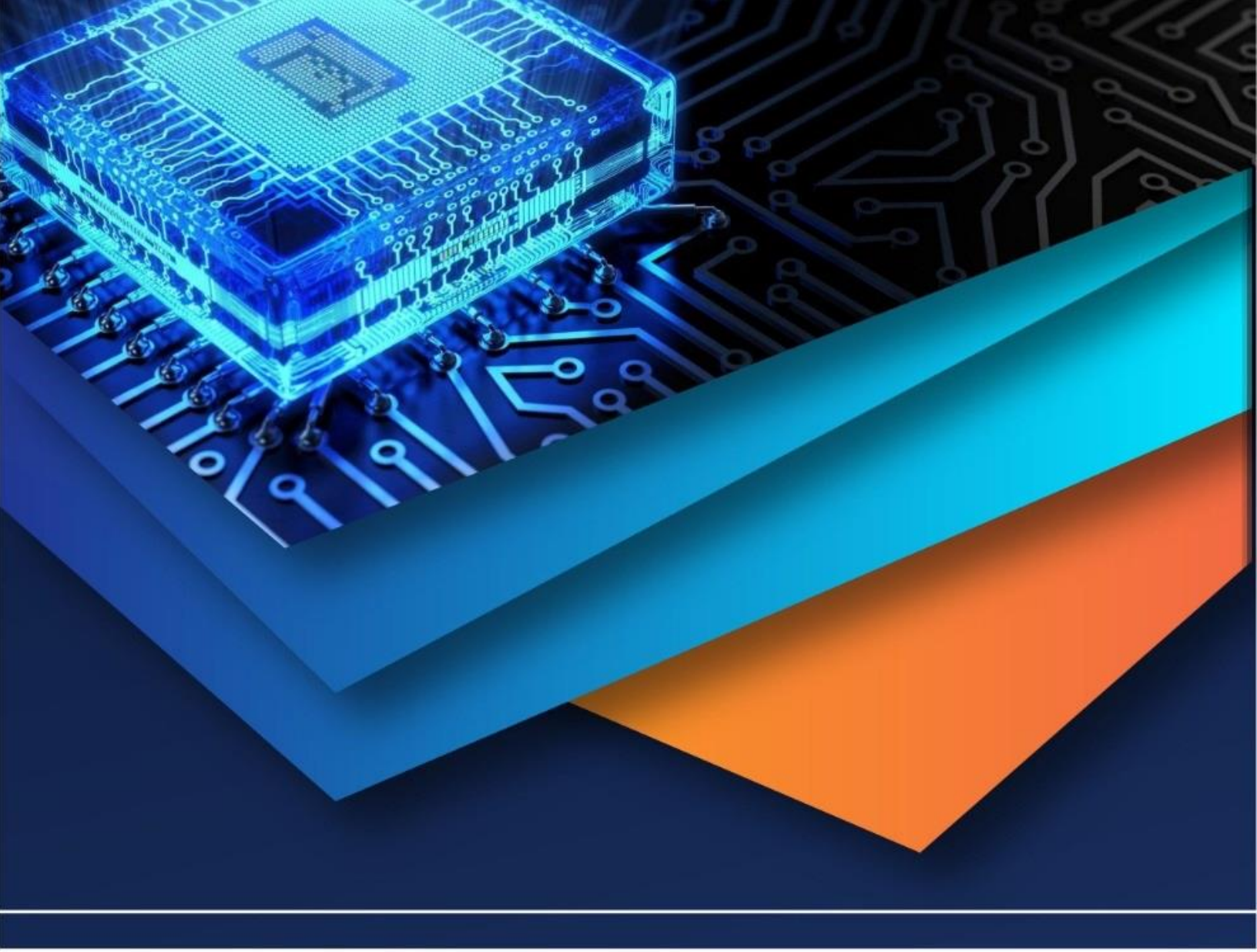
[7] A. Urooj and A. Nasir, “Review of intelligent energy management techniques for hybrid electric vehicles,” Journal of Energy Storage, vol. 92, Art. no. 112132, 2024. doi: 10.1016/j.est.2024.112132. Review of intelligent energy management techniques for hybrid electric vehicles - ScienceDirect

[8] V. Mittal and R. Shah, “Energy management strategies for hybrid electric vehicles: A technology roadmap,” World Electric Vehicle Journal, vol. 15, no. 9, Art. no. 424, 2024. doi: 10.3390/wevj15090424. Energy Management Strategies for Hybrid Electric Vehicles: A Technology Roadmap

[9] L. Tresca, L. Pulvirenti, L. Rolando, and F. Millo, “Development of a deep Q-learning energy management system for a hybrid electric vehicle,” Transportation Engineering, vol. 16, Art. no. 100241, 2024. doi: 10.1016/j.treng.2024.100241. Development of a Deep Q-Learning Energy Management System for a Hybrid Electric Vehicle | Request PDF

[10] M. Ben Slimene, M. A. Khlifi, F. Mahroogi, and I. Tlili, “Energy management for low-cost and high-efficiency of hybrid electric vehicles,” International Journal of Energy Research, 2026, Art. no. 4534295. doi: 10.1155/er/4534295. Energy Management for Low-Cost and High-Efficiency of Hybrid Electric Vehicles - Slimene - 2026 - International Journal of Energy Research - Wiley Online Library

[11] Espressif Systems, “ESP32 Series Datasheet,” version 5.3. [Online]. Available: https://documentation.espressif.com/esp32_datasheet_en.html. Accessed: 3 Oct. 2026. ESP32 Datasheet | Espressif Documentation



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)