International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org **ISSN: 3139-759X**

https://doi.org/10.68104/ijasit.v1.i4.46

Design and Simulation of an Event Driven Sensorless IoT System for Automotive Battery Monitoring and Fault Detection

Thushani Mallikarathne[1, *], Hashan Abeysinghe[2]

[1] Department of Computing, Faculty of Technology, University of Turku, Finland. *mthushani97@gmail.com*

[2] Department of Instrumentation and Automation, Faculty of Technology, University of Colombo, Sri Lanka. *hashanmadumal786@gmail.com*

*** Corresponding author.**
Email: *mthushani97@gmail.com*

Article History:
Received Date: 30 August 2026
Revised Date: 04 September 2026
Accepted Date: 06 September 2026
Published Date: 26 October 2026

Keywords:

automotive battery; event driven monitoring; fault detection; sensorless monitoring, lead acid battery

ABSTRACT

In the automobile industry, lead acid battery failure continues to be a major factor in vehicle immobility. In traditional monitoring systems, high inactive current draw, which adds to parasitic drain, or a lack of remote accessibility for real time user notification are common problems. A unique, event driven Internet of Things (IoT) framework for real time vehicle power diagnostics is proposed in this research. The sensorless architecture that uses straightforward resistor divider network to transmit voltage data straight into Analog to Digital Converter (ADC) of the microcontroller, is a significant innovation since it prevents the need for external sensors. Also, this suggested framework makes use of a state machine that alternates between an active sensing mode and a deep sleep period that is only activated by abnormal voltage occurrences. The proposed design validated through an electrical system and supervisory state logic modelled using MATLAB/Simulink, where Python serving as a secondary simulation and communication layer. Five main operating conditions are distinguished by the monitoring algorithm: parked, engine cranking, regular operation, alternator failure, and critical shutdown. The automotive voltage is scaled to an ADC compatible range using a 10 kΩ/2.7 kΩ resistor divider network, and signal filtering is provided by a 0.1 μF capacitor. With voltage settings of 12.60 V during parked operation, 9.80 V during engine cranking, 14.20 V during normal charging, 11.74 V during simulated alternator failure, and 10.30 V at critical shutdown, the simulation replicated the desired vehicle operating sequence. 1000 separate Monte Carlo runs with additive Gaussian measurement noise made up the stochastic validation. Relative mean voltage errors for the statistically assessed parked, alternator failure, and shutdown states were 0.0016%, 0.0009%, and 0.0029%, respectively, whereas the empirical detection success rate under the simulated conditions was 100%. The findings show that the suggested state machine architecture can reduce the need for ongoing high-power monitoring while differentiating between sustained abnormal voltage circumstances and normal beginning transients. As a result, the system offers a useful basis for controlled automotive validation and future ESP32 based physical implementation.

1. Introduction

Every driver has experienced the depressing sensation when the engine of the car refuses to start. Nearly every car on the road today, from entry level sedans to luxury hybrids, still need a 12V lead acid battery to start. Even with the quick emergence of high-performance electric vehicles require this type of battery. Recent market [11] [15] shows that lead acid batteries continue to be the most popular option due to their affordability, dependability, and recyclable nature. More than 280 million cars use this technology in North America alone.

However, the most common cause of vehicle immobilization is still the battery failure. We now have a new problem as our cars become software defined entities; how can we keep an eye on this crucial component without unintentionally exhausting its lifespan?

The answer is with IoT. Real time monitoring using the Internet of Things (IoT) has been the subject of much recent research. Low-cost microcontrollers, particularly the ESP8266 and ESP32, have been shown to be useful for remote health estimate by some articles [17][18]. Nonetheless, contemporary research often finds a strong observer effect. High power platforms, such as the Raspberry Pi, were used in some research papers [2][20] for vehicle state monitoring, however, these systems frequently require an extra current that is unsustainable for long term car parking. A crucial information gap is represented by these cases, in which the monitoring gadget turns into the cause of battery exhaustion. Some research articles [3][17] stated that constant broadcasts in IoT based battery management systems (BMS) may unintentionally hasten chemical degradation. In a similar vein, some cases [14] contend that always on communication protocols circumvent a car's natural power saving modes by generating a continuous parasitic load. Hence IoT solutions still have some failures in the context of energy saving.

Due to these IoT failures, industry leaders [7] and authorities [4] have pushed for a change to event driven telemetry to solve these energy limitations. These experiments offer a far higher insight per joule ratio since the gadget stays in a deep sleep state and only wakes up in response to a particular electrical anomaly. Artificial Intelligence (AI) and the Extended Kalman Filter (EKF) may increase State of Charge (SoC) accuracy, according to research that has done previously [8][19]. However, these models are frequently computationally demanding for ultra low power edge nodes. While deep learning-based defect detection is successful, it requires large scale datasets that are challenging to analyse on device, according to researchers [9][12]. This study closes this gap by suggesting a lightweight, state machine driven logic that provides high diagnostics without the computational or energy cost of conventional AI. However, robust validation is necessary due to the complexity of the automotive electrical environment, which is marked by high noise and huge transients. The significance of differentiating between normal transients (such as engine cranking) and anomalous failures (such as alternator diode leakage) is emphasized by some articles [15] [21] Although remote alerting systems are widely used, their sustainability depends on efficient power management, according to recent systematic evaluations [13] and [16].

Additionally, the foundation for contemporary embedded platforms is provided by the [1] on lightweight detectors and on hybrid BMS systems. Lastly, some researchers [5][6] predict that cloud integrated, flexible systems will be the BMS of the future. This work closes the gap between realistic IoT deployment and high-fidelity physical modelling. This study offers a translational design framework that has been verified in two different computing settings, in contrast to earlier studies that only concentrate on the hardware limitations or the software logic. This work enables the fault injection of high-risk scenarios detected [15] such as parasitic drain and charging system failure, using a virtual prototype in MATLAB R2025b. Approach uses Python 3.12 to verify real time asynchronous communication over the Telegram Bot API and MATLAB/Simulink to physically model battery transients. By building on a previous validation of Telegram Bot API for quick alerting, [10] that method applies in this research paper for alert users quickly. By putting in place an event driven state machine that stays in a 10 μA the upcoming generation of vehicle battery monitoring, our work provides a very energy efficient and scalable deep sleep state until a diagnostic threshold is exceeded.

The suggested solution uses a finite state machine architecture to overcome this restriction. Instead of being categorized only based on instantaneous magnitude, the voltage signal is interpreted based on the present operational context. PRE_START/PARKED, ENGINE_STARTING, NORMAL_RUNNING, ALTERNATOR_FAILURE, and CRITICAL_SHUTDOWN are the main states considered in this study. Because State flow offers explicit states and transitions for depicting sequential and event driven system behaviour, finite state machine modelling is especially well-suited for this application. By utilizing voltage behaviour as the main diagnostic variable, the suggested architecture also lowers hardware complexity. Before being fed to the microcontroller ADC, the automotive voltage is scaled using a resistor divider network rather than an external current sensor. To scale the roughly 0–15 V automotive input range to a level acceptable with a nominal 3.3 V microcontroller interface, a 10 kΩ/2.7 kΩ divider is chosen. For lowpass filtering and mitigation of high frequency electrical disturbances, a 0.1 μF capacitor is integrated.

A communication layer based on Telegram is used to provide remote diagnostic capabilities. The Python interface, which creates HTTP queries to the Telegram Bot API, receives the running state from MATLAB/Simulink during simulation. To prevent identical states from producing repeated notifications, a persistent state variable is employed. This communication approach helps achieve the low power design goal and minimizes needless network activity. The main goal of this project is to create and simulate validate a low power, sensorless, event driven Internet of Things framework that may minimize needless energy use while identifying important automotive battery and charging system conditions. The study specifically assesses whether the suggested state machine logic can reliably generate remote notifications during significant state transitions, whether classification is stable under simulated measurement noise, and whether it can differentiate between typical engine start transients and abnormal charging and shutdown conditions.

The novel of this study is the integration of four complementary design elements into a single simulation validated architecture, (i) voltage only sensorless acquisition without a dedicated current sensor; (ii) context-aware finite state machine (FSM) classification that differentiates between normal transient events and persistent abnormal conditions; (iii) event driven low-power operation meant to reduce needless sensing and communication during vehicle inactivity; and (iv) state-change-based remote notification that suppresses repeated transmission of unchanged operating conditions. The suggested method focuses on system level integration of lightweight sensing, contextual defect classification, power-aware operation, and remote IoT notification for automotive battery and charging-system monitoring rather than developing a new battery State-of-Health (SoH) estimation technique. So the contribution of this study is therefore not the development of a complete automotive battery state of health estimator, but the development and validation of a lightweight event driven fault monitoring architecture that can serve as a foundation for future embedded implementation. The work establishes a bridge between simulation-based algorithm development and subsequent physical ESP32 based automotive testing.

2. Methodology

2.1 Research Design

The suggested vehicle battery monitoring framework was assessed using a quantitative, simulation based experimental approach. Four main components are integrated into the methodology:

1. an electrical model that depicts the behaviour of batteries and alternators;
2. a supervisory finite state machine that classifies working stages;
3. a low power management approach based on sleep and wakeup states; and
4. an Internet of Things communication layer for remote notification

Instead of using actual vehicles, the research was first carried out in a virtual setting. This was chosen so that the diagnostic procedure could be tested deterministically prior to hardware implementation. The behaviour of the suggested state machine can be assessed independently of vehicle specific hardware variation thanks to simulation's controlled operating conditions. Three functional layers comprised the entire system.

The battery, alternator, ignition transition, and voltage signal are all part of the physical electrical domain, which is represented by the first layer. The observed voltage is categorized into predetermined operational states in the second layer, which stands for supervisory intelligence. State changes are translated into user facing diagnostic alerts at the third layer, which stands for remote communication. The planned end to end monitoring sequence is preserved while the diagnostic algorithm can be assessed separately from the communication method thanks to this layered structure.

2.2 Simulation Environment

The primary modelling environment was MATLAB/Simulink. The interaction between the alternator and lead acid battery is represented by the electrical subsystem, and State flow supported state machine principles were used to create the supervisory logic. State flow is suitable for capturing vehicle operating modes and fault management logic since it was created for modelling reactive systems using operating states and transitions. The secondary simulation and communication environment was Python 3.12.10. In addition to providing the interface between the remote notification service and the simulated diagnostic condition, the Python layer replicates the vehicle's operational sequence. The Telegram Bot API HTTP POST requests were created using the Requests library.

As shown in Fig.1. a battery supply, an alternator charging source, and a voltage-controlled switch that represents the ignition/engine transition are all part of the simulated physical system. When the car is parked, the battery sustains the vehicle supply voltage while the charging source is disconnected. The model represents starter motor loads by introducing a brief voltage drop during engine activation. The alternator activates and raises the system voltage once the engine begins. The supervisory classification logic receives the voltage waveform that results. The simulated voltage behaviour is monitored and visualized via a parallel signal route.

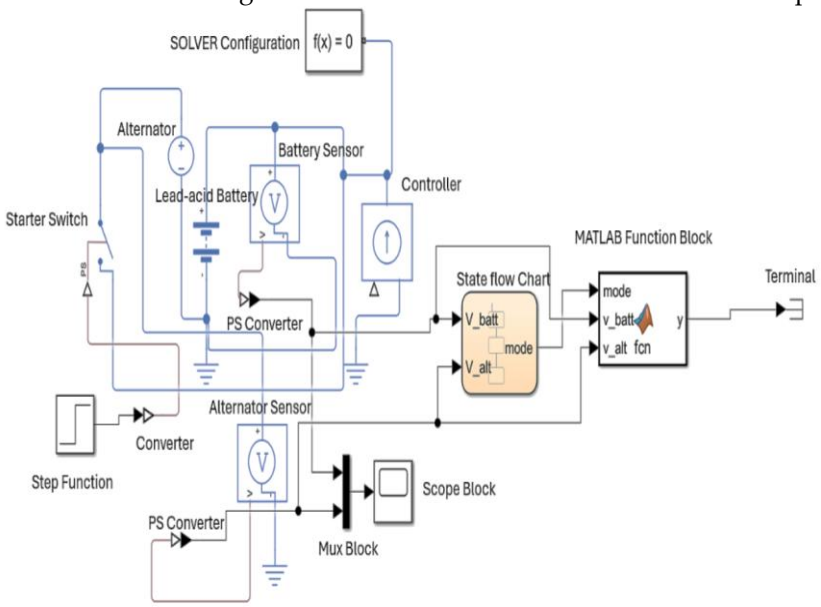


Fig. 1. MATLAB and Simscape Electrical Experimental Setup

2.3 Operating States and Thresholds

Instead of evaluating each sample separately, the FSM employs context-dependent voltage regions. The chosen set points are not universal diagnostic limitations for all vehicles; rather, they represent proof-of-concept simulation parameters. Their magnitude is in line with typical 12 V lead-acid behaviour, an engine-running charging voltage is usually above 13 V and frequently in the 13.7–14.7 V range; a charged resting battery is usually around 12.6 V or higher; and the cranking/load-test minimum voltage depends on temperature and is typically near the 9–10 V region. While the precise decision bounds must be adjusted for battery chemistry, temperature, vehicle architecture, and manufacturer-specific charging techniques, these ranges support the physical plausibility of the simulated set points.

The main states that are simulated are:

1. ALTERNATOR_FAILURE: insufficient post start charging voltage;
2. CRITICAL_SHUTDOWN: severe undervoltage indicating a potentially imminent loss of electrical availability;
3. NORMAL_RUNNING: alternator operating and maintaining the electrical system voltage;
4. ENGINE_STARTING: temporary voltage reduction caused by starter motor demand; and
5. PARKED/PRE_START: vehicle stationary with the battery providing the vehicle's standby load.

Therefore, rather than being generally applicable automotive requirements, the thresholds in this study should be seen as simulation parameters chosen for proof-of-concept validation as in Table1.

Table 1
Operating States and Simulation Parameters

Operating Condition	Nominal/target voltage	Diagnostic Interpretation
Parked/PRE_START	12.60V	Health standby condition
Engine starting	9.80V	Temporary cranking transient
Normal running	14.20V	Alternator supporter charging

Alternator failure	11.74V	Insufficient post-start charging
Critical shutdown	10.30V	Severe undervoltage

2.4 Sensorless Voltage Acquisition

Instead of using a specific external current sensor, the suggested architecture uses the battery/vehicle system voltage to obtain the main diagnostic data. Prior to ADC measurement, the automotive voltage is attenuated using a resistor divider network made up of 10 kΩ and 2.7 kΩ resistors. The output of the divider is provided by Eq. (1),

$$Vadc = Vbat \frac{R2}{R1+R2} \quad (1)$$

Where, R1 = 10 kΩ and R2 = 2.7 kΩ. Therefore, Vadc = 0.2126Vbat

The theoretical divider output for a 15 V automotive input is roughly 3.19 V, below the interface level of 3.3 V. As a result, the division offers a straightforward way to transform the automotive voltage range into a signal that is compatible with ADCs. The measurement network's continuous current is additionally decreased by the comparatively high resistor values. Although the final hardware design must consider ADC input characteristics, sampling configuration, resistor tolerance, protective circuitry, and the total parasitic current budget, the divider current at 15 V is roughly 1.18 mA. As a filtering element, a 0.1 μF capacitor is connected. Using the divider's approximation High frequency disturbances are attenuated by Thevenin resistance, which results in first order lowpass behaviour. Because automotive electrical noise is not always Gaussian or stationary, the filter properties should be confirmed experimentally in the actual implementation.

2.5 Power Management

Three operational conditions; SLEEP, WAKEUP, and ACTIVE, are included in the suggested embedded design. The ESP32 is designed to stay in deep sleep mode during regular parked operation and only switch to active mode when a pertinent electrical event needs to be verified and categorized. When the controller is turned on, it takes a sample of the battery voltage, runs the finite state machine (FSM), and sends out a notification if it detects a major state transition. The controller reverts to the low power state following communication and event verification. The ESP32 MCU level deep sleep design target represented by the roughly 10 μA number taken into consideration in this study should not be regarded as observed total system standby consumption. The voltage regulator quiescent current, resistor divider network, ADC and signal conditioning circuitry, wake up circuitry, protective components, and wireless communication hardware would all contribute extra current in a physical implementation. Specifically, unless it is switched or otherwise modified for ultra low power operation, the proposed 10 kΩ/2.7 kΩ voltage divider network would introduce a continuous current route.

As a result, the MCU deep sleep figure may be significantly lower than the actual system level standby current. The following is the anticipated event driven operation sequence:

Deep Sleep → Voltage Verification → Event Detection → Wakeup → State Classification → Notification → Deep Sleep

This architecture aims to reduce unnecessary sensing, processing, and communication when the vehicle is not in use. However, no measured system level power consumption or % energy savings is claimed because neither the full ESP32-based prototype nor a traditional always on baseline have been physically tested in the current investigation. Future hardware validation will be used to characterize sleep, wake up, processing, and communication currents experimentally.

2.6 State machine Decision Logic

A context aware finite state machine serves as the main diagnostic tool. The current operational state has no bearing on how the classifier interprets the immediate voltage. It is anticipated that the battery would stay close to the nominal standby voltage while in the PARKED condition. A significant brief drop in voltage is linked to a shift into ENGINE_STARTING. As a result, the voltage can drop well below the 11.9 V threshold without being instantly read as a permanent battery failure. The system anticipates that the voltage will rebound and surpass the charging threshold after the engine is started. Thus, the NORMAL_RUNNING state is produced at a voltage of about 14.2 V. The state machine can identify a situation as ALTERNATOR_FAILURE

if the engine has started but the voltage is still below the necessary charging area. CRITICAL_SHUTDOWN is the result of a further reduction into the severe undervoltage area. Because it distinguishes between transient occurrences and permanent defects, this context aware design is crucial.

2.7 IoT Integration

During the simulation, Python was used to create the communication layer. The Python interface receives both the corresponding voltage value and the diagnostic condition produced by MATLAB/Simulink. The diagnostic message is included in an HTTP request that is created by the Python communication process and sent to the Telegram communication service. Telegram's messaging interfaces allow for programmatic message transmission, which allows the simulated system to replicate remote user notifications. The most recent communicated operating state is stored in a persistent variable called last_mode. Only when the current state is different from the stored state is a new notification generated. Repeated messages during persistent faults are prevented by this method. For instance, the user receives a single fault message instead of a constant stream of similar alerts if the system stays in ALTERNATOR_FAILURE for multiple consecutive sampling cycles. Thus, the order of communication is:

State Transition → Event Verification → Message Creation → Transmission via Telegram → Last Mode Update

Depending on the needs for vehicle communication, the ESP32's built in wireless capability or a specialized cellular module like the SIM800L can take the place of the Python layer for future physical deployment.

2.8 IoT Integration

A consistent baseline model of measurement uncertainty was developed by adding additive white Gaussian noise (AWGN) with zero mean and $\sigma = 0.05$ V to the simulated voltage signal. Although AWGN is not a comprehensive model of an automotive electrical environment, it is helpful for controlled stochastic validation. Inductive switching spikes, alternator diode ripple, ignition related interference, load dump events, and non-Gaussian impulse disturbances are all possible in real cars. While the FSM uses operating context and persistent state transitions to reduce sensitivity to isolated threshold crossings, the 0.1 μF capacitor is anticipated to mitigate some of the high frequency content. However, in the current work, neither the capacitor nor the FSM have been physically verified against standardized automotive transients. Consequently, resilient Gaussian performance should be viewed as robustness. Transient testing and non-Gaussian EMI are specifically set aside for hardware validation.

2.9 Noise Modelling

Additive white Gaussian noise (AWGN) was added to the simulated voltage signal to assess the classification framework's resilience in the face of measurement uncertainty. The noise was simulated as a zero mean Gaussian process with a standard deviation of 0.05 V. In a simplified controlled setting, the ensuing stochastic fluctuation was meant to simulate electrical disturbance and measurement uncertainty. Gaussian noise offers a repeatable way to assess classifier stability. It should be noted, though, that no Gaussian disturbances from relay switching, inductive loads, alternator ripple, ignition events, and other electrical subsystems may be present in real automobile electrical environments. As a result, the AWGN model is not a comprehensive depiction of vehicular electromagnetic interference, but rather a controlled robustness test.

2.10 Monte Carlo Validation

For PARKED, ALTERNATOR_FAILURE, and CRITICAL_SHUTDOWN, a Monte Carlo experiment with 1000 separate simulation cycles was conducted. Nominal voltage, simulated mean, absolute deviation, relative mean error, 95% confidence interval, and empirical detection success were calculated for every evaluated state.

The calculation of the relative mean voltage error was calculated using Eq. 2,

$$Relative\ Error\ (\%) = 100 \times \frac{|\,\mathrm{Vsim} - \mathrm{Vnom}\,|}{Vnm} \quad (2)$$

Where, V sim is the simulated mean voltage and V nom is the normal voltage. The distribution of simulated voltages was used to compute the 95% confidence interval. Because the stored aggregate Monte Carlo result lacks a class-by-class confusion matrix and a dedicated negative event dataset, precision, recall, F1-score, and false positive rate are not supplied. It would be inaccurate to report such metrics based on the current aggregate success value. As a result, they are determined to be necessary metrics for the multi class hardware validation that follows.

3. Results

3.1 Simulation Results

The anticipated automobile operating sequence was effectively replicated by the suggested simulation. The battery voltage was kept at roughly 12.60 V during the parked condition Fig.2. (a), which is the nominal healthy standby condition chosen for the experiment. The voltage dropped to about 9.80 V when the engine start event was triggered Fig.2. (b). This indicates a decrease in:

((12.60 – 9.80) / 12.60) x 100 = 22.22% with relation to the parked voltage nominal.

Because the reduction happened inside the anticipated starting context, it was regarded as an engine cranking event rather than an imminent battery failure scenario. The simulated alternator raised the electrical system voltage to about 14.20 V once the engine was started Fig.2. (c). This indicates an increase of in comparison to the nominal parked voltage of 12.60 V Fig.2. (d).

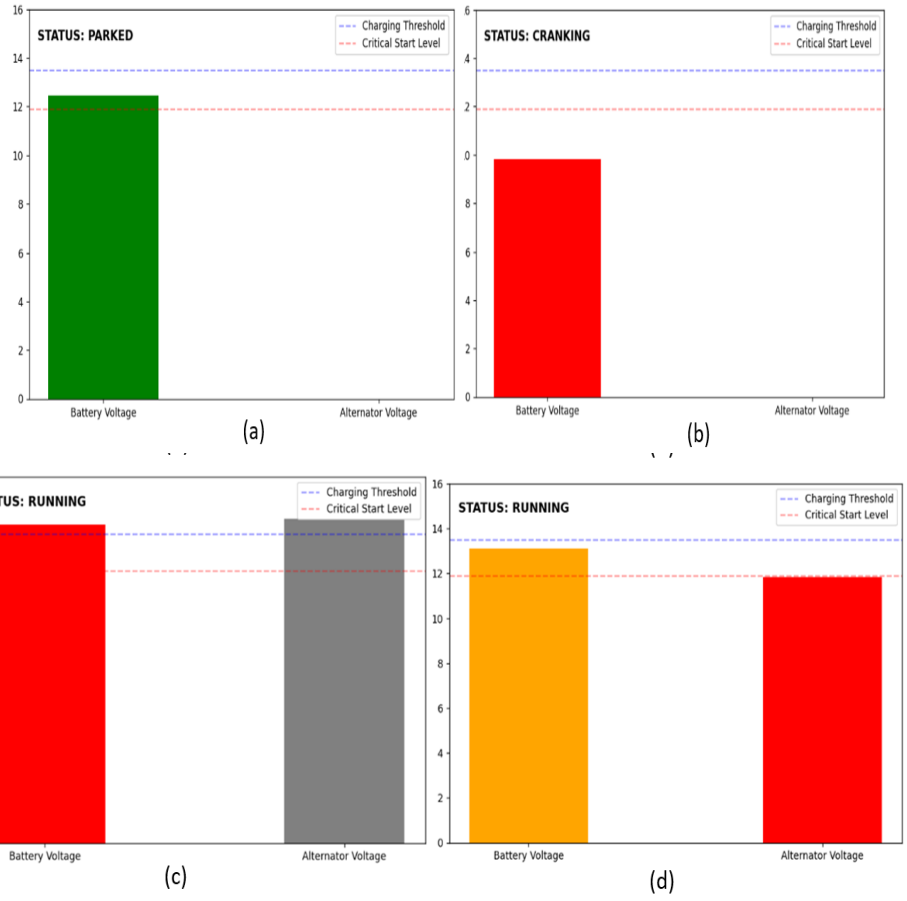
((14.20 – 12.60) / 12.60) x 100 = 12.70%

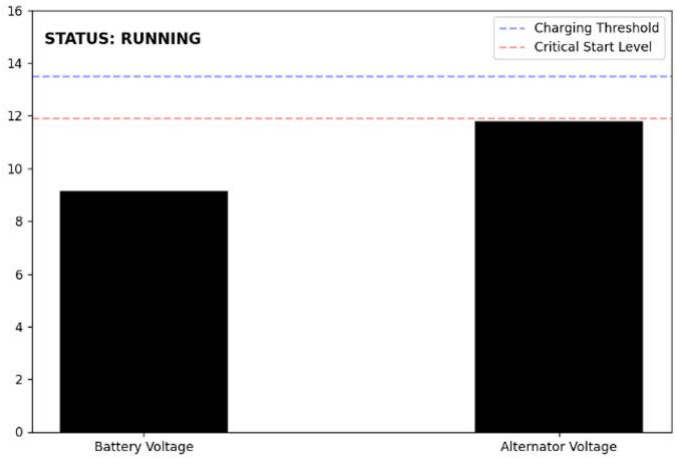
The resulting voltage was categorized as NORMAL_RUNNING since it was higher than the 13.5 V charging threshold. The electrical system voltage was then lowered to 11.74 V to simulate the alternator failure scenario Fig.2. (e). In relation to the typical operating voltage of 14.20 V, this means:

((14.20 – 11.74) / 14.20) x 100 = 17.32% reduction.

This persistent low voltage condition was identified by the state machine as ALTERNATOR_FAILURE, and a high-priority diagnostic event was produced. Ultimately, the voltage dropped to 10.30 V under the critical shutdown condition. This indicates the following in relation to the nominal parked voltage of 12.60 V:

((12.60 – 10.30) / 12.60) x 100 = 18.25% reduction. The state was categorized as CRITICAL_SHUTDOWN.





(e)

Fig. 2. Simulation Graphs. (a) Parking stage (b)Cranking stage (c)Running Stage (d) Alternator failure (e) System shutdown

3.2 Operating Sequence Analysis

The desired order of functioning states was replicated in the simulated lifespan. As in Figure 3, in PARKED/PRE_START, the vehicle system voltage was 12.60 V; during ENGINE_STARTING, it dropped to 9.80 V; during NORMAL_RUNNING, it increased to 14.20 V; during the simulated ALTERNATOR_FAILURE, it plummeted to 11.74 V; and during CRITICAL_SHUTDOWN, it reached 10.30 V.

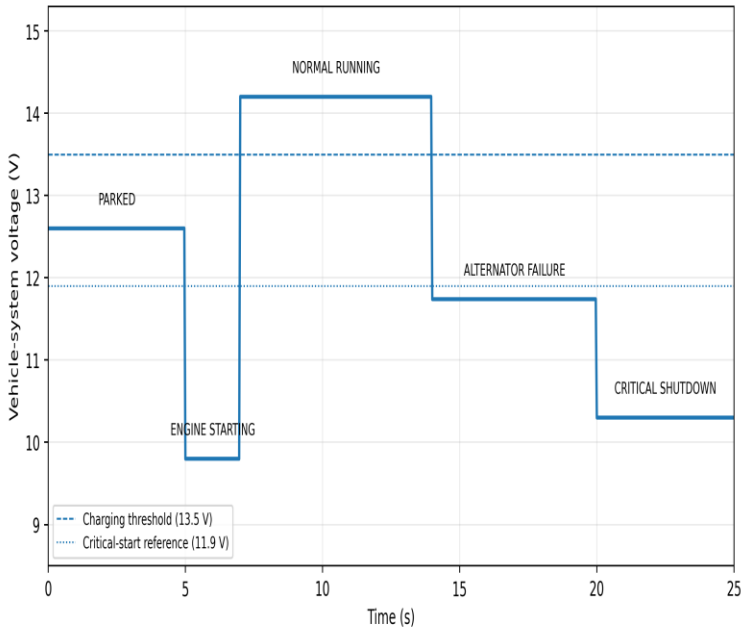


Fig. 3. Continuous Voltage vs. Time Representation for Operating Sequence 3.3 Simulation Results Summary

Table 2
Simulation Experiment Results

State	Simulated Voltage (V)	Reference Voltage (V)	Relative Change	Classification
Parked	12.60	12.60	0.00%	Healthy/PRE_START
Engine Starting	9.80	12.60	22.22%	ENGINE_STARTING

Normal Running	14.20	12.60	+12.70%	NORMAL_RUNNING
Alternator Failure	11.74	14.20	17.32%	ALTERNATOR_FAILURE
Critical Shutdown	10.30	12.60	18.25%	CRITICAL_SHUTDOWN

The deterministic results in Table 2. show that when the temporal operating context is taken into account, the suggested classifier can detect both significant transient voltage changes and chronic low voltage circumstances.

3.4 Remote Notifications

The communication layer used the event driven procedure shown in Fig. 4 to translate important state changes into remote diagnostic signals. When a relevant state shift takes place, the Python communication layer receives the classified operational status from MATLAB/Simulink and sends an HTTP POST request to the Telegram Bot API. The communication layer functions as an event driven diagnostic interface instead of a continuous telemetry stream because the persistent last_mode variable prohibits repetitive transmission of an unaltered condition. Fig. 5 shows examples of notifications issued during the simulated operation sequence, such as HOME CHECK, ENGINE STARTED, ALTERNATOR FAILURE, and SYSTEM SHUTDOWN messages.

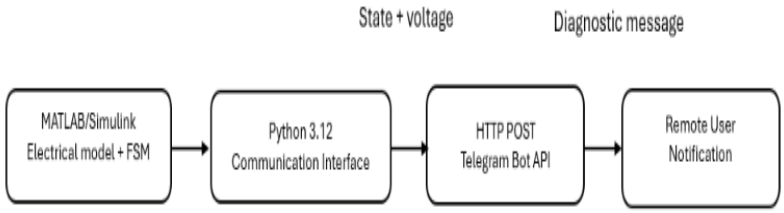


Fig. 4. Event driven remote notification workflow

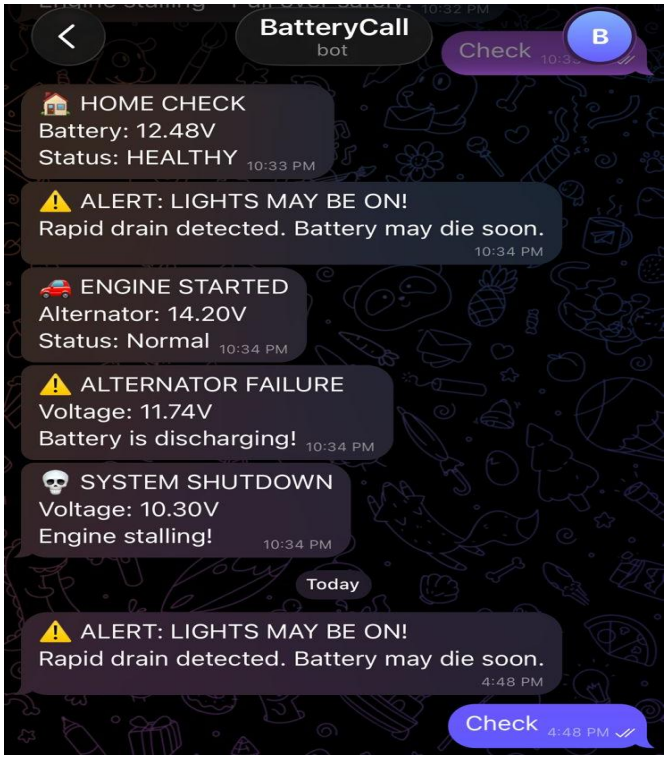


Fig. 5. Example Telegram BOT output

3.5 Monte Carlo Validation

Over the course of the 1000 simulation attempts, the Monte Carlo analysis yielded extremely reliable voltage estimates as shown in Table 3. The simulated mean for the PARKED situation was 12.4802 V, whereas the nominal voltage was 12.48 V. As a result, the relative mean error was roughly 0.0016%, and the absolute deviation was only 0.0002 V. 12.4771–12.4833 V was the matching 95% confidence interval. The simulated mean for the ALTERNATOR_FAILURE condition was 11.7401 V, but the nominal voltage was 11.74 V. The 95% confidence interval was 11.7370–11.7432 V, and the absolute deviation was 0.0001 V, resulting in a relative mean error of roughly 0.0009%. The simulated mean for the CRITICAL_SHUTDOWN condition was 10.2997 V, but the nominal voltage was 10.30 V. The relative mean inaccuracy was roughly 0.0029%, with an absolute deviation of 0.0003 V. 10.2966–10.3028 V was the 95% confidence interval.

Table 3

Results of Monte Carlo Validation

State	Nominal Voltage	Simulated Mean	Relative Mean Error	95% Confidence Interval (V)	Detection Success
Parked	12.48	12.4802	0.0016%	12.477112.4833	100%
Alternator Failure	11.74	11.7401	0.0009%	11.737011.7432	100%
Critical Shutdown	10.30	10.2997	0.0029%	10.296610.3028	100%

Among the three statistically assessed states, the highest relative mean error was roughly 0.0029%. Thus, under the implemented noise model, the findings show a very close agreement between the simulated and nominal mean voltages. Around their matching nominal values, the 95% confidence intervals stayed close together. For the parked, alternator failure, and shutdown scenarios, the relative confidence interval halfwidths were roughly 0.025%, 0.027%, and 0.030%, respectively. For the key failure states assessed throughout the course of the 1000 Monte Carlo cycles, the empirical detection success rate was 100%. However, it is crucial to note that this 100% result does not guaranty a field detection rate; rather, it indicates the observed performance under the given simulated settings.

4. Discussion

The simulation findings show that the suggested state machine design can differentiate between prolonged abnormal situations and typical automotive voltage transients. The voltage dropped from 12.60 V to 9.80 V (a 22.22% reduction) during engine cranking, yet this transient was appropriately categorized as ENGINE_STARTING rather than battery failure. The voltage rose to 14.20 V when the engine was started, indicating regular alternator supported functioning. On the other hand, the voltage dropped to 11.74 V during the simulated alternator failure condition, which was accurately classified as ALTERNATOR_FAILURE. At 10.30 V, the critical shutdown status was identified. Additionally, stable categorization under the applied noise levels was shown by the Monte Carlo analysis. The assessed states attained a 100% empirical detection success rate across 1000 simulation cycles, with relative mean voltage errors less than 0.003%. These findings show that the suggested state machine logic was able to withstand the simulated Gaussian measurement noise. Nevertheless, this outcome is simulation based and does not ensure comparable performance in the event of actual automobile electrical disruptions. By enabling the monitoring controller to stay in deep sleep during idle times and only wake up in response to a big electrical event, the event driven architecture further supports the low power goal. Rather than being a measured total system consumption, the roughly 10 μA value is regarded as an MCU level design target. A percentage energy savings is not claimed because no traditional always on baseline was measured experimentally. By doing away with the requirement for a specific current sensor, the sensorless voltage monitoring technique lowers hardware complexity. Nevertheless, voltage only monitoring is unable to accurately describe the health of batteries.

Therefore, future advances should take temperature, battery aging, internal resistance, and load factors into account. State transitions were effectively translated into remote diagnostic notifications via the Telegram communication layer, and repetitive messages for unchanged situations were avoided using the persistent last_mode variable. However, more research is needed to assess network availability and communication latency in physical implementations. Overall, the findings confirm that the suggested architecture is viable as a

low power voltage-based condition monitoring system. Event detection and fault classification are its key current contributions; more long-term experimental data is needed for thorough predictive maintenance and battery health estimate.

4.1 Limitations of the Study

There are no claims for ESP32 or in car measurements, the current validation is simulation based. Although the AWGN model offers a controlled robustness test, it is unable to accurately capture non-Gaussian automotive disturbances including electromagnetic interference, switching transients, alternator ripple, and inductive events. Additionally, the roughly 10 μA deep sleep value is an MCU level design target rather than measured total system standby consumption, practical deployment would also be impacted by protection circuitry, wiring effects, ADC non linearity, resistor divider parasitic current, voltage regulator quiescent current, component tolerances and temperature drift, and protection circuitry.

The study's fixed voltage thresholds are proof-of-concept parameters that would need to be calibrated for each vehicle and temperature before being employed in a real-world setting. Lastly, precision, recall, F1 score, and false positive rate are not presented because the available simulation dataset was not created as a balanced multi class classification benchmark, even if the Monte Carlo analysis assesses detection stability under the applied noise model. These elements serve as crucial benchmarks for further hardware certification.

5. Conclusions

A low power, sensorless, event driven Internet of Things architecture for monitoring car batteries and charging systems was created and simulation validated in this study. Voltage divider sensing, deep sleep operation, stochastic validation, finite state machine classification, and remote Telegram notification are all combined in the suggested architecture. Five typical operating conditions, parked, engine starting, normal running, alternator failure, and critical shutdown were satisfactorily replicated by the simulation. 12.60 V, 9.80 V, 14.20 V, 11.74 V, and 10.30 V were the corresponding voltage levels. The state machine architecture was able to differentiate between persistent charging and undervoltage errors, and the transient cranking voltage drop. For the assessed failure states, Monte Carlo validation with 1000 simulation cycles yielded an empirical detection success rate of 100%, and the relative mean voltage errors stayed below 0.003%. Strong numerical consistency is shown by these results under the given simulation noise circumstances. With a roughly 10 μ reduction in needless processing and transmission, the event driven architecture offers a potential method. The goal MCU level consumption is thought to be a deep sleep current. The overall system energy savings, however, have not yet been measured experimentally. Overall, the study shows that a lightweight, remotely accessible automotive battery monitoring framework is feasible and lays the groundwork for future physical ESP32 implementation and in car certification. The design will be implemented on ESP32 hardware in future development, and total current in processing, communication, wake up, and sleep modes will be measured against an always on baseline. The prototype will be calibrated for temperature and ADC non-linearity using divider components that are 1% or better. Prior to in vehicle testing, controlled laboratory testing will introduce alternator ripple, switching spikes, impulse disturbances, and representative load dump/EMI situations. To calculate precision, recall, F1-score, false positive rate, and detection delay, trial-level ground truth labels will be recorded. Further research could investigate temperature correction, CAN-bus context, switched/high impedance voltage detection, adaptive thresholds, and long-term battery health estimation.

References

[1]. Arabi, S., E. Anyanwu, and M. Haider. 2020. “A Lightweight Fault Detector for Automotive LeadAcid Batteries.” In 2020 IEEE International Conference on Electro Information Technology (EIT). IEEE. https://doi.org/10.1109/EIT48999.2020.9204712.

[2]. BinMasoud, A., and Q. Cheng. 2025. “Design of an IoTBased Vehicle State Monitoring System Using Raspberry Pi.” Western Sydney University Research. https://www.semanticscholar.org/paper/DesignofanIoTbasedVehicleStateMonitoringPiBinMasoudCheng/3c8b9932516bb32cc9e29dce660761737fd4a683.

[3]. Chen, X., and A. Gozuoglu. 2024. "Real Time Monitoring System for LithiumIon Cell Using IoT." In Proceedings of the International Conference on Smart Cities and Systems (SCS). https://www.scitepress.org/publishedPapers/2024/135795/pdf/index.html.

[4]. ELINK Communication Technology. 2026. "EventDriven Telemetry vs. Periodic Heartbeats for Asset Tracking." Accessed May 26, 2026. https://www.eelinktech.com/blog/eventdriventelemetryassettrackinghardware/.

[5]. Gupta, M., and R. Saini. 2025. "Wireless BMS Monitoring and Automated Alarm Systems in Smart Vehicles." In Lecture Notes in Electrical Engineering. Singapore: Springer. https://doi.org/10.1007/9789819919122_45.

[6]. IndustryARC. 2025. "Automotive Battery Management System Market Forecast (2025–2030)." Market Research Report. Hyderabad, India. Accessed May 26, 2026. https://www.industryarc.com/Report/1270/batterymanagementsystemsmarketanalysisreport.html.

[7]. Keysight Technologies. 2025. "Battery Drain Analysis for Low Power IoT Devices." Application Note 59921765. Santa Rosa, CA. https://www.keysight.com/us/en/assets/701805369/applicationnotes/59921765.pdf.

[8]. Kulkarni, S., R. Kumar, and A. Singh. 2025. "Advanced Battery Diagnostics for Electric Vehicles Using CANBased BMS Data with EKF." Sensors 25 (2): 142–159. https://doi.org/10.3390/s25020142.

[9]. Lipu, M. S. H., M. S. Miah, and M. A. Hannan. 2023. "Realistic Fault Detection of LiIon Battery via Dynamical Deep Learning." Nature Communications 14 (1): 7247. https://doi.org/10.1038/s41467023412265.

[10]. T. Mallikarathne, H. Abeysinghe, C. Rathnayake and D. Gauder, "Optimization of Real Time Panel Trip Detection in Industrial Systems Using IoT and Mathematical Modeling," 2023 8th International Conference on Information Technology Research (ICITR), Colombo, Sri Lanka, 2023, pp. 1-6, doi https://doi.org/10.1109/ICITR61062.2023.10382838.

[11]. Mordor Intelligence. 2026. "LeadAcid Battery Market Size & Share Analysis—Growth Trends & Forecasts (2026–2031)." Market Research Report. Hyderabad, India. Accessed May 26, 2026. https://www.mordorintelligence.com/industryreports/leadacidbatterymarket.

[12]. Muller, T., and J. Schmidt. 2025. "Fault Detection of Li–Ion Batteries in Electric Vehicles: A Comprehensive Review." Sustainability 17 (14): 6322. https://doi.org/10.3390/su17146322.

[13]. Pradnya, P., S. Meshram, and S. Pawar. 2025. "Battery Health Monitoring and Remaining Useful Life Prediction: A Systematic Review." Batteries 11 (1): 31. https://doi.org/10.3390/batteries11010031.

[14]. Rao, K. S., P. Kumar, and M. Reddy. 2024. "Performance Analysis of IoTBased Battery Management Systems in Hybrid Vehicles." International Journal of Energy Research 48: 112–125. https://doi.org/10.1155/2024/112125.

[15]. SAE International. 2025. Battery Diagnostic Terminology and Testing for Automotive Applications. SAE Standard J2982. Accessed May 26, 2026. https://www.sae.org/standards/content/j2982.

[16]. Sharma, R., and A. Verma. 2025. "IoTBased BMS for Remote Battery Health Monitoring and Alerting Systems." International Journal of Innovative Science and Research Technology 10. https://www.ijisrt.com/.

[17]. Syafii, S., A. Ahmed, and R. Kurniawan. 2025. "IoTBased Battery Monitoring System for Electric Vehicle." Journal of Mechatronics, Electrical Power, and Vehicular Technology 16 (1). https://restpublisher.com/wpcontent/uploads/2025/04/BatteryMoniteringSystemforLithiumIonBatteries UsingIOT.pdf.

[18]. Vekhande, S., and S. Makse. 2024. "IoT Based Battery Monitoring System Using ESP32." International Journal for Research in Applied Science and Engineering Technology (IJRASET) 12 (1). https://doi.org/10.22214/ijraset.2024.

[19]. Wang, L., Y. Zhang, and H. Liu. 2026. "NextGeneration State of Charge Estimation Using Hybrid Neural Networks." Journal of Energy Storage. https://doi.org/10.1016/j.est.2026.

[20]. Wibowo, L., R. Wahyusari, T. Yuwono, and A. Shofia. 2024. "IoTBased HighAccuracy Monitoring System for OnGrid Photovoltaic Power System Using NodeMCU ESP8266 and PZEM004T." Journal of

Mechatronics, Electrical Power, and Vehicular Technology 15 (2): 230–241. https://doi.org/10.55981/j.mev.2024.823.

[21]. Zhang, L., T. Gao, and X. Wang. 2025. "RealTime Fault Diagnosis for LithiumIon and LeadAcid Batteries in Hybrid Microgrids." Energies 18 (3): 450–472. https://doi.org/10.3390/en18030450.