



An Arduino-Based Framework for Early Electrical Fault Identification in Electric Vehicles

Nikhil Meshram¹, Dharmendra Nimje², Jayesh Patode³, Rohit Raut⁴, Piyush Patil⁵, Prof. C. S. Kamble^{*6}

¹⁻⁵ Student, Guide^{*6}

Department of Electrical Engineering Priyadarshini College of Engineering, Nagpur, India

DOI : <https://doi.org/10.55248/gengpi.07.1026.3236>

ABSTRACT

Electric vehicles (EVs) rely on interconnected electrical circuits and auxiliary systems, and any failure in these components may affect vehicle availability, user convenience, and operational safety. This paper proposes a practical approach for identifying selected low-voltage electrical faults using an Arduino UNO R3 as the primary monitoring and decision-making unit. The developed prototype is designed to monitor circuits associated with fuses, relays, the 12 V accessory/cigarette-lighter charging socket, communication connections, and charging-port wiring. The controller receives input signals indicating the operating condition of each circuit and evaluates them against predefined logic representing normal operation. If an abnormal condition or inconsistency is detected, the system displays the affected circuit, helping users identify the possible source of the fault and initiate troubleshooting more efficiently. Previous research has investigated data-driven fault analysis in EV charging stations and the application of machine-learning techniques for diagnosing faults in EV drive systems [1], [2]. In contrast, the proposed system emphasizes an affordable and easy-to-understand hardware prototype for monitoring selected low-voltage circuits rather than a fully validated predictive model for the entire high-voltage traction system. This prototype also offers a basis for further development, including sensor calibration, fault-event recording, severity classification, and predictive-maintenance techniques. However, its actual detection accuracy and predictive capabilities must be established through systematic experimental testing before any performance claims can be made.

Keywords—Electric vehicle, electrical fault detection, Arduino UNO, low-voltage circuit, relay, fuse, early warning, fault diagnosis.

1. Introduction :

Electric vehicles (EVs) are becoming increasingly common in both private and commercial transportation. Their electrical systems consist of a traction battery and drive mechanism, along with low-voltage circuits responsible for control operations, accessories, charging interfaces, indicators, and communication. Although a fault in an auxiliary circuit may not immediately disable the vehicle, it can affect specific functions, make fault diagnosis more difficult, or signal a potential electrical issue. Therefore, components such as fuses, relays, connectors, and accessory sockets are suitable for developing a basic electrical fault-monitoring prototype.

Traditional fault diagnosis generally involves visual examination, manual electrical measurements, and the use of diagnostic instruments. While these techniques continue to be essential, a small electronic monitoring system can help identify the particular circuit that needs inspection. In the proposed approach, an Arduino UNO collects signals from individual fault-detection inputs and checks them against predefined expected conditions. Based on the evaluation, the corresponding status is presented on a display to support faster fault identification. The system follows a modular design, enabling individual circuits to be examined separately and allowing additional monitoring channels to be incorporated in future developments.

Automated and intelligent fault diagnosis has attracted considerable research interest. Shufian et al. explored machine-learning approaches for identifying and analysing faults in EV charging stations [1]. Similarly, Kumar et al. examined support-vector-machine-based techniques for fault diagnosis and prediction in EV drive systems using simulated electrical data [2]. These studies demonstrate the importance of measurement-based analysis and automated fault identification. However, the present work focuses on developing a cost-effective and interpretable hardware-based monitoring system for a limited set of low-voltage electrical circuits.

2. Problem Statement and Motivation

A fault in an EV auxiliary electrical circuit can be difficult to locate quickly when several circuits share a compact vehicle electrical system. A simple indicator that identifies the affected monitored circuit may reduce the time required for preliminary troubleshooting. The challenge addressed here is to

build a prototype that can acquire fault-state inputs, process them consistently, and display a readable fault message without requiring a complex computing platform.

The proposed system should be understood as a prototype for selected low-voltage circuits. It does not replace vehicle-grade battery management, high-voltage interlocks, insulation monitoring, certified protection devices, or manufacturer diagnostic equipment. Any real-vehicle integration would require appropriate electrical isolation, automotive-rated components, fault-tolerant design, and validation against relevant safety requirements.

3. Objectives

To develop an Arduino-based monitoring prototype for selected low-voltage EV electrical circuits.

To identify abnormal states associated with fuse, relay, accessory charging socket, communication connection, and charging-port-related circuits.

To provide separate and understandable fault messages through a display interface.

To establish a modular platform that can later support event logging, fault severity levels, and data-based predictive maintenance.

To evaluate the prototype through controlled fault-injection tests and record its detection performance.

4. Related Work

4.1 Machine learning for EV charging-station fault analysis

Shufian et al. presented work on automatic fault detection and analysis for EV charging stations using machine-learning techniques [1]. Their study highlights the importance of monitoring power-electronic equipment and analysing measured operating behaviour. Such data-driven methods can help distinguish normal operation from abnormal conditions, provided that representative data and a suitable validation process are available. The present prototype shares the goal of automated fault identification but uses explicit circuit-state inputs and rule-based logic, which are easier to inspect during an initial hardware demonstration.

4.2 SVM-based diagnosis of EV drive systems

Kumar et al. examined support vector machine (SVM) methods for diagnosing and predicting faults in EV drive systems [2]. The study describes the use of electrical quantities and simulated operating conditions to create data for model development. Machine-learning methods may identify patterns that are difficult to capture with simple threshold logic; however, their performance depends on data quality, fault coverage, and testing on conditions not used for training. In contrast, this paper outlines a basic embedded monitor for a limited set of low-voltage circuits.

4.3 Research gap

The two studies above address charging-station equipment and EV drive-system diagnosis, while the proposed project focuses on a low-cost, circuit-level monitor for selected auxiliary electrical connections. The key contribution is not a claim of superior predictive accuracy, but a practical and explainable prototype architecture that can be tested through deliberate fault insertion. A further gap to address is the distinction between detecting a fault after it occurs and predicting a future failure. Prediction would require time-series measurements, labelled historical examples, or a validated degradation model; it cannot be established from simple fault-state inputs alone.

5. Proposed System Architecture

The system consists of a regulated DC supply, circuit-specific sensing or fault-input interfaces, an Arduino UNO R3, a display unit, and the electrical circuits used for demonstration. Each monitored circuit is associated with a defined input. The input interface must scale, isolate, or condition signals as required so that the microcontroller pins are never exposed to unsafe voltage or current.

Power supply: provides the low-voltage DC supply required by the controller and interface circuits.

Fault-input circuits: represent the state of each monitored circuit. The final hardware design must specify whether each input is a switch, sensor output, continuity detector, or voltage-sensing circuit.

Arduino UNO R3: reads input states, applies the decision logic, and assigns a fault label.

Display unit: presents the normal state or the name of the circuit requiring inspection.

Relay and protection components: used only within their specified ratings and with appropriate driver/protection circuitry.

Figure 1. Functional block diagram (text representation): DC supply → input conditioning and circuit-state sensing → Arduino UNO R3 → display/alert. The monitored fuse, relay, accessory socket, communication link, and charging-port-related circuit feed the sensing stage. This description should be converted into a circuit-specific diagram after the final wiring and display hardware are confirmed.

6. Methodology

6.1 Define normal operating states

For every monitored circuit, the project must define what constitutes a normal state and what signal represents a fault. For example, a fuse monitor may use a properly designed continuity-sensing circuit, while a relay monitor may verify the commanded and observed contact state. These examples are design options, not confirmation that the current prototype already implements those measurements.

6.2 Signal acquisition

The conditioned input signals are connected to suitable Arduino input pins. Digital inputs can be used when the interface provides a clear logic level. Analogue measurement may be required where a voltage value is used to determine the state. Input protection, common-reference arrangements, and electrical isolation must be designed before connecting the prototype to any vehicle circuit.

6.3 Decision logic

The controller periodically reads the inputs and compares each reading with the expected state. If an input violates its defined condition, the corresponding fault flag is set. A simple rule table can be used to map each flag to a message. Debouncing, filtering, and a short confirmation interval can help avoid false indications caused by switching transients or unstable contacts.

6.4 Fault indication

The display should show either a normal-system message or the specific monitored circuit reported as abnormal. If more than one fault is present, the software should either list all active faults or cycle through them without hiding a safety-critical indication. A fault-clear condition should also be defined and tested.

6.5 Controlled testing

Testing should begin on an isolated bench setup using low-voltage circuits. Each fault is introduced one at a time, then in selected combinations. The expected indication is documented before each test, and the actual result is recorded. No performance values should be reported until these trials have been completed.

7. Fault Detection Logic

A transparent rule-based approach can be represented as follows:

Read the state of each monitored input.

Check whether the input is valid and stable.

Compare the measured state with the expected normal state.

If the state is abnormal, assign the corresponding fault identifier.

Update the display and retain the fault until the defined clear condition is met.

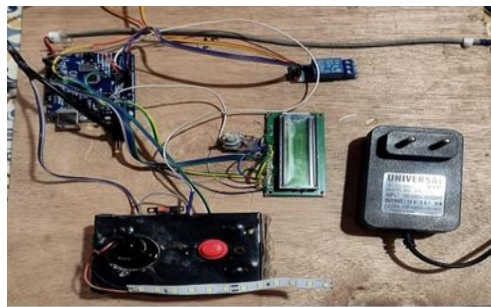
A conceptual fault table is shown below. The exact electrical sensing method and logic level must be confirmed from the implemented circuit before this table is used as a final design specification.

Monitored area	Example abnormal condition	Proposed indication
Fuse circuit	Continuity/voltage state differs from the defined healthy state	Fuse circuit fault
Relay circuit	Expected relay/contact state is not observed	Relay circuit fault
12 V accessory socket	Output voltage or circuit state is outside the defined range	Accessory socket fault
Communication connection	Expected communication/connection check fails	Communication fault

Charging-port-related circuit	Defined input or connection check fails	Charging-port circuit fault
-------------------------------	---	-----------------------------

8. Hardware Components and Design Considerations

The progress seminar material identifies an Arduino UNO R3, a 5 V single-channel relay module, a 12 V vehicle accessory socket, a glass cartridge fuse, and a DB9-to-mini-DIN communication cable among the project components. The relay module is intended for low-voltage switching and must be driven according to its specifications. A fuse rating and voltage rating must be selected for the actual circuit and its available fault current; a component's printed rating alone does not prove that it is appropriate for a particular vehicle installation.



The communication cable type alone does not establish the protocol, pinout, or compatibility of the connected equipment. These details must be checked from the equipment documentation. Likewise, a 12 V accessory socket should not be treated as a sensing device by itself; a suitable measurement or continuity interface is required if its condition is to be monitored.

The Arduino prototype is suitable for educational demonstration, but it is not automatically automotive-qualified. Direct connection to high-voltage traction circuits is outside the scope of this design. Any real-vehicle trial must be reviewed by a qualified supervisor and use correctly rated isolation and protection.

9. Evaluation Plan and Performance Metrics

The prototype should be evaluated using repeatable tests. For each monitored fault, record the number of trials, the number of correct detections, missed detections, false alarms, and the time between fault insertion and display update. Useful metrics include:

Detection rate = correctly detected fault trials / total fault trials $\times$ 100%.

False-alarm rate = normal trials incorrectly reported as faulty / total normal trials $\times$ 100%.

Indication latency = time from fault insertion to the corresponding display message.

Fault identification accuracy = correctly identified fault labels / total detected fault events $\times$ 100%.

These metrics should be calculated only after collecting test data. Since no completed numerical test dataset was provided in the project material, this paper does not claim a measured detection rate, response time, or predictive accuracy.

10. Expected Results and Discussion

The expected outcome is a working bench prototype that identifies the selected fault inputs and displays a corresponding message. The main practical benefit is a more structured initial diagnosis than an unlabelled indicator or manual trial-and-error process. Because the decision rules are explicit, the system's behaviour can be traced and demonstrated to a reviewer.

The current concept has important limits. First, a simple state monitor detects conditions represented by its inputs; it does not necessarily forecast a fault before any abnormality appears. Second, a fault may not be observable if the sensing circuit does not measure the relevant electrical quantity. Third, a small demonstration dataset cannot establish reliability under automotive temperature, vibration, electromagnetic interference, or long-term ageing. These limitations should be stated clearly in the final project report.

A future version could add voltage, current, and temperature acquisition, timestamped event logs, fault severity levels, and a communication interface. Once adequate labelled operating data are collected, statistical or machine-learning methods could be investigated for prediction. Such an extension should be evaluated against a simple baseline and tested on independent data.

11. Safety and Scope

This project is an educational low-voltage monitoring prototype. It must not be presented as a certified EV protection system or as a substitute for the vehicle's battery-management system, fuse coordination, contactors, interlocks, insulation monitoring, or manufacturer diagnostics. Testing should use an isolated bench circuit and current-limited supply wherever possible. Before any connection to an actual EV, the design requires qualified review, appropriate isolation and protection, and a documented risk assessment.

12. Conclusion

This paper has outlined an Arduino-based framework for identifying selected low-voltage electrical faults associated with EV auxiliary circuits. The approach combines circuit-state inputs, microcontroller-based decision logic, and a display that names the affected monitored circuit. Compared with data-intensive diagnostic methods, the proposed rule-based approach is relatively simple to implement and explain, making it useful as an educational prototype. However, its ability to detect faults depends on the sensing circuits and the conditions explicitly represented in the software. Predictive maintenance remains a future extension requiring time-series data and validation. The next stage is to complete circuit integration, carry out controlled fault-injection trials, calculate detection metrics, and document limitations before drawing quantitative conclusions.

References

- [1] A. Shufian, N. Hannan, M. M. H. Emon, S. Kabir, S. A. Fattah, and M. S. Hossain, "Automatic Fault Detection and Analysis of Electric Vehicle Charging Station by Machine Learning," in 2024 IEEE Region 10 Symposium (TENSYP), 2024. [Bibliographic details should be checked against the final IEEE record before submission.]
- [2] M. P. Kumar, S. Velpula, and C. Saiprakash, "Fault Diagnosis and Prediction of Electric Vehicle Drive using Support Vector Machine Algorithm," in 2025 IEEE 5th International Conference on Sustainable Energy and Future Electric Transportation (SEFET), 2025. [Bibliographic details should be checked against the final IEEE record before submission.]
- [3] International Organization for Standardization, ISO 26262:2018, Road vehicles—Functional safety, 2018. This standard is cited as general safety context; the prototype described here is not claimed to be compliant with it.
- [4] International Organization for Standardization, ISO 6469-3:2021, Electrically propelled road vehicles—Safety specifications—Part 3: Electrical safety, 2021. Consult the applicable edition and requirements before any vehicle-level implementation.
- [5] M. Ehsani, Y. Gao, S. E. Gay, and A. Emadi, Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design, 2nd ed. Boca Raton, FL, USA: CRC Press, 2009.
- [6] M. Z. Khaneghah, M. Alzayed, and H. Chaoui, "Fault Detection and Diagnosis of the Electric Motor Drive and Battery System of Electric Vehicles," Machines, vol. 11, no. 7, Art. no. 713, 2023.
- [7] Artificial Intelligence-Based Technique for Fault Detection and Diagnosis of EV Motors: A Review," IEEE Transactions on Transportation Electrification, vol. 8, no. 1, pp. 384–406, 2022, doi: 10.1109/TTE.2021.3110318.
- [8] R. Isermann, Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Berlin, Germany: Springer, 2006.