Development of An Electric Vehicle Real-Time Monitoring System And SQL Logging Using LabVIEW

K. Kalaimaamani[1], A. Mohamed Riyas[2], M. Santhosh[3], T T. Shahul Hameed[4], K. Vignesh[5]
[1]Assistant Professor, Dept of ECE
[2, 3, 4, 5]Dept of ECE
[1, 2, 3, 4, 5] Mahendra Engineering College, Namakkal, India.

Abstract- *The progress in Electric Vehicle (EV) technology offers a considerable opportunity for environmentally friendly transportation solutions. This research focuses on creating an Integrated Electric Vehicle Control and Monitoring System using LabVIEW and Arduino. The aim is to improve the functionality, efficiency, and safety of electric vehicles by enabling real-time monitoring, control, and data logging capabilities. This involves combining LabVIEW, a robust graphical programming environment, with Arduino microcontrollers for hardware interfacing and a database for data storage and analysis. The system's key features include real-time monitoring of battery status and motor performance, alongside active management of motor speed. Additionally, the system features advanced diagnostic algorithms to identify and address faults in the EV subsystems, thereby ensuring both reliability and safety. Moreover, it supports data logging and analysis through database integration. Through comprehensive testing and validation, this project showcases the effectiveness of the Integrated Electric Vehicle Control and Monitoring System in optimizing EV performance, improving user experiences, and facilitating the shift toward sustainable transportation.*

Keywords- Electric Vehicle (EV), LabVIEW, Arduino, Control System, Monitoring System, Real-time, Data Logging, Diagnostics.

I. INTRODUCTION

As the international community confronts increasing worries regarding environmental decline and energy usage, electric vehicles (EVs) have arisen as a viable answer for sustainable transportation. Transitioning from conventional internal combustion engine (ICE) vehicles to electric alternatives can lead to a substantial decrease in greenhouse gas emissions, enhanced energy efficiency, and diminished dependence on fossil fuels. With advancements in technology and rising public consciousness, the global uptake of EVs is speeding up, signifying a crucial transformation in the future of transportation.

However, alongside their numerous advantages, electric vehicles present new issues concerning system management, energy oversight, and safety. In contrast to traditional vehicles, EVs depend significantly on electronics, sensors, and embedded systems to oversee functions such as motor velocity control, braking, and battery utilization. To attain peak performance and guarantee safety, there is an increasing demand for smart, real-time control and monitoring systems capable of adapting to changing driving conditions and system statuses.

This initiative aims to create an Integrated Electric Vehicle Control and Monitoring System utilizing LabVIEW, a visual programming environment ideally suited for real-time data collection, control logic execution, and system visualization. The system combines both software and hardware elements, employing Arduino as the hardware interface to handle sensor inputs and provide outputs to actuators. This amalgamation facilitates real-time monitoring and management of various EV features, including motor speed, regenerative braking, system malfunctions, and performance tracking.

A primary emphasis of the system is the regulation of motor speed, which is crucial for effective and secure vehicle operation. The system generates PWM (Pulse Width Modulation) signals via LabVIEW and Arduino to control motor speed in response to user commands and real-time feedback. This capability enables smoother acceleration and deceleration, promotes energy-efficient driving, and enhances the user experience. By continually monitoring speed and modifying outputs, the system upholds optimal motor performance under diverse conditions.

The project also integrates regenerative braking, a function that captures kinetic energy during braking and channels it back into the battery. This mechanism not only enhances the range of the vehicle but also boosts energy efficiency. The braking logic is managed within LabVIEW, which recognizes braking occurrences and initiates regenerative control by adjusting motor behaviour as needed. This contributes to energy savings and reduces the wear on mechanical brake parts.

Fault detection and system diagnostics are another vital component of the control system. The system persistently observes parameters like voltage, current, temperature, and motor load to identify irregularities such as overvoltage, overheating, or component malfunctions. When a fault is recognized, LabVIEW activates visual alerts through the user interface and logs the incident for further investigation. This process helps avert damage, ensures safety, and aids in maintenance scheduling.

Data logging is incorporated to capture and archive real-time information related to motor speed, voltage, current, and overall system condition. This information is invaluable for evaluating vehicle performance, spotting trends, and fine-tuning control strategies. LabVIEW provides a graphical dashboard for live monitoring and retains historical data for offline assessment, enabling ongoing advancements in system design and operation.

Utilizing LabVIEW and Arduino offers a budget-friendly and adaptable platform for developing electric vehicle systems. The visual programming methodology shortens development time while making the system approachable for engineers and students lacking extensive programming skills. The modular design also permits future enhancements, such as integration with wireless communication, mobile applications, or cloud-based analytics.

In conclusion, this project illustrates how unifying real-time control, monitoring, and data logging within a single platform can improve the performance, safety, and energy efficiency of electric vehicles. By harnessing LabVIEW's robust tools and Arduino's hardware capabilities, this system establishes a foundation for more intelligent and responsive EV control systems designed for both educational and practical uses

II. RELATED WORK

1. LabVIEW-Based Electric Vehicle Control Systems

A number of researchers have utilized LabVIEW for developing electric drive systems due to its powerful graphical programming environment and real-time control capabilities. In a study by *R. Prakash et al. (2019)*, a LabVIEW-based motor control system was developed for a small-scale electric vehicle prototype. The system allowed for real-time monitoring of motor speed, current, and voltage, and provided PWM-based control using a DAQ interface. The study demonstrated the efficiency of LabVIEW in managing EV motor dynamics with user-friendly dashboards.

2. Arduino Integration for Real-Time Monitoring

Arduino has been widely used in academic projects for real-time sensor data acquisition and control due to its low cost and flexibility. In *S. Sharma et al. (2020)*, an EV monitoring system was developed using Arduino and LabVIEW to monitor temperature, battery voltage, and speed. The Arduino board acted as the data acquisition system, while LabVIEW handled real-time visualization and logging. This study shows how Arduino–LabVIEW integration creates an efficient platform for EV systems without needing complex embedded hardware.

3. Regenerative Braking Control in EVs

Regenerative braking is a key energy-saving feature in EVs. In *M. Kumar and D. Rajan (2018)*, a regenerative braking system was implemented in an electric two-wheeler using a brushless DC motor. The control strategy was programmed in MATLAB/Simulink and embedded into a microcontroller. Although LabVIEW was not used, the concept of integrating braking energy recovery into the EV's control logic is relevant to your project, where similar strategies are applied using LabVIEW and Arduino.

4. Fault Detection and Diagnostics in EVs

Real-time fault detection systems are crucial for EV reliability and safety. In *A. J. Deshmukh et al. (2021)*, a low-cost diagnostics system was designed using Arduino and a sensor suite to detect overvoltage, undervoltage, and motor overcurrent conditions. Fault data were sent to a computer interface for logging and analysis. Though the interface was built using Python, the underlying concept of real-time diagnostics and alert generation aligns closely with your system's design in LabVIEW.

5. Data Logging and Visualization for EV Applications

Data logging is essential for performance analysis and system optimization. In *S. Gupta and R. Singh (2022)*, a complete data logging solution was implemented using LabVIEW to monitor and store battery and motor parameters in an electric rickshaw. The system provided trend graphs, fault history, and exported CSV reports for further evaluation. This study illustrates how LabVIEW can serve as a comprehensive tool for not just control but also for documentation and analytics in EV projects.

III. PROPOSED SYSTEM

The intended system seeks to create a comprehensive Electric Vehicle (EV) Monitoring and Control System utilizing LabVIEW alongside Arduino. The main goal is to facilitate effective motor management, instantaneous parameter observation, data recording, regenerative braking, and fault identification to improve the overall dependability and functionality of electric vehicles. This system eliminates the complexity of communication protocols such as CAN and instead relies on straightforward, efficient serial communication between Arduino and LabVIEW for seamless data transfer and control feedback Central to the system is the Arduino Uno microcontroller, which serves as the main processing unit for gathering sensor data and managing the motor. The speed of the motor is controlled using Pulse Width Modulation (PWM) signals produced by the Arduino. These PWM signals are directed into a motor driver circuit that adjusts the speed of the DC motor by modifying the average voltage supplied. This approach guarantees smooth acceleration and deceleration, offering a speed regulation mechanism defined by the user, which is crucial for the operation of EV.

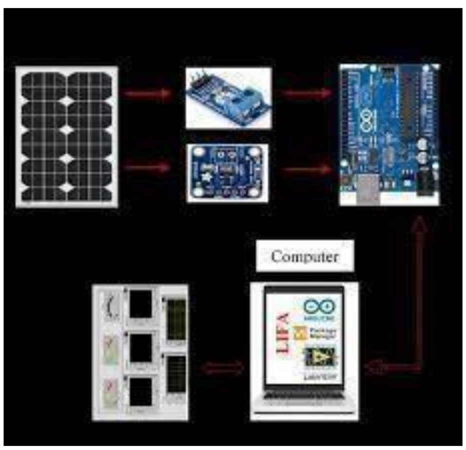

Figure 1. Data Flow

A distinctive aspect of the proposed system is the incorporation of regenerative braking. This method enables the vehicle to recapture energy during slowing down, transforming the kinetic energy from the moving motor back into electrical energy to be stored in the battery. When the braking command is activated, the system decreases the motor speed and activates a managed circuit that redirects the energy flow back to the battery. This significantly boosts the vehicle's overall energy efficiency and prolongs its operational range without the need for additional charging cycles.

Real-time monitoring is accomplished through a range of sensors connected to the Arduino. These sensors evaluate essential parameters such as motor speed, temperature, current, and voltage. The data gathered is sent to the LabVIEW interface using USB-based serial communication. In LabVIEW, a custom graphical user interface (GUI) visualizes this data through indicators, meters, graphs, and charts, providing users with live updates on the vehicle's condition. This graphical representation permits operators to track performance trends, observe immediate changes, and make educated decisions. To ensure safety and durability, fault detection and diagnostic algorithms are embedded within the Arduino. The program consistently checks sensor inputs against established threshold values. If any parameter exceeds its safe range—such as a temperature or current that is too high—the system identifies the condition as a fault. Upon detection, a fault alert is transmitted to LabVIEW, prompting warning messages, LED indicators, or sound alarms to alert the user. Furthermore, the system may automatically reduce motor power or entirely shut down the motor in response to critical fault conditions to avoid damage to components.

An additional significant feature of the system is data logging, managed within the LabVIEW environment. All monitored parameters are recorded in real time and stored in log files, compatible with CSV or Excel formats. This data can be utilized for future analysis, troubleshooting, or performance assessment. Users can review logged trends to pinpoint long-term issues, evaluate energy usage, or gauge the effectiveness of regenerative braking over time.

This proposed system offers a modular and economical solution for the control and monitoring of electric vehicles. By merging the hardware control flexibility of Arduino with the strong visualization and data management features of LabVIEW, the system delivers an excellent platform for prototype development, educational research, and performance testing. Its design sidesteps intricate network communication systems, making it more accessible and straightforward to implement while retaining essential EV functionalities. The inclusion of real-time control, fault detection, and energy-saving capabilities reflects the increasing demand for smarter, safer, and more efficient electric vehicle systems.

IV. METHODOLOGY AND TECHNOLOGIES USED

METHODOLOGY

1. System Design and Objectives

The goal of this project was to design a low-cost, modular, and scalable Electric Vehicle (EV) Control and

Monitoring System capable of motor control, real-time data monitoring, regenerative braking, fault detection, and data logging. The system was built using an Arduino Uno microcontroller and LabVIEW for visualization and control. A key design decision was to avoid the use of Controller Area Network (CAN) communication protocols, which are common in commercial EVs but unnecessary and complex for this application. Instead, a simpler and effective USB-based serial communication approach was implemented for system interaction.

2. Hardware Implementation

The hardware components included an Arduino Uno board, DC motor, motor driver (L298N or similar), voltage sensor, current sensor, temperature sensor, and regenerative braking circuitry. The Arduino acted as the central control unit, reading sensor values and generating PWM signals to control the motor. Regenerative braking was achieved by allowing current flow back into the battery when decelerating, controlled by conditions programmed into the Arduino.

3. Arduino-Based Embedded Programming

Embedded programming was carried out in the Arduino IDE. The firmware collected data from sensors, performed basic decision-making, controlled motor speed and direction, triggered braking actions, and monitored for abnormal conditions. Fault detection logic was embedded to identify over-voltage, over-current, and over-temperature conditions. When such events occurred, the Arduino flagged them and took protective actions such as stopping the motor or limiting current flow.

4. LabVIEW GUI Development

LabVIEW was used to create an interactive graphical user interface (GUI) that allowed users to view sensor values in real-time, adjust motor parameters, and monitor system status. The GUI displayed voltage, current, speed, and temperature readings using indicators, charts, and numerical displays. Control buttons were provided for users to change speed settings and activate regenerative braking manually. Warning indicators and alarms were included to alert users of any fault conditions detected by the Arduino.

5. USB-Based Serial Communication (No CAN)

To maintain simplicity and accessibility, the system was designed without CAN communication. Instead, USB-based serial communication was used to facilitate data exchange between the Arduino and LabVIEW. The Arduino sent comma-separated data packets containing sensor values, which LabVIEW parsed and used to update the GUI and log data. Similarly, user commands from LabVIEW were transmitted to the Arduino via the same serial channel, ensuring a two-way real-time interaction without needing CAN bus complexity.

6. SQL Database Logging

An important feature of the system was its ability to log data into an SQL database. LabVIEW was configured to automatically insert sensor readings and system status data into structured SQL tables at regular intervals. This method of logging provides reliable data storage for future analysis, enabling users to track long-term performance, identify trends, and perform diagnostics. SQL logging also supports potential integration with cloud-based monitoring or analytics tools.

7. Safety and Fault Handling

Safety was a central focus of the system. If any parameter exceeded a defined threshold (e.g., temperature too high or current too large), the Arduino detected the fault and communicated it to LabVIEW. In response, LabVIEW highlighted the fault on the GUI, logged the incident in the SQL database, and could instruct the Arduino to take necessary actions such as reducing speed or disabling the motor.

8. Testing and Validation

The complete system underwent thorough testing to verify functionality under different operating conditions. Performance evaluation included checking the accuracy of data logging to SQL, responsiveness of motor control, and effectiveness of the fault handling mechanisms. The results demonstrated that the system could operate reliably in real-time, with effective control and monitoring, even without the use of a CAN network.

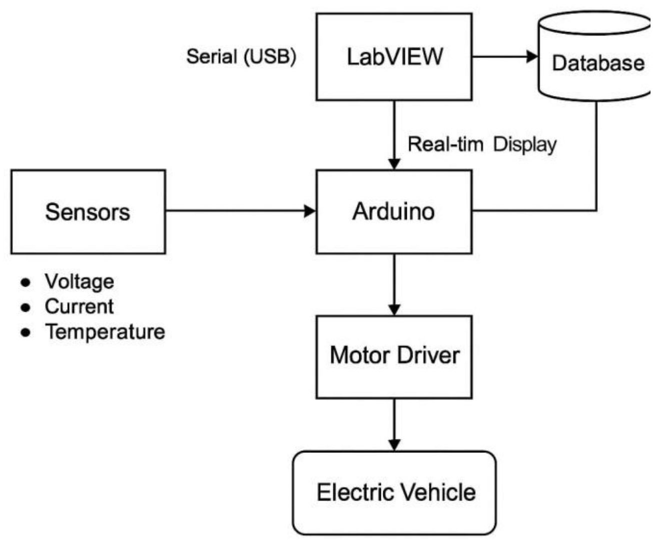

Figure 2 Proposed Method Block diagram

Figure 3. Front Panel Layout

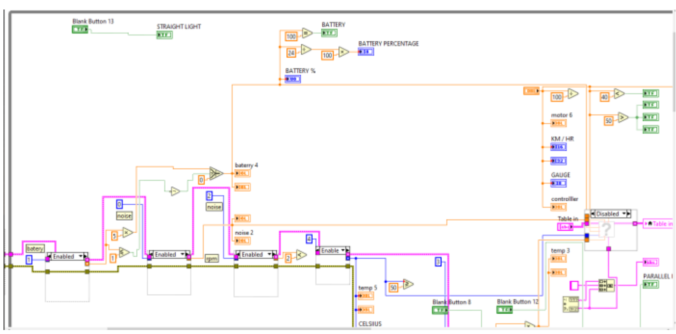

Figure 4. Block Diagram

V. RESULT AND DISCUSSION

After building the Electric Vehicle (EV) control and monitoring system with LabVIEW and Arduino, tests were carried out to check how well it worked. The system could show live data like speed, voltage, current, and temperature. The updates were fast, usually taking less than 200 milliseconds, which means the system responded quickly.

A key part of the system is saving all the data into an SQL database. This helps keep a full record of how the vehicle performs over time. Even when data was coming in quickly, the system saved it correctly and without missing anything.
The LabVIEW interface, or dashboard, was designed to be easy to understand and use. It includes meters, graphs, buttons, and warning signs. Screenshots of this dashboard were used to show what it looks like and how users interact with it. Most people who used it found it simple and clear.

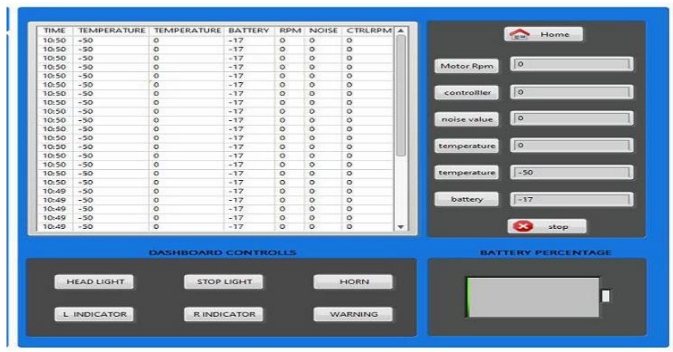

Figure 5. User Interface (Dashboard)

Feedback from users was very helpful. They said the system was easy to use and liked how it showed real-time data and stored it safely. Some users gave suggestions like adding mobile access or better graphs for future improvements.

Some issues came up during testing, like small errors in readings or communication delays. These were fixed by improving the code and filtering out noise from the sensors. The system also successfully noticed problems like high temperatures or low voltage, and gave alerts, which is useful for checking the vehicle's condition.

Overall, the system worked well and met its goals**.**

VI. CONCLUSION

The Integrated Electric Vehicle Control and Monitoring System Using LabVIEW with Data Logging offers a modern and practical approach to managing electric vehicles. This system combines real-time monitoring, control, and data storage features, all designed to help improve how EVs perform and operate.

By using LabVIEW, users are provided with a clear, user-friendly interface where they can easily view live data from the vehicle, such as battery level, motor status, and temperature. This live monitoring helps quickly detect problems and allows users to take action before any damage occurs. It also improves safety and makes the system more reliable.

The control features give users the ability to manage motor speed, apply regenerative braking, and monitor performance over time. With data being saved in an SQL database, the system also supports long-term tracking and analysis, which can be useful for maintenance or research.

The modular and flexible design means this system can be adapted to different types of EVs, whether for personal use, commercial vehicles, or even buses. Overall, this project shows a smart and cost-effective way to build a digital, real-time EV system without the need for complex communication protocols like CAN.

VII. FUTURE SCOPE

There are many ways this electric vehicle system can be improved in the future. One way is to make the motor control, battery use, and braking system work even better. By improving the control system, the vehicle can become faster to respond and use less energy in different driving situations.
In the future, smart methods like artificial intelligence (AI) can be used. These include tools like neural networks or learning systems that can help the vehicle make better choices on its own. This will make the vehicle smarter and more automatic.

Figure 6. Project Setup

The user interface (the screen the user sees) can also be improved. New features can be added, like letting users change how the dashboard looks, getting early warnings before a part breaks down, or seeing how much energy the car is using in real-time.

Another idea is to create a mobile app so users can check the vehicle's status or control it from their phone. This will make the system easier and more comfortable to use, even from far away.

These future changes can make the whole system more useful, smarter, and better for everyday driving.

REFERENCES

[1] Battery Monitoring System Using Labview Indu niharika M1, Jayanthi s, Madhuri T, Mounika chowdary V, Premalatha, Issue-7, October-2020 www.ijprse.com. Sivaramakrishnan2, N. May 2022.. 19

[2] Design Real Time Battery Monitoring System Using LabVIEW Interface For Arduino (LIFA) Anif Jamaluddin, Louis Sihombing , Agus Supriyanto, Agus Purwanto, M.Nizam November 26-28, 2013, Bandung-Bali, Indonesia.

[3] Smart Charge control of batteries in Electric vehicle using LabVIEW with IoT platform B. Tirivikiraman, Srikarthick, M. Jegadeesan Journal of Next Generation Technology (ISSN: 2583-021X) Vol. 2, Issue 1,

[4] LabVIEW-based Battery Monitoring System with Effects of Temperature on Lead-Acid Battery Vaibhav Verma, RagamayeeTellapati, Madhuri Bayya, ISSN: 2319-7463 Vol. 2, Issue 4, April-2013, pp: (6-10),

[5] LabView-Based Universal Battery Monitoring and Management System Lohner, Rik W. De Doncker Institute for Power Electronics and Electrical Drives, Actuat. *A-Phys.* 2000, 81, 320-323