iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83800

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 VI June 2026- Available at www.ijraset.com

CAN Based Battery Monitoring and Protection System with IoT Remote Dashboard for Electric Vehicle Applications

Mrunal Ravindra Dhawade[1], Prajwal Laxman Godage[2], Dipak Santosh Kulwade[3], Dr. Rohit Gawade[4]
Electronic & Telecommunication Department, KJ College of Engineering and Management Research

Abstract: ***This paper presents a Controller Area Network (CAN) based battery monitoring and protection system with an Internet-of-Things (IoT) remote dashboard, developed for electric vehicle applications and targeted at an eBAJA Electric All-Terrain Vehicle (EATV). The system is implemented as a distributed two-node architecture in which both nodes use an ESP32 microcontroller interfaced with an MCP2515 stand-alone CAN controller over SPI, operating at a 500 kbps CAN bit-rate with an 8 MHz oscillator. The transmitting node (Node 1) acquires battery pack voltage, temperature using an LM35 sensor, and current using a Hall-effect current-sensor module, computes the State of Charge (SOC) from a linear voltage mapping, assigns a fixed State of Health (SOH), and transmits the five parameters in a single 8-byte CAN frame with identifier 0x101 once per second. The receiving node (Node 2) decodes the frame and presents the data on a 20×4 I2C character LCD and on the Arduino serial monitor. The measured parameters are also published to a Blynk IoT dashboard for remote monitoring. A prototype battery of three series-connected lithium-ion cells (3.7 V, 2 Ah each, approximately 12 V pack) was used for bench validation, and a separate pulse-frequency motor-speed measurement routine was tested. Experimental outputs from the LCD, serial monitor, IoT dashboard, and a commercial smart-BMS application are presented and discussed. Available project outputs are presented and discussed qualitatively due to limited recorded datasets.***

Keywords: Battery Management System, Controller Area Network (CAN), ESP32, MCP2515, State of Charge (SOC), State of Health (SOH), IoT Dashboard, Blynk, Electric Vehicle, eBAJA EATV.

I. INTRODUCTION

Electric vehicles depend on a reliable battery system whose key parameters must be continuously observed to ensure safe operation. Real-time knowledge of pack voltage, current, temperature, and charge level is essential both for protecting the cells and for informing the driver. In off-road competition vehicles such as electric all-terrain vehicles, the monitoring electronics must be distributed across the chassis and must communicate over a robust in-vehicle network. The Controller Area Network (CAN) bus is widely used for this purpose because of its differential signalling, multi-node arbitration, and noise immunity.

This work describes a CAN-based battery monitoring and protection system with an IoT remote dashboard intended for an eBAJA Electric All-Terrain Vehicle (EATV). The system is realised as two ESP32 nodes connected through MCP2515 CAN controllers. One node senses the battery parameters and transmits them on the CAN bus, while the second node receives the data and displays it on a local LCD and the serial monitor. The same parameters are made available on a Blynk cloud dashboard for remote viewing.

A. Problem Statement

In an electric all-terrain vehicle the battery pack experiences large and rapidly varying loads, and the monitoring electronics are physically separated from the display and supervisory units. A point-to-point wiring approach does not scale and is vulnerable to electrical noise. There is therefore a need for a distributed monitoring scheme in which a sensing unit located at the battery transmits the measured parameters over a robust serial bus to a display and supervisory unit, while also exposing the data for remote monitoring. The present project addresses this need using a CAN bus link between two ESP32 nodes together with an IoT dashboard.

B. Objectives

- To acquire battery pack voltage, temperature, and current using an ESP32 and dedicated sensor modules.
- To compute State of Charge (SOC) from the measured pack voltage and to report a State of Health (SOH) value.

©IJRASET: All Rights are Reserved

- To transmit the measured and computed parameters over a CAN bus using MCP2515 controllers at 500 kbps.
- To receive and decode the CAN frame on a second ESP32 node and display the parameters on a 20×4 I2C LCD and the serial monitor.
- To publish the parameters to a Blynk IoT dashboard for remote monitoring.
- To validate the system on a prototype three-cell lithium-ion battery and in the context of an eBAJA EATV platform.
- To acquire motor-speed information through pulse-frequency measurement techniques.

The core problem is straightforward: farmers deliver goods but cannot guarantee they will be paid fairly or on time. No existing agricultural platform provides an automated, tamper-proof payment enforcement mechanism that operates without intermediaries. AgriHandshake was motivated by this precise gap — the need for a system where payment is not promised but programmatically guaranteed, where trade terms are encoded in smart contracts rather than verbal agreements, and where trust is built into the technology itself.

II. LITERATURE REVIEW

The Controller Area Network was introduced by Bosch as a serial communication protocol for distributed real-time control in vehicles, and its data-link and physical-layer behaviour is standardised in ISO 11898 [1], [2]. Its multi-master arbitration and differential signalling make it the de-facto in-vehicle network and a natural choice for distributed battery monitoring.

Battery Management Systems (BMS) for electric vehicles have been reviewed extensively in the literature. Lu et al. surveyed the key issues of lithium-ion battery management, including state estimation, balancing, and safety protection [7]. Rahimi-Eichi et al. provided an overview of BMS functions and their application in electric vehicles and smart grids, emphasising voltage, current, and temperature acquisition as the basis for state estimation [8]. Brandl et al. discussed battery and battery-management architectures for electric vehicles and the role of standardised communication interfaces between modules [9].

At the component level, the ESP32 system-on-chip provides integrated wireless connectivity and multiple analog inputs suitable for sensor acquisition [3], the MCP2515 stand-alone CAN controller adds CAN capability to a microcontroller over an SPI interface [4], the LM35 provides a linear analog temperature output [5], and Hall-effect current sensors provide an analog output proportional to the measured current [6]. Cloud IoT platforms such as Blynk allow microcontroller data to be visualised remotely through virtual datastreams [10]. The present work combines these established components into a CAN-linked, IoT-enabled battery monitoring and protection system for an electric all-terrain vehicle.

III. PROPOSED SYSTEM

The proposed system is designed as a distributed electric vehicle monitoring architecture utilizing Controller Area Network (CAN) communication to exchange information between multiple embedded nodes. The system was developed for an SAE eBAJA Electric All-Terrain Vehicle (EATV) and consists of two independent but complementary subsystems.

A. Battery Monitoring and Protection Module

The first subsystem is responsible for battery monitoring, parameter transmission, fault detection, and remote visualization. The monitored parameters include:

- Battery Voltage
- Battery Current
- Battery Temperature
- State of Charge (SOC)
- State of Health (SOH)

The measured parameters are acquired by an ESP32-based transmitting node and transmitted through the CAN bus using an MCP2515 controller. A receiving node decodes the CAN messages and displays the information on a 20×4 LCD dashboard. Simultaneously, the same data are uploaded to the Blynk IoT platform for remote monitoring.

To validate the monitoring architecture, a laboratory-scale battery pack consisting of three lithium-ion cells connected in series was developed. Each cell has a nominal voltage of 3.7 V and a capacity of 2 Ah, resulting in an approximate pack voltage of 12 V.

It is important to note that the actual SAE eBAJA battery pack rated at 72 V and 80 Ah was not directly integrated into this study. The OEM battery pack involved proprietary hardware and communication complexities, making direct experimentation impractical during the development phase.

Therefore, the prototype battery was utilized to validate the monitoring architecture and communication framework.

B. Motor Speed Monitoring Module

The second subsystem focuses on motor-speed monitoring using data obtained from the actual SAE eBAJA electric vehicle. Unlike battery testing, which employed a prototype battery pack, motor-speed measurements were obtained from the operational vehicle. Pulse signals generated by the motor-speed sensing mechanism were processed using interrupt-based counting techniques. The acquired pulse data were converted into:

- Pulses per second
- Frequency (Hz)
- Rotational speed (RPM)

The resulting measurements were displayed through the Arduino Serial Monitor and used to evaluate motor operating behavior under different speed conditions.

IV. HARDWARE ARCHITECTURE

The hardware components used in the project are listed in Table 1. Each ESP32 node is connected to an MCP2515 CAN controller through the SPI bus, with the chip-select line on GPIO5 and the SPI lines configured as SCK = GPIO18, MISO = GPIO19, and MOSI = GPIO23. The MCP2515 modules use an 8 MHz oscillator. On Node 1 the voltage-sensor output is read on GPIO32, the LM35 temperature sensor on GPIO35, and the current-sensor output on GPIO34. On Node 2 the I2C character LCD is connected at address 0x27. The prototype battery consists of three series-connected lithium-ion cells (3.7 V, 2 Ah each), giving an approximately 12 V pack.

Table 1. Hardware Components Used

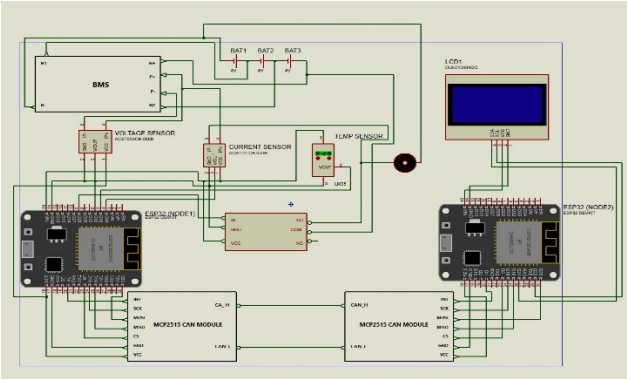
Component	Details (from project files)
Microcontroller	ESP32 (DEVKIT, two units)
CAN controller	MCP2515, SPI, CS = GPIO5, 8 MHz
CAN bit-rate	500 kbps
Temperature sensor	LM35, analog input GPIO35
Voltage sensing	Voltage-sensor input GPIO32 (×5 scaling)
Current sensing	Hall-effect current module, GPIO34 (100 mV/A, 2.5 V offset)
Display	20×4 I2C character LCD, address 0x27
Relay module	Listed in project hardware
Battery (prototype)	3 × Li-ion, 3.7 V, 2 Ah, ≈12 V pack

V. SOFTWARE ARCHITECTURE & CIRCUIT DIAGRAM

The firmware was developed in the Arduino environment. Node 1 uses the SPI and mcp2515 libraries; Node 2 additionally uses the Wire and LiquidCrystal_I2C libraries for the I2C LCD. Both nodes initialise the MCP2515 by issuing a reset, setting the bit-rate to 500 kbps with an 8 MHz crystal, and entering normal mode. The serial port runs at 115200 baud on both nodes.

Node 1 executes a periodic loop that reads the three analog channels, converts each reading, computes SOC and SOH, packs the values into a CAN frame, transmits the frame, prints the values to the serial monitor, and waits one second before repeating.

Node 2 continuously polls the MCP2515 for incoming messages; when a frame with identifier 0x101 is received, it extracts the five parameters and writes them to the LCD.



Circuit Diagram in proteus

VI. CAN COMMUNICATION FRAMEWORK

Communication between the two nodes uses the CAN protocol at a bit-rate of 500 kbps, configured through the MCP2515 controllers with an 8 MHz oscillator. The transmitting node sends a single standard data frame with identifier 0x101 and a data length code (DLC) of 8 bytes. The five measured and computed parameters are written to the first five data bytes as integer values, and the remaining three bytes are set to zero. The byte mapping is given in Table 2.

Table 2. CAN Frame Structure (ID 0x101, DLC = 8)

Data Byte	Parameter (from transmitter code)
data[0]	Battery voltage (V)
data[1]	Current (A)
data[2]	Temperature (°C)
data[3]	State of Charge, SOC (%)
data[4]	State of Health, SOH (%)
data[5]–data[7]	Reserved, set to 0

On reception, Node 2 verifies that the frame identifier equals 0x101 and reads the same byte positions back into the voltage, current, temperature, SOC, and SOH variables. Because the parameters are transmitted as single-byte integers, fractional precision is not preserved across the bus; this is reflected in the displayed values and discussed in Section below.

VII. BATTERY MONITORING AND PROTECTION LOGIC

The transmitter firmware acquires each analog channel using the ESP32 12-bit ADC (reference 3.3 V, full scale 4095) and converts the readings as follows, exactly as implemented in the source code:

vout = adc * (3.3 / 4095)
batteryVoltage = vout * 5.0
temperature = (adc * 3.3 / 4095) * 100 // LM35
current = ((adc * 3.3 / 4095) - 2.5) / 0.100

The State of Charge is computed from the measured pack voltage using a linear mapping between 9.0 V and 12.6 V, which corresponds to a three-cell lithium-ion pack, and the result is clamped to the range 0–100 %:

SOC = ((batteryVoltage - 9.0) / (12.6 - 9.0)) * 100

The State of Health is set to a fixed value of 100 % in the transmitter code, where it is explicitly marked as a placeholder. The five values are then packed into the CAN frame described in Section VI.

Regarding protection, a relay module is included in the project hardware, and an experimental output (Fig. 4) shows the LCD displaying a “HIGH TEMP! / LOAD OFF” warning, which demonstrates a temperature-fault response in which the load is disconnected.

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

VIII. IMPLEMENTATION

The measured parameters are presented on a Blynk IoT dashboard. The Blynk.Console shows a device named “BMS SYSTEM” in the “2621SH” organization, reported as Online. Five gauge widgets are bound to virtual datastreams: voltage on V0 (0–25 V), temperature on V1 (0–100 °C), current on V2 (0–25 A), SOC on V3 (0–100 %), and SOH on V4 (0–100 %). This mapping is summarised in Table 3.

Table 3. Blynk Dashboard Datastreams (from dashboard screenshot)

Datastream	Parameter	Gauge Range
V0	Voltage (V)	0 – 25
V1	Temperature (°C)	0 – 100
V2	Current (A)	0 – 25
V3	SOC (%)	0 – 100
V4	SOH (%)	0 – 100

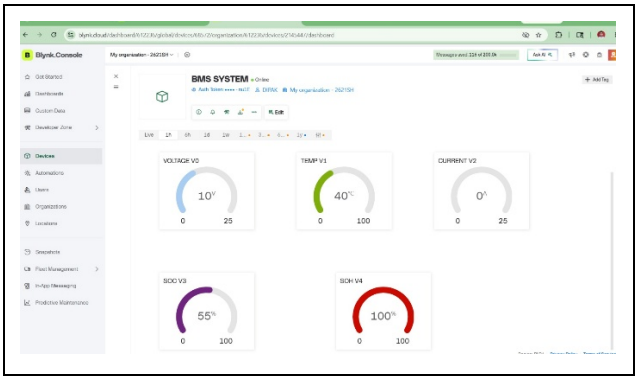


Fig. 2. Blynk IoT remote dashboard for the “BMS SYSTEM” device (V0–V4 gauges).

IX. EXPERIMENTAL SETUP

Subsystem A : 1. Two ESP32 nodes, each fitted with an MCP2515 CAN module, were connected over the CAN bus. Node 1 was wired to the voltage-sensor, LM35, and current-sensor inputs and to the prototype three-cell lithium-ion battery; Node 2 was wired to the 20×4 I2C LCD. Both nodes were monitored through the Arduino serial monitor at 115200 baud. The receiver LCD was used to display the decoded parameters in real time (Fig. 3).

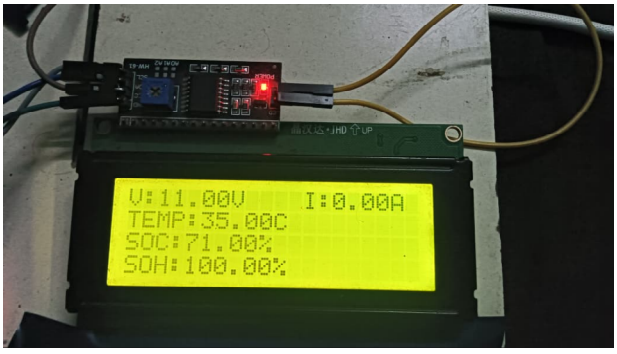


Fig. 3. Receiver-node 20×4 LCD displaying voltage, current, temperature, SOC, and SOH.

Subsytem B : In addition, a separate motor-speed measurement routine was tested. This routine configures a digital input (GPIO2) with an interrupt on the rising edge, counts the pulses accumulated over one second to obtain the pulse frequency in hertz, and converts the frequency to motor speed in RPM using the relation RPM = frequency × 13.58. The resulting frequency and RPM values were monitored through the serial interface under both low-speed and high-speed operating conditions.

These measurements were used to evaluate the responsiveness and effectiveness of the implemented speed estimation methodology.

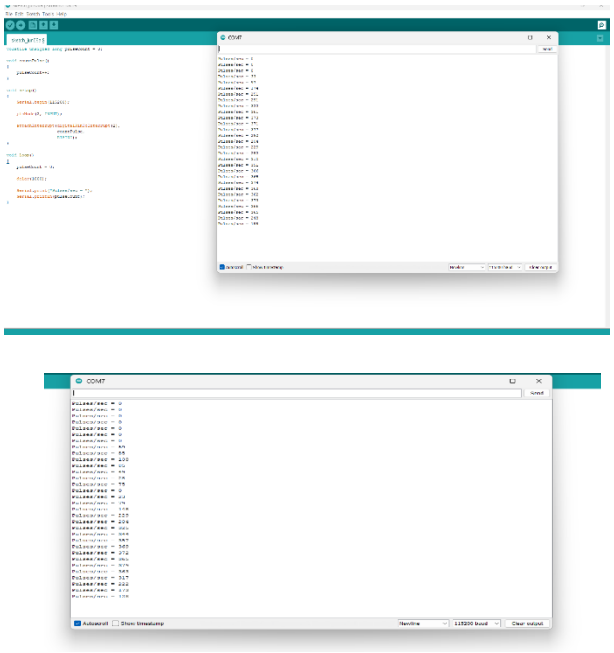


Fig 3b. Pulses/sec at low and high speeds of motor

X. RESULTS AND DISCUSSION

Available project outputs are presented and discussed qualitatively due to limited recorded datasets. The CAN link operated correctly: parameters transmitted by Node 1 under identifier 0x101 were received and decoded by Node 2 and rendered consistently on the serial monitor and the LCD.

On the receiver LCD (Fig. 3), the decoded values were observed as V = 11.00 V, I = 0.00 A, TEMP = 35.00 °C, SOC = 72.00 %, and SOH = 100.00 %.

Table 4. Observed Output Readings

Source / Parameter	Observed Value
LCD — Voltage	11.00 V
LCD — Current	0.00 A (no load)
LCD — Temperature	35.00 °C
LCD — SOC	71.00 %
LCD — SOH	100.00 %

On the Blynk dashboard the gauges read voltage = 10 V, temperature = 40 °C, current = 0 A, SOC = 55 %, and SOH = 100 %, confirming that the parameters were available for remote monitoring. A temperature-fault condition was also captured on the LCD as a “HIGH TEMP! / LOAD OFF” message (Fig. 4), demonstrating a protection response in which the load is switched off.

Fig. 4. LCD showing the “HIGH TEMP! / LOAD OFF” protection warning.

The motor-speed routine produced a continuous stream of readings on the serial monitor; the pulse frequency rose from 0 Hz up to about 404 Hz and the corresponding computed speed reached approximately 5486 RPM before returning toward zero, as shown in Fig. 5.

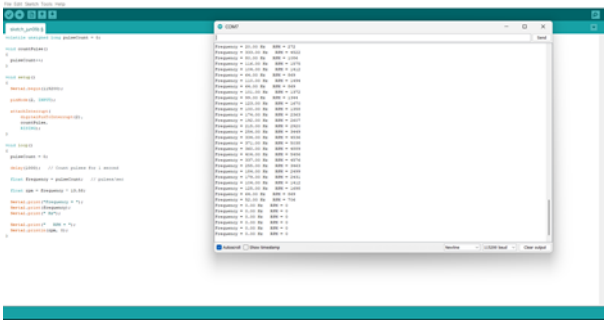


Fig. 5. Serial-monitor output of the motor-speed routine (Frequency in Hz and computed RPM).

The results confirm reliable CAN-based transfer of the battery parameters between the two ESP32 nodes and their consistent presentation on the LCD, the serial monitor, and the IoT dashboard. It is noted that the temperature read 0.00 °C on the bench captures and that, because the CAN payload carries the parameters as single-byte integers, the displayed values appear without fractional resolution; these observations are discussed further under limitations.

XI. ADVANTAGES

The developed system offers several advantages for electric vehicle monitoring applications:

- Distributed architecture using CAN communication reduces wiring complexity and improves scalability.
- Simultaneous local and remote monitoring is achieved through LCD and IoT dashboards.
- Real-time visualization of battery voltage, current, temperature, SOC, and SOH improves operator awareness.
- Relay-assisted protection enhances battery safety during abnormal thermal conditions.
- Modular implementation allows independent development of battery and motor monitoring subsystems.
- Low-cost hardware components make the system suitable for academic and prototype vehicle development.
- The architecture can be extended to support additional vehicle parameters and control functions.

XII. LIMITATIONS

- Despite successful implementation, several limitations exist in the current system:
- Battery monitoring was validated using a prototype 12 V battery rather than the actual 72 V vehicle battery.
- SOC estimation is based on voltage mapping and does not incorporate advanced battery modeling techniques.
- SOH is presently used as an indicative parameter and is not derived from detailed battery degradation analysis.
- Motor parameters are monitored independently and are not currently transmitted through the CAN network.
- Historical cloud data logging and advanced analytics have not been implemented.
- Cell-level voltage and temperature monitoring are not included in the present design.

XIII. FUTURE SCOPE

- Several enhancements can be implemented to expand the capabilities of the proposed system:
- Integration with the actual 72 V, 80 Ah SAE eBAJA battery pack.
- Direct CAN communication with the vehicle Battery Management System (BMS).
- Development of advanced SOC and SOH estimation algorithms.
- Cloud-based historical data storage and visualization.
- Predictive maintenance and fault forecasting using machine learning techniques.
- CAN-based acquisition of motor controller parameters.
- Vehicle speed, torque, and power monitoring through a unified dashboard.
- GPS-enabled fleet tracking and telemetry functions

- Cell-level voltage and temperature monitoring.
- Integration of motor-speed information into the CAN communication framework and IoT dashboard.
- Real-time diagnostic and fault classification capabilities.
- Full-scale deployment within competitive and commercial electric vehicle platforms.

XIV. CONCLUSION

A CAN-based battery monitoring and protection system with an IoT remote dashboard was implemented and tested for electric vehicle application, targeting an eBAJA Electric All-Terrain Vehicle. The system uses two ESP32 nodes communicating through MCP2515 CAN controllers at 500 kbps. The transmitting node acquires voltage, temperature, and current, computes SOC and reports SOH, and sends the parameters in an 8-byte CAN frame with identifier 0x101; the receiving node decodes the frame and displays the data on a 20×4 I2C LCD and the serial monitor, while the parameters are also published to a Blynk dashboard. Bench testing on a three-cell lithium-ion prototype confirmed correct CAN transfer and consistent display across all interfaces, an observed temperature-fault "LOAD OFF" response, a pulse-based motor-speed measurement, and monitoring of the 20-cell vehicle pack with a commercial smart-BMS application. The results demonstrate a working distributed monitoring framework that can be extended with firmware-based protection and higher-resolution data handling.

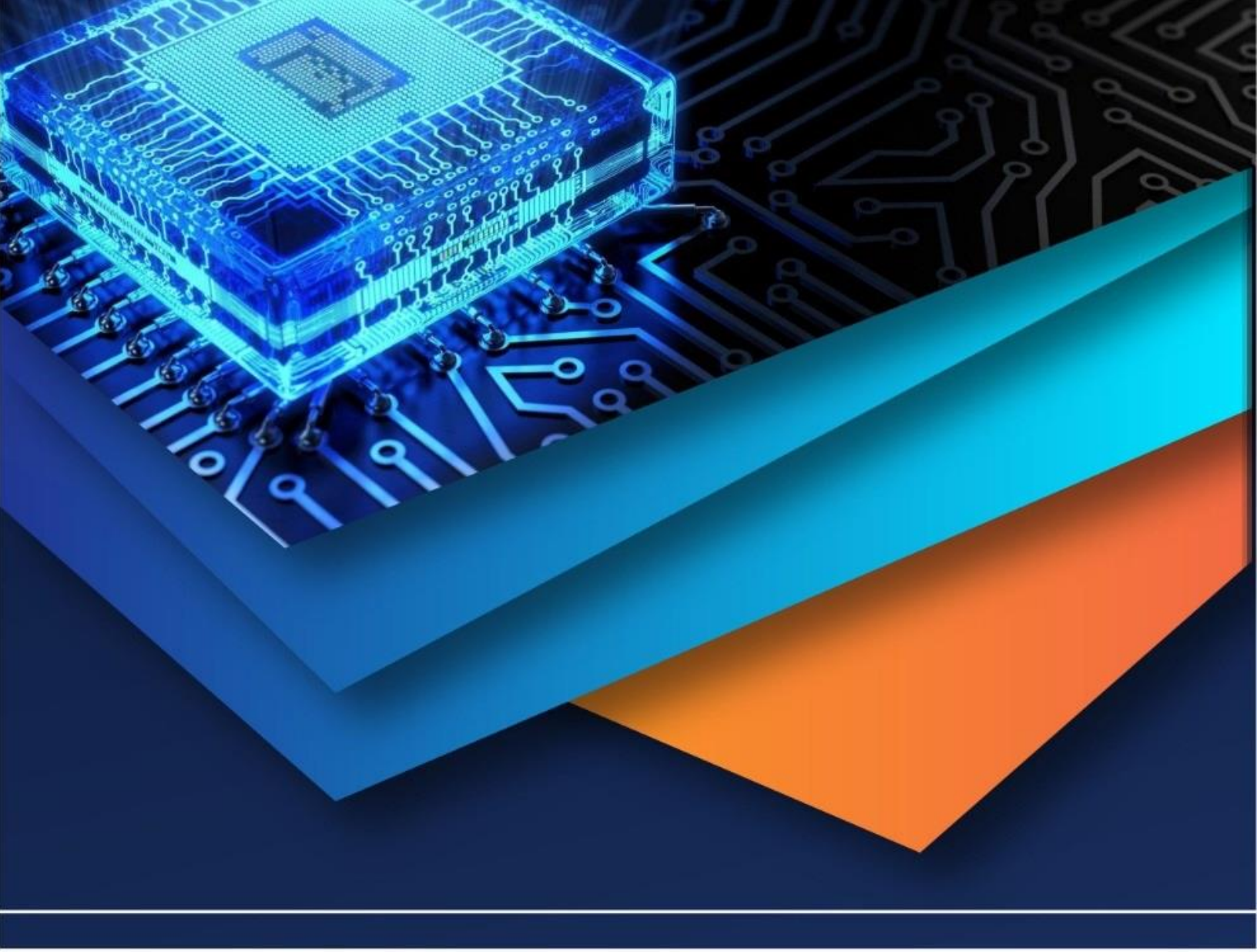
XV. ACKNOWLEDGEMENT

The authors acknowledge the support provided for the development and testing of this project. [Detailed acknowledgements — Information Not Available in Provided Files.]

REFERENCES

[1] R. Bosch GmbH, "CAN Specification, Version 2.0," Stuttgart, Germany, 1991.
[2] ISO 11898-1:2015, "Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling," International Organization for Standardization, 2015.
[3] Espressif Systems, "ESP32 Series Datasheet," Espressif Systems, 2023.
[4] Microchip Technology Inc., "MCP2515 Stand-Alone CAN Controller with SPI Interface Datasheet," 2019.
[5] Texas Instruments, "LM35 Precision Centigrade Temperature Sensors Datasheet," 2017.
[6] Allegro MicroSystems, "ACS712 Fully Integrated, Hall-Effect-Based Linear Current Sensor IC Datasheet," 2017.
[7] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," Journal of Power Sources, vol. 226, pp. 272–288, 2013.
[8] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," IEEE Industrial Electronics Magazine, vol. 7, no. 2, pp. 4–16, 2013.
[9] M. Brandl et al., "Batteries and battery management systems for electric vehicles," in Proc. Design, Automation & Test in Europe Conference (DATE), 2012, pp. 971–976.
[10] Blynk Inc., "Blynk IoT Platform Documentation," Blynk Inc. [Online].

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



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)