Smart Battery Charging and Protection System

Dr. Prateek Singhal[1], Satyam[2], Vinay Kumar[3], Vishal Jyani[4], Rahul Sharma[5]
[1-5]Department of Electrical Engineering,
Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur

Abstract- ***This project presents the development of a Battery Management System (BMS) tailored for Electric Vehicles (EVs), utilizing an Arduino UNO as the central controller. The system is designed to ensure the safe and efficient operation of 18650 Li-ion rechargeable battery cells by integrating key components such as a 16x2 LCD display for real-time monitoring, ACS712 current sensor for current flow measurement, and NTC 103 thermistor for temperature sensing. Voltage regulation is managed by an LM7805 regulator, while charging control and safety features are supported by relays, diodes, and electrolytic capacitors. Additional safety and interface elements include a buzzer for alerts, a CPU cooling fan for thermal management, and switches for manual control. The system provides essential protections such as overcurrent, overtemperature, and overvoltage safeguards, contributing to battery longevity and operational reliability.***

The BMS incorporates a user-friendly interface and intuitive control via a push-button switch and potentiometer, allowing for dynamic system adjustments. Data from sensors is processed and displayed, providing immediate feedback on battery health and operational status. The inclusion of a DC motor simulates load conditions, enabling practical testing and validation of the BMS under variable scenarios. This project demonstrates the feasibility of constructing a cost-effective and customizable BMS prototype suitable for educational and experimental EV applications, promoting a deeper understanding of battery safety and energy management in sustainable transport systems.

Keywords– BMS, Arduino Uno, Sensors, Protection.

1. INTRODUCTION

As the global demand for sustainable transportation grows, electric vehicles (EVs) have emerged as a pivotal solution to reduce carbon emissions and dependence on fossil fuels.[12] At the heart of every EV lies a battery system that requires careful monitoring and control to ensure safety, efficiency, and longevity. A Battery Management System (BMS) plays a critical role in achieving these goals by overseeing various parameters such as temperature, voltage, current, and overall battery health. Without a reliable BMS, lithium-ion batteries—commonly used in EVs—are vulnerable to risks like overheating, overcharging, and deep discharging, which can lead to reduced performance or even hazardous situations.

This project focuses on the design and implementation of a basic yet functional BMS prototype using an Arduino UNO microcontroller. It integrates essential components including a 16x2 LCD display for real-time feedback, sensors for current and temperature monitoring, voltage regulators, relays, and cooling mechanisms. The system is designed to detect abnormal operating conditions and respond with appropriate safety actions, such as activating a buzzer or shutting down the load. With features that replicate real-world EV battery challenges and controls, this project serves as an educational platform and a cost-effective prototype to explore and understand the fundamental operations and importance of battery management in electric vehicles.

1.1 MOTIVATION

The motivation for this research is driven by three critical factors:

A. Safety and Thermal Stability

Battery failures, particularly those involving thermal runaway batteries, put users and infrastructure at serious danger. A number of widely reported incidents in the electric car industry have highlighted the need for the country to create sophisticated battery monitoring systems that employ predictive analytics to identify malfunctions before they occur rather than merely reacting to them after the fact.

B. Economic Longevity

The costliest parts of solar storage systems and electric cars are their batteries. A battery pack is only as powerful as its weakest individual cell because current management strategies cause the cells inside a battery to age at various rates. We can extend the battery's useable life, lower the total cost of ownership, and lessen the amount of electronic waste generated by battery use if we can enhance the process used to balance cells.

C. Resource Efficiency (Sustainability)

Utilizing every gram of material to obtain the maximum amount of energy is crucial due to the rising cost of raw materials (such cobalt and lithium). An effective BMS will guarantee that all of the battery's capacity is used during its whole life cycle and that no energy is wasted to heat while charging or discharging.

1.2. LITERATURE REVIEW

"Design and Analysis of Battery Management System for Electric Vehicles"
by R. Suresh Kumar, S. Elango, M. Vijayan, S. A.Vijay. This study presents a comprehensive design of a BMS tailored for electric vehicles, focusing on monitoring and controlling battery parameters to ensure safety and efficiency. The authors detail the integration of various sensors and control mechanisms to manage charging and discharging cycles effectively.[1]

"An Efficient Battery Monitoring System for Electrical Vehicles Based on Arduino" *by Jinyu Yang, Hongxiu Zhao*. The authors propose an Arduino-based battery monitoring system that efficiently tracks battery parameters in electric vehicles. The system aims to optimize performance and ensure safety by providing real-time data and alerts.[2]

"Battery Management System in Electric Vehicles" *by A. Hariprasad, I. Priyanka, R. Sandeep, O. Shekar, V. Ravi.* This study focuses on the role of BMS in monitoring and controlling the charging and discharging of rechargeable batteries in electric vehicles. It highlights techniques for estimating state of charge, health, and life to maintain battery reliability and safety.[3]

"IoT Based Battery Management System for Electric Vehicles". *by Dr. N. N. Ghuge, Dandge Vishal, Bari Kunal, Mahesh Kulkarni*. The authors propose an IoT-enabled BMS that allows real-time monitoring of battery performance in electric vehicles. The system is designed to detect deteriorated battery conditions and notify users, facilitating proactive maintenance and safety.[4]

"A Smart Battery Management System for Electric Vehicles Using Deep Learning-Based Sensor Fault Detection" *by Venkata Satya Rahul Kosuru, Ashwin Kavasseri Venkitaraman*. This paper introduces a smart BMS that leverages deep learning techniques to detect sensor faults in electric vehicles. By employing advanced algorithms, the system enhances the reliability and safety of battery operations, ensuring accurate monitoring and fault diagnosis.[5]

"Battery Cloud with Advanced Algorithms" *by Xiaojun Li, David Jauernig, Mengzhu Gao, Trevor Jones.* The study presents a cloud-based BMS that utilizes advanced algorithms to improve battery performance and safety. By collecting and analyzing data from electric vehicles, the system estimates the state of charge and health, and detects thermal anomalies, thereby enhancing battery management strategies.[6]

"Two-Layer Model Predictive Battery Thermal and Energy Management Optimization for Connected and Automated Electric Vehicles" *by Mohammad Reza Amini, Jing Sun, Ilya Kolmanovsky*. This research proposes a two-layer model predictive control strategy for optimizing battery thermal and energy management in electric vehicles. The approach aims to improve energy efficiency by considering long-term traffic information and battery temperature dynamics, offering a comprehensive solution for connected and automated EVs.[7]

"Electric Vehicle Battery Management System with Charge Monitor and Fire Protection" *by Mohammed Mudassir Alam, Parikshith Jagli, Mohammed Ayman, Mohammed Javeed Akhtar, Dr. M. Srinivasan*. This paper discusses the design of a BMS equipped with charge monitoring and fire protection features. Utilizing analog/digital sensors and an STM32 microcontroller, the system monitors voltage, current, and temperature to maintain battery safety and reliability, addressing critical aspects of EV battery management.[8]

2. HARDWARE IMPLEMENTATION

2.1 Arduino Uno *(ATmega328P)*

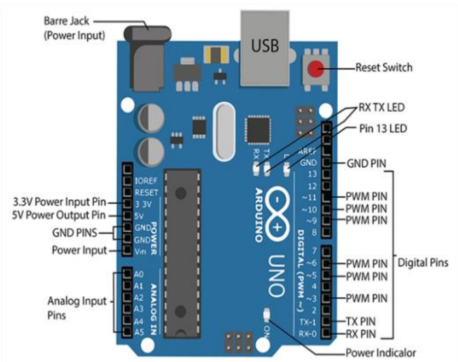

Fig.1: Arduino Uno board

It acts as the brain of the project as it commands all the actions that take place in the system[9] [10]. It collects the real time data from all the sensors like current from current sensor, voltage from voltage sensor, temperature from thermistor, etc. And then processes them through the instructions fed to it in the form of codes to detect any risk like overvoltage, undervoltage, increase in temperature of battery, etc. It then commands the relay to open or close the circuit of exhaust fan (in case of heating), buzzer (in case of sudden voltage drop).

2.2 Relay Module (4-Channel SRD-05VDC-SL-C)

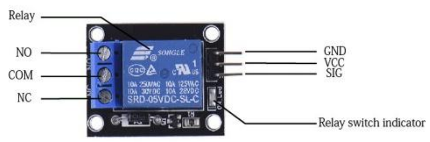

Fig.2: 4-channel relay module

The relay module that we have used is a 4-channel relay module which can operate at 10A for 250V AC and 30V DC. It acts as an interface between microcontroller and the battery pack and the fan. When the battery reaches its threshold value of temperature the microcontroller (Arduino uno) orders relay to close the circuit of exhaust fan and it opens the circuit when the temperature of battery reduces to a predefined value. Relay also regulates charging mode with the help of push buttons. When a specific button, with low resistance in its path, is pressed the relay closes the circuit for fast charging and opens it when button is pressed again, and the charging rate goes to pervious state. This same action is performed by the relay when the other button, with higher resistance in its path, is pressed but this time for slow charging.

2.3 Current Sensor (ACS712 30A)

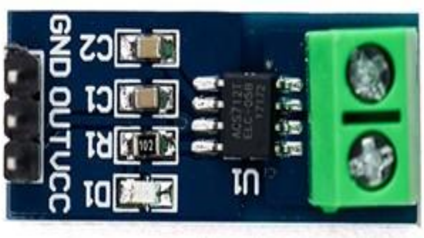

Fig.3: Current sensor

It detects the real-time consumption of the current or we can say the current flowing from the battery or to the motor. It sends the analog voltage signals to the Arduino which interprets this signal to monitor the consumption. And this data is shown in LED which helps the user to assess the state of charge or health of battery so as to regulate the charging mode of the battery.[13]

2.4 Temperature Sensor (NTC 10K Thermistor)

Fig.4: Thermistor

It is fundamental for heat protection of battery and it is cost effective option for this purpose [10]. It senses the change in temperature of battery pack. When temperature reaches the threshold value(33°C), the microcontroller activates the DC Cooling Fan with the help of relay module. And when the sensor senses the temperature of the battery pack to be below 33°C the fan turns off.

2.5 LCD (16X2)

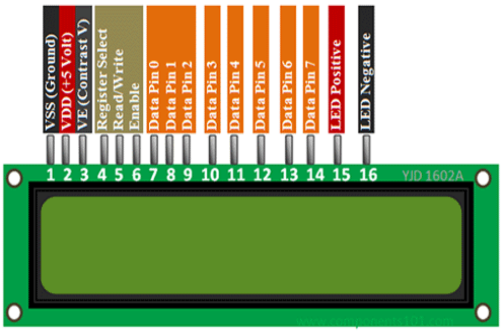

Fig.5: LCD

It is interfaced with the Arduino uno to display the real-time data of the battery. This data involves the real time current consumption of the battery, the voltage across the battery, temperature of the battery. It also shows if the battery is in fast charging, slow charging, or no charging mode.

2.6 DC Motor

Fig.6: DC Motor

It is a 12V DC motor rotating at 60 to 100 rpm as it simulates the high torque movement of the Electric Vehicle. It is powered by the Li-ion battery pack of 12V rating. It helps demonstrating real world scenario.

2.7 DC Cooling Fan (SU-8025 12V)

Fig.7: Coolong fan

It works as the heat management system by forced convection of heat generated by battery due to high current discharge or overcharging. It comes in action when the temperature of battery pack reaches 33ºC and gets turned off as soon as the battery's temperature reaches below 33ºC. It helps in increasing the lifespan of battery and reduce the risk of the battery catching fire. Hence it saves a lot of expenses of users.

2.8 Li-ion Battery Pack (3.7V per cell or 12V as whole)

Fig.8: Li-ion battery pack

It works as the primary energy storing device. It provides current to the motor and to the electronics of the system (EV). It is the heart of the whole system as it is the most crucial and vulnerable part of the EV. So, it is justifiable to make a BMS to monitor the battery's temperature, charging rate, etc.

2.9 Buzzer (5V Active Piezo)

Fig.9: Buzzer

It acts as a safety/alarming system which operates when the voltage level across the battery suddenly drops to zero or there is a short circuit.

3. CODE

The control logic was implemented using the Arduino C++ programming language [16] and compiled within the Arduino IDE [18]. To optimize pin usage on the microcontroller, the 16x2 LCD was interfaced via the I2C protocol using the LiquidCrystal_I2C library [17].

```
#include<LiquidCrystal.h>

// LCD: RS, E, D4, D5, D6, D7
 LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

 // === Macros for Thresholds ===
#define VOLTAGE_THRESHOLD_LOW 8.0  // V
 #define TEMP_THRESHOLD_HIGH  45.0   // °C #define
CURRENT_SENSOR_SENSITIVITY 0.066  // V/A for
ACS712 30A
 #define VREF 5.0
 #define VOLTAGE_DIVIDER_RATIO 5.0   // Adjust this
to match your divider (e.g., 5:1)

// === Pin Definitions ===

#define THERMISTOR_PIN A0
#define CURRENT_SENSOR_PIN A1
 #define VOLTAGE_SENSOR_PIN A2
 #define FAN_PIN 12
#define BUZZER_PIN 13
#define BTN1_PIN 8
#define BTN2_PIN 9
#define RELAY1_PIN 10
#define RELAY2_PIN 11

 // === Globals ===

unsigned long lastLcdUpdate = 0;
byte lcdScreenIndex = 0;
bool btn1State = false;
bool btn2State = false;
bool relay1On = false;
bool relay2On = false;

 // === Function Prototypes ===

float readTemperature();
float readCurrent();
float readBatteryVoltage();
void updateRelays();
void updateLCD(float voltage, float temp, float current);
void checkAlarms(float voltage, float temp);
void toggleRelay(bool &state, int pin, int buttonPin, bool
&prevButtonState);
void setup() {
 lcd.begin(16, 2);
 pinMode(FAN_PIN,OUTPUT);
 pinMode(BUZZER_PIN,OUTPUT);
```

```
 pinMode(RELAY1_PIN,OUTPUT);
 pinMode(RELAY2_PIN,OUTPUT);
pinMode(BTN1_PIN,INPUT_PULLUP);
pinMode(BTN2_PIN,INPUT_PULLUP);
digitalWrite(FAN_PIN,LOW);
digitalWrite(BUZZER_PIN,LOW);
digitalWrite(RELAY1_PIN,LOW);
digitalWrite(RELAY2_PIN, LOW);
}
 void loop() {
float voltage = readBatteryVoltage();
float temp = readTemperature();
float current=readCurrent();
checkAlarms(voltage,temp);
updateRelays();
updateLCD(voltage, temp, current);
delay(200);
}
void toggleRelay(bool &state, int pin, int buttonPin, bool &prevButtonState) {
 bool pressed = !digitalRead(buttonPin);
if (pressed && !prevButtonState)
{
state = !state;
digitalWrite(pin, state);
 }
 prevButtonState = pressed;
}
void updateRelays() {
 toggleRelay(relay1On, RELAY1_PIN, BTN1_PIN, btn1State);
 toggleRelay(relay2On, RELAY2_PIN, BTN2_PIN, btn2State);
}
void checkAlarms(float voltage, float temp)
 {
digitalWrite(FAN_PIN, temp > TEMP_THRESHOLD_HIGH);
digitalWrite(BUZZER_PIN, voltage < VOLTAGE_THRESHOLD_LOW);
}
void updateLCD(float voltage, float temp, float current)
{
  unsigned long now = millis();
 if (now - lastLcdUpdate < 5000) return;

lcd.clear();
switch (lcdScreenIndex) {

case 0:

lcd.setCursor(0, 0);
lcd.print("Volt:");
lcd.print(voltage, 1);
lcd.print("V");
lcd.setCursor(0, 1);
lcd.print("Temp:");
lcd.print(temp, 1);
lcd.print((char)223); // degree symbol
lcd.print("C");
break;

case 1:

lcd.setCursor(0, 0);
lcd.print("Current:");
lcd.print(current, 1);
lcd.print("A");
lcd.setCursor(0, 1);
lcd.print("Relay:");
if (relay1On && relay2On)
lcd.print("Fast Chg");
else if (relay1On)
lcd.print("Slow Chg");
else
lcd.print("Chg Off ");
break;

case 2:

if (voltage < VOLTAGE_THRESHOLD_LOW) {
lcd.setCursor(0, 0);
lcd.print("Warning:");
lcd.setCursor(0, 1);
lcd.print("Low Voltage!");
 }
else if (temp > TEMP_THRESHOLD_HIGH) {
lcd.setCursor(0, 0);
lcd.print("Warning:");
lcd.setCursor(0, 1);
lcd.print("High Temp!");
}
else
{
lcd.setCursor(0, 0);
lcd.print("System OK");
lcd.setCursor(0, 1);
lcd.print("Monitoring...");
}
break;
}
lcdScreenIndex = (lcdScreenIndex + 1) % 3;
lastLcdUpdate = now;
}
float readBatteryVoltage() {
 int raw = analogRead(VOLTAGE_SENSOR_PIN);
float voltage = (raw * VREF / 1023.0) * VOLTAGE_DIVIDER_RATIO;
return voltage;
}
float readTemperature() {
int raw = analogRead(THERMISTOR_PIN);
float voltage = raw * VREF / 1023.0;
float resistance = (VREF * 10000.0 / voltage) - 10000.0; // 10k series resistor   float tempK = 1.0 / (log(resistance / 10000.0) / 3950.0 + 1 / 298.15);
return tempK - 273.15;
}
float readCurrent() {
 int raw = analogRead(CURRENT_SENSOR_PIN);
float voltage = raw * VREF / 1023.0;
float current = (voltage - 2.5) / CURRENT_SENSOR_SENSITIVITY;
return current;
}
```

4. METHODOLOGY

This Smart Charging and Battery Protection system is a closed loop automated system which ensures the safety of the 12 V Li ion battery pack and provides us with the real time data of the different parameters of the battery. Its working starts from stepping down the 240V AC supply to 12V AC supply with the help of a center taped 12-0-12 transformer, then this supply is converted into a stable DC supply by a IN4007 full bridge diode rectifier[14]. A LM7805 voltage regulator then converts it into 5V DC which can be used by the Arduino and all other electronic appliances. This supply reaches to the 12V Li-ion battery pack through a set of push buttons and a 4-channel relay module. The synergy of push buttons, Arduino and relay module helps to regulate the charging rate to toggle between fast, slow, and no charging. Pushing a specific push button connects the battery through a path with high current for fast charging and pushing the other button connects the battery thorough a path with lower current for slow charging, and relay cuts the connection when no charging needed. Pushing the buttons again discontinues the rate of charging selected previously.[1]

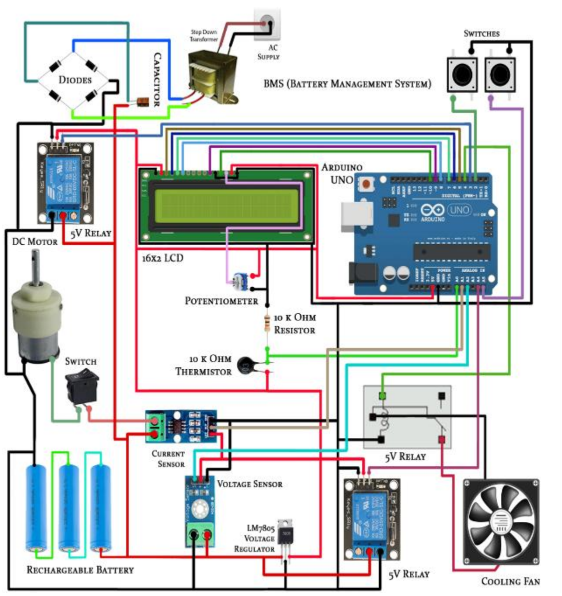

Fig.10: Circuit diagram

Sensors like current sensor (ACS712), voltage sensor (DCVS30-H16S8), and temperature sensor (NTC 10K Thermistor) are connected in circuit, which measures the real time parameters like current, voltage, and temperature and sends the analog signa to the Arduino which interprets these signals and displays them, in digital form, on the screen of a 16X2 LCD connected to it.

If the voltage across the battery drops suddenly to a negligible value the microcontroller orders relay to close circuit of the Piezo Buzzer which alerts the user to take necessary actions. And a warning of "low voltage" is also displayed on the LCD screen.

As the battery operates for a long time its temperature rises. If the temperature reaches to a predefined value, in the coding for Arduino, of 33^{o}C, the temperature sensor senses it and sends data to Arduino which immediately orders the relay module to close the circuit of DC Cooling Fan (SU-8025 12V) which reduces the temperature of battery by forced convection of heat produced due to continuous use or high current drawn by the battery. A message of "high temperature" is also displayed on the LCD screen to alert the user. When the temperature of the battery reaches below the value of 33^{o}C, after a few time, it is detected by the temperature sensor and this data is sent to Arduino which order the relay module to open the circuit of the DC fan.

5. RESULT

This Smart Battery Charging and Protection System allows the Li – ion battery to integrate the whole system (hardware and software) with enhanced efficiency. It increased the lifespan of the battery and reduced wear and thermal degradation of battery by precise coding in Arduino Uno and the coordination with the sensors and relay module of the system. It also helps in maintain the core temperature of battery within harmless limit when the battery draws high current. It provides effective regulation of charging rate of the battery, toggling between fast, slow, and no charging which also aids in the improvement of battery's health. The buzzer and LCD provide with the real time condition of the battery, auditory and visual alerts, and helps with timely information, on the basis of which the user can take proper measure to tackle the abnormality by disconnecting the supply. These results support the fact that this system has robust fail-safe mechanism and allows the battery to operate within the golden window of operation which provides high efficiency while preventing the risk of thermal runaway.

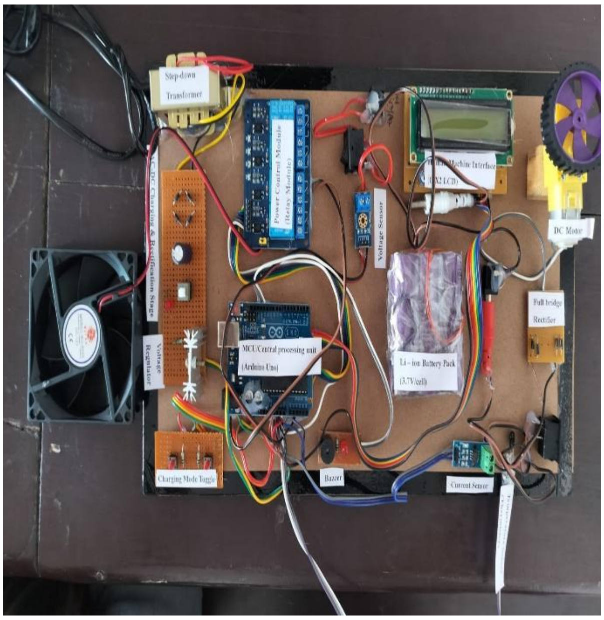

Fig.11: Project Hardware

6. CONCLUSION

The presented Smart Battery Charging and Protection System proves that a cost-effective, Arduino based electronic system can reduce the primary risks associated with a Li-ion based battery in Electrical Vehicles. The integration of various sensors in the system shows its ability to regulate the charging rates of battery and to reduce the heat of battery. Heat reduction aids in prevention of thermal stress on battery. The buzzer and LCD provides operational transparency and the alerts given by them helps in timely action to tackle the problem. Hence, this project provides us with a scalable framework for battery longevity and safety, proving that an accurate software program integrated with robust hardware switching can significantly reduce the possibility of cell degradation and thermal runaway of batteries of Electric Vehicles.

7. REFERENCES

[1]. R. Suresh Kumar, S. Elango, M. Vijayan, and S. A. Vijay, "Design and Analysis of Battery Management System for Electric Vehicles."

[2]. J. Yang and H. Zhao, "An Efficient Battery Monitoring System for Electrical Vehicles Based on Arduino."

[3]. A. Hariprasad, I. Priyanka, R. Sandeep, O. Shekar, and V. Ravi, "Battery Management System in Electric Vehicles."

[4]. N. N. Ghuge, D. Vishal, K. Bari, and M. Kulkarni, "IoT Based Battery Management System for Electric Vehicles."

[5]. V. S. R. Kosuru and A. K. Venkitaraman, "A Smart Battery Management System for Electric Vehicles Using Deep Learning-Based Sensor Fault Detection."

[6]. X. Li, D. Jauernig, M. Gao, and T. Jones, "Battery Cloud with Advanced Algorithms."

[7]. M. R. Amini, J. Sun, and I. Kolmanovsky, "Two-Layer Model Predictive Battery Thermal and Energy Management Optimization for Connected and Automated Electric Vehicles."

[8]. M. M. Alam, P. Jagli, M. Ayman, M. J. Akhtar, and M. Srinivasan, "Electric Vehicle Battery Management System with Charge Monitor and Fire Protection."

[9]. Maltezo, M. R. C. (2021). "Arduino-Based Battery Monitoring System with State of Charge Estimation." *Semantic Scholar*.

[10]. Srujana, B., et al. (2025). "Battery Temperature Monitoring System Using Arduino." International Journal of Advances in Applied Sciences (IJAAS).

[11]. Chavan, M., et al. (2025). "Thermal Management System for Electric Vehicles." *RSIS International*.

[12]. Al-Gabalawy, M., et al. (2024). "Battery Energy Storage Systems: A Review of Energy Management Systems and Health Metrics." *MDPI Energies*.

[13]. *ACS712 Fully Integrated, Hall Effect-Based Linear Current Sensor IC Data Sheet*, Allegro MicroSystems, 2022.

[14]. P. S. Bimbhra, *Power Electronics*, 6th ed. New Delhi, India: Khanna Publishers, 2021.

[15]. M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*, 4th ed. London, UK: Pearson Education, 2017.

[16]. *Arduino Language Reference*, Arduino, 2026.

[17]. M. Schwartz, *LiquidCrystal_I2C Library for Arduino*, GitHub, 2025.

[18]. *Arduino Integrated Development Environment (IDE) v2.x*, Arduino, 2026.