International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

An Integrated Hardware Framework for Mitigating Critical Operational Threats in Next-Generation Battery Swapping Stations

Hardik Agravatt[1], Indrajit Trivedi[2], Snehal Purani[3]

[1]Research Scholar, Gujarat Technological University, Ahmedabad, India
[2]Professor, Power Electronics Department, Vishwakarma Government Engineering College, Ahmedabad, India
[3]Research Scholar, Gujarat Technological University, Ahmedabad, India

Abstract
Battery swapping stations are gaining traction as a workable route to shrink the downtime that ordinarily accompanies electric-vehicle charging; even so, deploying them in the field exposes a cluster of operational risks that span energy pilferage, weak traceability of battery health, and breaches of physical security. A consolidated, inexpensive, hardware-oriented mitigation framework is proposed in this work, organised around six cooperating modules. Battery voltage is acquired through a resistive divider that feeds a microcontroller analogue-to-digital converter, and the state of charge is reconstructed from the recovered reading. Access is governed by a Bluetooth authentication layer built on the HC-05 module together with relay switching, so that charging remains blocked until the supplied credentials are matched against stored ones. Additional platform headroom is obtained by migrating from 8-bit devices to the ESP32, whose dual-core processor, 12-bit conversion, and on-chip Wi-Fi permit sensing, logging, and connectivity to proceed together. Cycle history is captured with an ACS712 current sensor combined with temporal filtering and is retained in on-board EEPROM or flash so that it survives power interruptions. Physical protection is delivered by a subsystem that combines a NEO-6M receiver for live positioning, a SIM900A module for mobile notification, and ultrasonic ranging for tamper sensing. A swap-decision pipeline finally unites a MATLAB interface with ThingSpeak analytics to derive a dynamic swap price from a fixed component, the energy delivered, and a health-dependent penalty factor. The resulting architecture is intended for rapid replication on academic and early-stage industrial testbeds, and it supplies a structured engineering blueprint for secure, health-aware, and auditable swapping operations.

Keywords: Battery Swapping Station, Unauthorised Charging, Bluetooth Authentication, EEPROM Logging, Charging Cycle Counting, Voltage Divider, State of Charge Estimation, State of Health Estimation, MATLAB Interface, ThingSpeak, Operational Threats

1 Introduction

Battery swapping stations promise high availability because the time a vehicle spends in use is decoupled from the time a pack spends charging, which enables an almost instantaneous exchange in place of conventional plug-in charging. In practice, however, field deployments repeatedly encounter interlinked operational risks that erode safety margins, station profitability, and user confidence. These risks originate

in cyber-physical weaknesses affecting sensing accuracy, the enforcement of access control, and the durability of lifecycle records. The integrity of measurements, of access decisions, and of a traceable battery history therefore determines whether swapping remains safe, fair, and scalable.

Six operational threats are considered in this work and are addressed directly by the proposed modules.

Threat T1 (Unknown Battery State and Unsafe Charging). When the pre-swap voltage is unknown and repeated charge cycles are not tracked, packs may be driven into unsafe charge regions, which accelerates thermal and electrical ageing. A station that cannot measure voltage for state-of-charge inference, or record sustained current for cycle counting, cannot enforce a safe charging policy.

Threat T2 (Unauthorised Access and Energy Theft). In shared infrastructure, charging that begins without authentication constitutes energy theft and introduces a safety hazard. A low-cost yet deterministic access-control mechanism is therefore required so that the charging path can be isolated by a relay until a valid credential is presented.

Threat T3 (Processing Bottlenecks and Limited Connectivity). Early prototypes that rely on 8-bit controllers lack the processing capacity and the native connectivity needed to sense, log to the cloud, and compute at the same time. The consequence is latency in data transfer and reduced fidelity during high-frequency monitoring.

Threat T4 (Hidden Degradation and Loss of Traceability). The accumulation of charge cycles is a primary driver of degradation and of economic value; when this history is held in volatile memory it is lost during interruptions, and health-aware decisions are undermined.

Threat T5 (Physical Theft and Absence of Tracking). Beyond energy theft, a pack can be physically removed from its housing. Without geolocation and tamper sensing, the operator has no means of recovering or tracking the asset.

Threat T6 (Operational Ambiguity and Unfair Pricing). Station throughput depends on how quickly a pack's health and price can be settled. In the absence of an interface that turns validated signals into state-of-charge and state-of-health metrics, operation degrades into manual judgement, and pricing that ignores delivered energy and health can cause financial loss.

These threats are addressed through a structured, low-cost hardware framework composed of six modules: voltage acquisition by a resistive divider for state-of-charge reconstruction (M1); Bluetooth-driven relay control that prevents unauthorised charging (M2); migration to the ESP32 platform for dual-core processing and native connectivity (M3); charge-cycle monitoring with an ACS712 sensor and persistent storage (M4); an anti-theft subsystem that integrates positioning, cellular messaging, and ultrasonic sensing (M5); and a decision layer that couples a MATLAB interface with ThingSpeak for health-aware pricing (M6). The intent is a deployable blueprint in which operational threats are contained before scaling multiplies the impact of any failure.

2 Literature Review and Gap Analysis

2.1 Current Trends in Swapping Architectures and Second-Life Use

Contemporary research on swapping stations is dominated by station sizing, scheduling, and grid interaction. A practical gap nevertheless separates these theoretical treatments from what can be realised on the station floor. Large-scale studies commonly assume that users are already authenticated and that accurate health data are available, whereas early deployments must synthesise those very quantities from low-cost sensing. The reuse of second-life electric-vehicle batteries for stationary storage has been shown to be feasible, yet its lifespan and economics are reported to depend strongly on rigorous assessment and

continuous health monitoring. A clear need therefore exists for frameworks that examine the design and case behaviour of such ageing assets in order to preserve operational safety.

2.2 Security, Authentication, and Anti-Theft Gaps

Security in the swapping literature is frequently limited to backend billing or application-level protocols, while physical access control at the charger-enable line is often neglected; this omission facilitates theft and unauthorised charging. Although radio-frequency identification is widely used, a Bluetooth-based route through the HC-05 module offers a more flexible, mobile-friendly path to authorised swapping. Physical protection remains a further gap, because many prototypes provide neither real-time geolocation nor tamper detection. Positioning and cellular modules for live tracking and mobile alerts are consequently required to safeguard assets in shared environments.

2.3 State-of-Charge and State-of-Health Estimation Gaps

Accurate estimation of state of charge and state of health often calls for electrochemical impedance models that are difficult to run on inexpensive hardware. Feature-based extraction methods have been demonstrated to yield high-fidelity health monitoring through intelligent algorithms. Early prototypes, however, need reproducible and actionable proxies derived from directly measurable quantities such as voltage and cycle count. Although on-board battery models and ageing-aware equivalent circuits have been explored, integrated frameworks that use simple interpolation to supply real-time decision variables to an operator interface remain scarce.

2.4 Traceability, Persistence, and IoT Integration

Battery history, and specifically the accumulation of charge cycles, directly shapes degradation and economic viability. Many prototype systems store this state in volatile memory, so it is lost when power is interrupted. EEPROM and flash-based logging are low-cost remedies but are seldom documented in the swapping literature. Moving to higher-performance controllers such as the ESP32 additionally enables native cloud integration and relieves the processing bottlenecks of 8-bit designs, and cloud monitoring through platforms such as ThingSpeak provides the transparency required for long-term health tracking and fleet management.

2.5 Charging Optimisation and Hardware Testing

Recent contributions have optimised charging strategies for residential and shipboard settings and have exercised hardware-in-the-loop emulators that test battery-management behaviour under a range of drive cycles. Despite these advances, a gap persists in local, low-cost feedback subsystems, such as audible alarms, that can continue to function independently of higher-level software during a fault.

2.6 Gap Addressed by This Work

The gaps identified above are addressed here by a complete, low-cost embedded framework that enforces physical authorisation through Bluetooth-relay control to prevent theft, migrates to the ESP32 for greater processing capacity and native connectivity, tracks charge cycles robustly with an ACS712 sensor and persistent EEPROM storage, integrates anti-theft protection through positioning and cellular reporting, and fuses a local MATLAB interface with ThingSpeak analytics to compute a health-aware swap price.

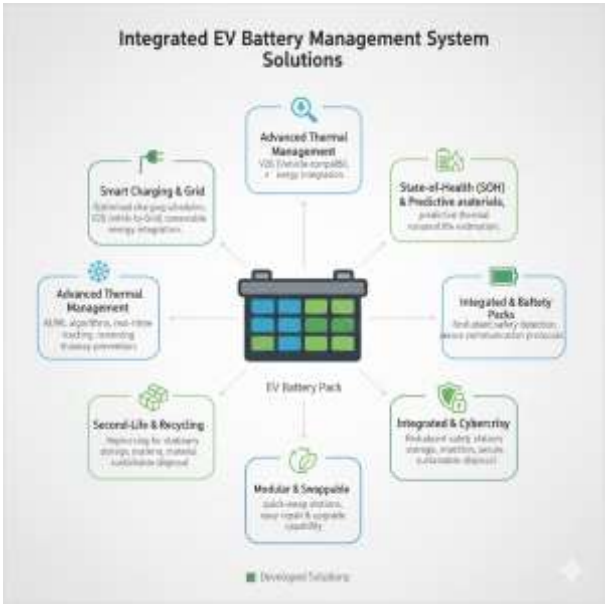
3 System Architecture and Threat-to-Module Mapping

3.1 Architecture Overview

The proposed integrated architecture is illustrated in Figure 1. A station controller acquires voltage and current, enforces credential-based access, records life cycles in non-volatile memory, and presents results

through a local display and operator interface. A separate low-cost alert node supplies audible signalling that is independent of the main controller, which improves resilience during faults.

Figure 1: Integrated System Architecture for Threat Mitigation in a Battery Swapping Station



3.2 Threat-to-Module Mapping

The correspondence between the six operational threats and the modules that mitigate them is summarised in Table 1. Each threat is linked to one or more modules so that the coverage of the framework can be verified at a glance.

Table 1: Mapping of Operational Threats to the Proposed Modular Framework

Threat	Failure Mode	Mitigating Modules
T1	Unknown battery state, unsafe charge region	M1 (voltage sensing), M4 (cycle history), M6 (decision gating)
T2	Unauthorised charging and energy theft	M2 (Bluetooth authentication), M2 (relay isolation), M3 (secure logic)
T3	Processing latency and connectivity gaps	M3 (dual-core), M3 (native Wi-Fi), M6 (cloud synchronisation)
T4	Hidden degradation and loss of traceability	M4 (cycle tracking), M4 (EEPROM or flash persistence)
T5	Physical battery theft and tampering	M5 (positioning and cellular), M5 (ultrasonic proximity), M5 (mobile alerts)
T6	Operational ambiguity and unfair pricing	M6 (interface metrics), M6 (health-based penalty pricing)

4 Methodology: Module-by-Module Design

The description of every module follows the same three-part structure: a theoretical framework that states the governing relations, a hardware implementation that lists connections and constraints, and a software logic that captures the control flow.

4.1 Module M1: Voltage Monitoring Through a Divider and Converter

Theoretical framework. Converter inputs on a microcontroller are usually limited to the range 0 to 5 V. A resistive divider is therefore used to scale a higher pack voltage into this range, as expressed in Equation 1.

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2} \quad (1)$$

For a 10-bit converter, the measured node voltage is reconstructed from the raw count through Equation 2.

$$V_{out} = \frac{\text{ADC} \times 5.0}{1024.0} \quad (2)$$

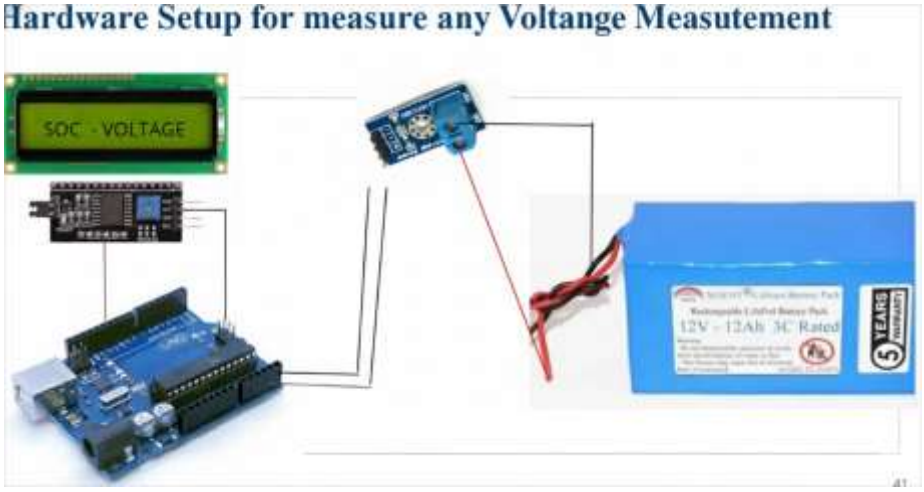
The input voltage is then recovered by inverting the divider ratio, as in Equation 3.

$$V_{in} = \frac{V_{out}}{\frac{R_2}{R_1 + R_2}} \quad (3)$$

With R1 = 30 kΩ and R2 = 7.5 kΩ, the scale factor is 0.2, which allows measurement up to approximately 25 V for a 5 V converter ceiling.

Hardware implementation. Analogue input A0 reads the node voltage. The divider input is connected to the battery positive terminal and the divider ground to the battery negative terminal, which is held common with the controller ground. A safety constraint requires that the node voltage stay at or below 5 V at the maximum expected pack voltage.

Figure 2: Hardware Connection for State-of-Charge Measurement Using a Resistive Divider and Converter Input



Software logic. The converter is sampled, the node voltage is obtained from Equation 2, the input voltage is reconstructed from Equation 3, and the result is published to the display, the serial link, and the interface.

Algorithm 1: Voltage Acquisition and Reconstruction (M1)

1. Read ADC <- analogRead(A0)
2. Vout <- (ADC x 5.0) / 1024.0
3. Vin <- Vout / (R2 / (R1 + R2))
4. Output Vin to display and log

Figure 3: Flowchart for State-of-Charge Estimation from Measured Voltage

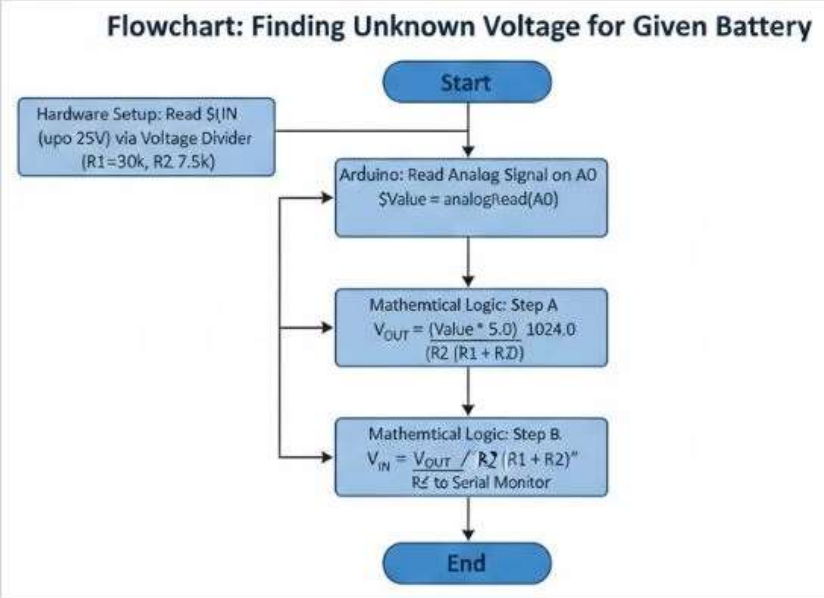


Figure 4: Experimental Prototype for State-of-Charge Measurement Validated on an E-Cycle Pack

4.2 Module M2: Bluetooth Authentication and Relay Control

Hardware interfacing. Module M2 forms the primary security layer against unauthorised charging. Its hardware comprises an HC-05 Bluetooth module for wireless input and a relay that acts as an automated

switch in the charging path. The components used are listed in Table 2.

Table 2: Components Used for the Bluetooth Authentication System

Number	Component	Rating	Function
1	Arduino Uno	ATmega328	Central processing
2	Bluetooth module	HC-05	Wireless input
3	Relay module	5 V SPDT	Circuit isolation
4	Display 16x2 with interface	I2C module	Local status display
5	Battery pack	12 V LiFePO4	Load

Password verification and relay operation. Charging is permitted only after the supplied credential is validated. The user transmits the battery password from an authorised Bluetooth device to the HC-05 module, and the controller compares the received string against a password stored in its internal memory. When the two match, a signal is issued to the relay to close the circuit and begin charging; when they do not, the relay is held in its normally open state so that electricity is physically prevented from reaching the battery.

Figure 5: Block Diagram of the Charging-Protection Circuit

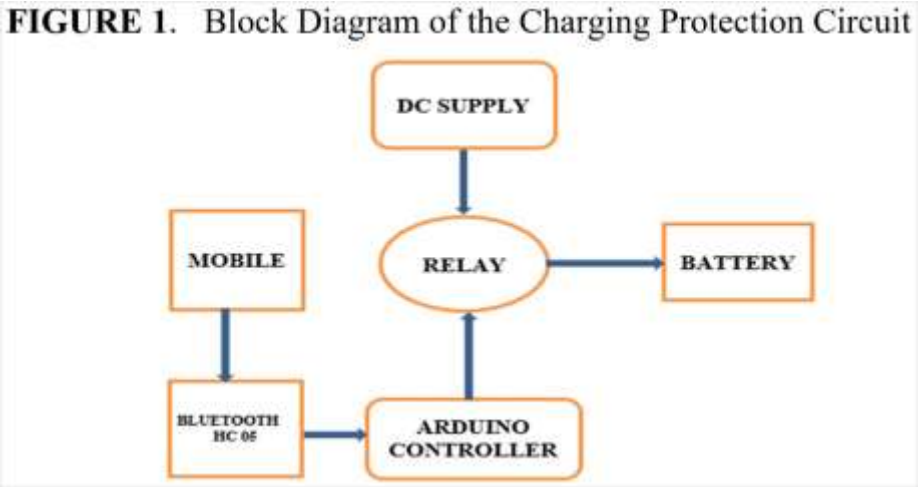


Figure 6: Hardware Setup Showing Integration of the Relay and Bluetooth Module

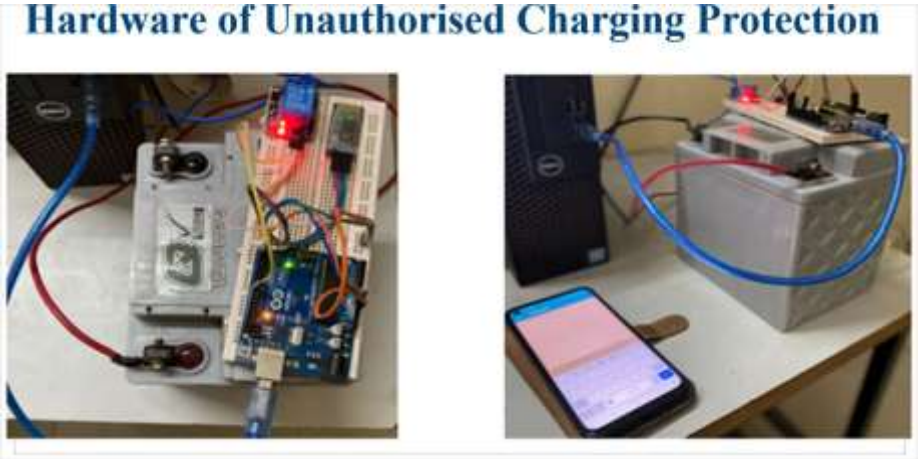


Figure 7: Interfacing Schematic of the HC-05 Module and Relay with the Controller

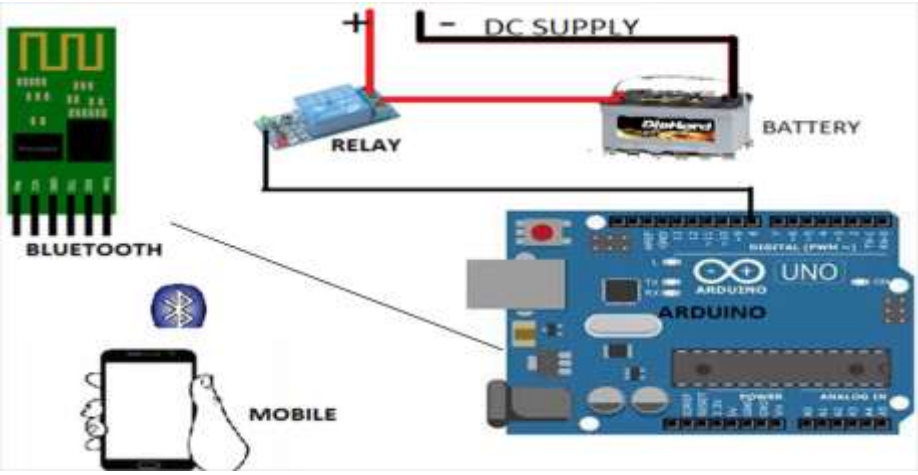


Figure 8: Bluetooth Control Application Used for Authorised Access

Algorithm 2: Bluetooth Authentication and Relay Control (M2)

1: Initialise HC-05 serial communication

2: Display "Enter Password" on the screen
3: if Bluetooth data is received then
4: inputPass <- read string from HC-05
5: if inputPass == stored battery password then
6: activate relay (pin HIGH) -> start charging
7: display "Authorized: Charging Started"
8: else
9: deactivate relay (pin LOW) -> charging prevented
10: display "Unauthorized: Access Denied"
11: end if
12: end if

4.3 Module M3: Controller Performance and Scalability with the ESP32

Rationale for platform migration. Module M3 proposes a move from the Arduino Uno to the ESP32. The two boards are comparable in price, but the ESP32 offers a substantial gain in processing capacity and in integrated features that a modern swapping station requires. The migration allows current sensing, voltage interpolation, and cloud communication to execute together without the latency inherent in 8-bit designs. A comparison of the two platforms is given in Table 3.

Table 3: Comparison of the Arduino Uno and the ESP32

Specification	Arduino Uno (Base)	ESP32 (Proposed Upgrade)
Microcontroller	ATmega328 (8-bit)	Xtensa LX6 (32-bit)
Connectivity	External module required	Native Wi-Fi and Bluetooth
Sensing precision	10-bit converter	12-bit converter (enhanced)
Timer accuracy	Standard clock	High-speed dual core
Approximate cost	450 rupees	525 rupees

Technical advantages. The higher clock and dual-core execution allow the two-minute timing used for cycle counting to be resolved more precisely; the built-in Wi-Fi and Bluetooth establish a direct link to the cloud without extra shields; the 12-bit converter raises the resolution of voltage sensing and thereby improves the interpolation used for state-of-charge extraction; and the internal flash provides more robust retention of counts and credentials during power-failure recovery.

Algorithm 3: Integrated Control and IoT Logic (M3)

1: Initialise dual-core processing and connect to the local Wi-Fi gateway
2: Acquire Vin and Iin using the 12-bit converter
3: Execute interpolation to determine the state-of-charge percentage
4: if Iin > 1 A for more than 2 minutes then
5: increment the local charge counter and synchronise with ThingSpeak
6: end if
7: Compute state of health and penalty factors for real-time pricing

8: Stream processed data to the MATLAB interface over the high-speed link

4.4 Module M4: Charging-Cycle Monitoring and Persistence with the ACS712

Current sensing. An ACS712 current sensor detects the flow of electricity between the supply and the pack, and the module distinguishes genuine charging cycles from transient noise or brief surges. The associated components are listed in Table 4.

Table 4: Components Used for the Charging-Counter System

Number	Component	Rating	Price
1	Battery	12 V, 28 Ah	3500 rupees
2	Arduino Uno microcontroller	ATmega328	450 rupees
3	Current sensor module	ACS712	100 rupees
4	Display 16x2 with interface	I2C module	190 rupees

Detection and persistence. A charging event is confirmed only when the current exceeds 1 A continuously for two minutes, which suppresses false counts from short surges. The count is written to non-volatile memory so that it survives reboots, brief interruptions of five to ten seconds are ignored, and the declining current at the end of a valid charge is used to mark the cycle complete.

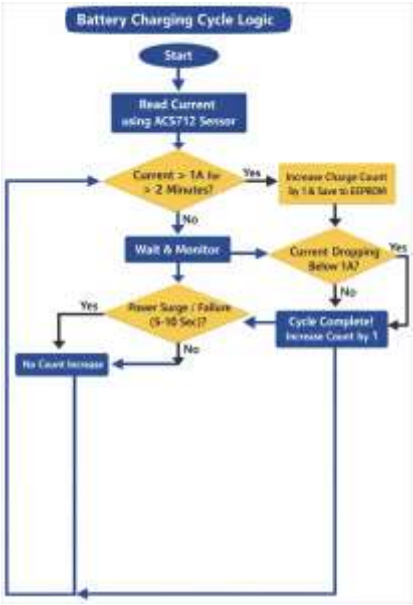
Figure 9: Hardware Prototype for E-Cycle Charging Monitoring

Algorithm 4: Charging-Cycle Counting and Persistence (M4)

1: Initialise CycleCount from non-volatile memory
2: Read Iin from the ACS712 sensor
3: if Iin > 1.0 A for more than 2 minutes then
4: CycleCount <- CycleCount + 1
5: store CycleCount in non-volatile memory
6: else if the interruption lasts 5 to 10 seconds then

7: do not increase CycleCount
8: end if
9: Monitor the declining current to confirm cycle completion

Figure 10: Logic Flowchart for Charging-Cycle Detection and Non-Volatile Storage



4.5 Module M5: Anti-Theft Protection, Live Tracking, and Alarm

Theoretical framework. This module is the final security layer and combines real-time geolocation with a physical deterrent. Its protection logic is defined by the state vector of Equation 4.

$$V_{state} = \{ L_{curr},\ D_{prox},\ I_{relay} \} \quad (4)$$

Here the current location is a coordinate pair, the proximity is the distance returned by the HC-SR04 ultrasonic sensor, and the relay term is the isolation state of the battery. Theft is inferred when the proximity exceeds a threshold, which indicates removal, and a cellular alert is then dispatched according to Equation 5.

$$\text{Alert} = f(L_{curr}) \rightarrow \text{mobile node via SMS} \quad (5)$$

Hardware implementation. An Arduino Uno serves as the central controller and interfaces with a NEO-6M receiver for coordinates over a software serial link, a SIM900A module for transmitting alerts, an HC-SR04 ultrasonic sensor for detecting the physical presence of the pack, and a relay that acts as an electronic lock to disconnect the supply during a breach. The components are listed in Table 5.

Table 5: Components Used for the Charging-Protection and Security System

Number	Component	Rating	Price
1	Battery	12 V, 28 Ah	3500 rupees
2	Arduino Uno microcontroller	ATmega328	450 rupees
3	Ultrasonic sensor	HC-SR04	250 rupees
4	Cellular module	SIM900A	800 rupees

Number	Component	Rating	Price
5	Positioning module	NEO-6M	300 rupees

Software logic. The ultrasonic sensor is polled continuously; when the measured proximity exceeds the configured limit the pack is assumed to have been moved or removed, whereupon the battery is isolated, the buzzer is triggered, and a message that carries the current coordinates is sent to the operator.

Figure 11: Functional Block Diagram of the Anti-Theft and Tracking Subsystem

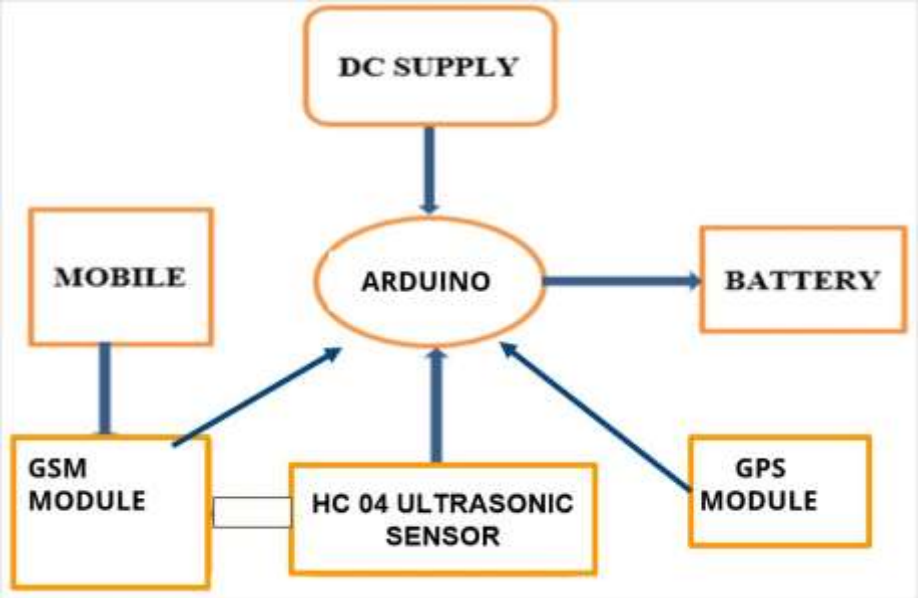


Figure 12: Wiring Schematic for the Relay, Battery, and Mobile Interface

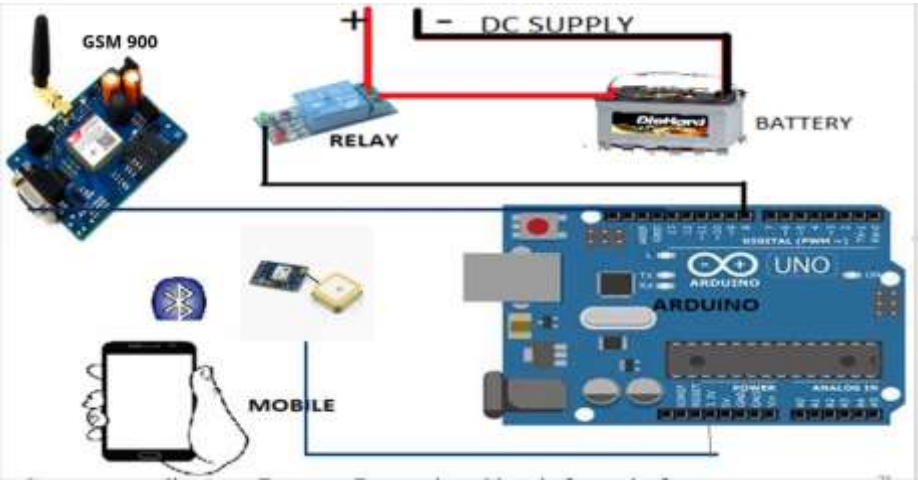
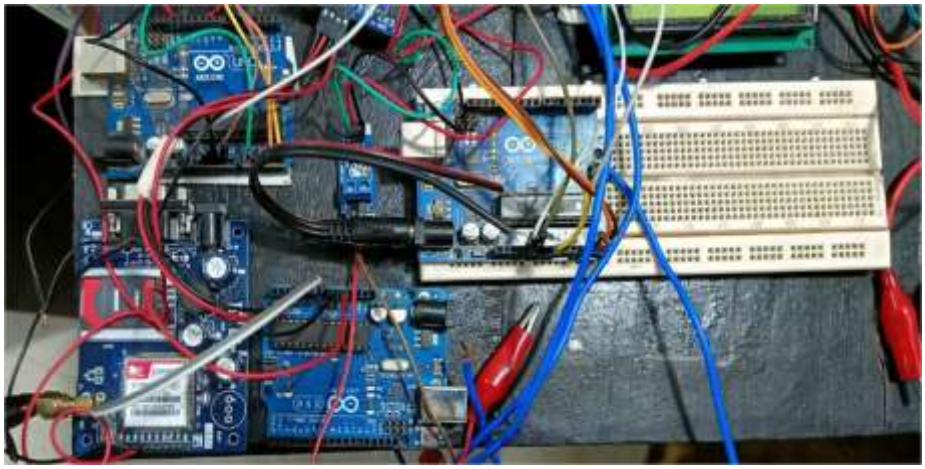


Figure 13: Hardware Setup Integrating Positioning, Cellular, and the Controller



Algorithm 5: Anti-Theft and Live Tracking (M5)

```
1: distance <- getUltrasonic(); coords <- getGPS()
2: if distance > SAFE_LIMIT or unauthorized_access == TRUE then
3:     setRelay(OFF)   // isolate the battery
4:     triggerBuzzer(ON)
5:     sendSMS("Alert: battery tamper detected. Location: " + coords)
6: else
7:     sendLocationUpdate(coords)   // optional real-time tracking
8: end if
```

4.6 Module M6: Integrated Swap Logic and Cloud Analytics

Hardware interfacing. A voltage-divider circuit connected to the battery monitors the voltage in real time, and the reading is shown locally on a display through an interface module. The components used for the integrated station are listed in Table 6.

Table 6: Components Used for the Integrated Swapping System

Number	Component	Rating
1	Battery	12 V, 28 Ah
2	Arduino Uno microcontroller	ATmega328
3	Voltage sensor (divider)	AVS 725
4	Display 16x2 with interface	I2C module
5	Interface software	MATLAB
6	Cloud software	ThingSpeak

Processing and pricing. The state of charge is extracted from the voltage reading by interpolation. When a pack is connected for more than two minutes, the life-cycle counter is incremented, and the state of health is then derived from the combined state-of-charge and life-cycle data inside the MATLAB environment. The total swap cost is finally computed from a fixed charge, the electricity consumed, and a

penalty factor determined by the state of health that is retrieved from previous ThingSpeak records. All parameters are uploaded to the cloud so that health and billing history can be tracked over the long term.

Figure 14: Functional Wiring of the Controller, Display, and Battery Interface

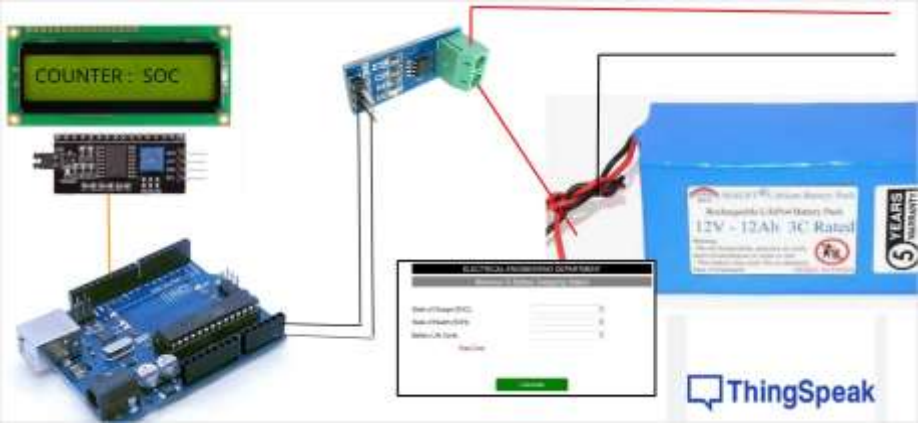
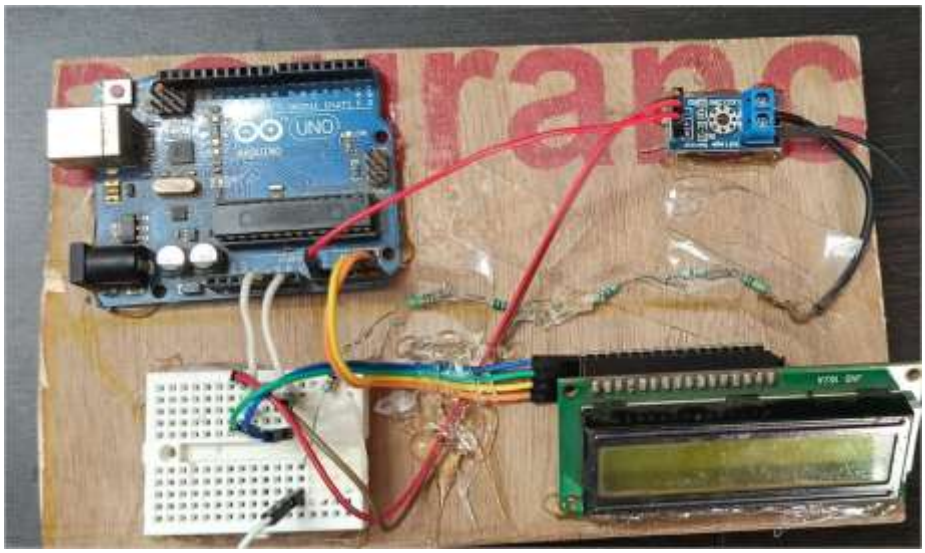


Figure 15: Physical Hardware Implementation on the Testing Board

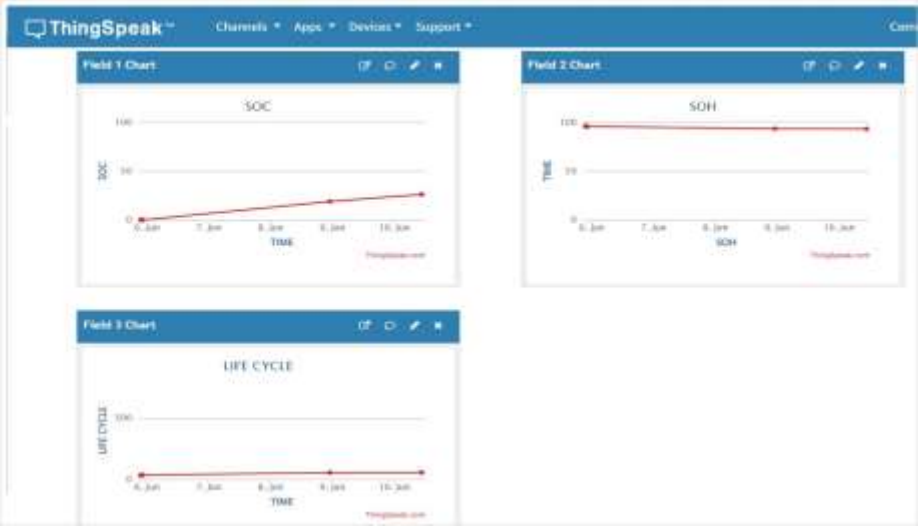


Algorithm 6: Integrated Swap Logic and Pricing (M6)

1: Acquire voltage from the AVS 725 sensor
2: Calculate state of charge by interpolation
3: if connection duration > 2 minutes then
4: increment the life-cycle counter
5: end if
6: Extract state of health through the MATLAB interface
7: Fetch previous records from ThingSpeak
8: Compute Price = FixedCharge + ElectricityCharge + Penalty(SOH)
9: Display metrics locally and upload them to ThingSpeak

Figure 16: MATLAB Interface for Real-Time Monitoring of State of Charge, State of Health, and Life Cycle

Figure 17: ThingSpeak Dashboard Showing Historical Trends of State of Charge, State of Health, and Life Cycle



5 Implementation Details: Bill of Materials and Pin Mapping

5.1 Consolidated Bill of Materials

A selection of low-cost, capable components supports the operational security and health traceability of the framework. The move to the ESP32 supplies native connectivity and higher-resolution sensing without a significant increase in cost. The consolidated bill of materials is presented in Table 7.

Table 7: Consolidated Bill of Materials for the Integrated ESP32-Based Framework

Component	Type or Rating	Indicative Cost (INR)
Battery pack	12 V LiFePO4 (28 Ah)	3500
ESP32 microcontroller	Dual-core, 240 MHz	525

Component	Type or Rating	Indicative Cost (INR)
Voltage sensor	AVS 725 divider	50
Current sensor	ACS712 (20 A module)	100
Bluetooth module	HC-05 (UART)	250
Relay module	5 V SPDT	150 to 250
Positioning module	NEO-6M	300
Cellular module	SIM900A	800
Ultrasonic sensor	HC-SR04	250
Display 16x2 with interface	I2C interface	190

5.2 Pin Mapping for the ESP32 Controller

The pin assignment shown in Table 8 exploits the multi-channel converter and hardware serial ports of the ESP32 so that sensing and security monitoring can proceed at the same time.

Table 8: Recommended ESP32 Pin Mapping for Integrated Station Control

Function	ESP32 Pin	Notes
Voltage sensing	GPIO 34 (ADC1)	12-bit converter input (M1)
Current sensing (ACS712)	GPIO 35 (ADC1)	Charge tracking (M4)
Relay control signal	GPIO 5	Path isolation (M2)
HC-05 transmit and receive	GPIO 17 and 16	Bluetooth authentication (M2)
Positioning transmit and receive	GPIO 27 and 26	Geolocation (M5)
Cellular transmit and receive	GPIO 4 and 2	Message alerts (M5)
Ultrasonic trigger and echo	GPIO 18 and 19	Theft detection (M5)
Display data line	GPIO 21	Serial data
Display clock line	GPIO 22	Serial clock

5.3 Power-Failure Resilience and Data Storage

A notable feature of the implementation is the use of the internal non-volatile storage of the ESP32 in place of a conventional external EEPROM. Cycle counts and authentication logs are consequently retained during a power failure, and the station resumes operation without any loss of historical health data.

6 Results and Performance Analysis

The tables that follow summarise representative performance for the six-module design and demonstrate the technical validation of the ESP32-based framework.

6.1 Voltage-Measurement Accuracy

Reconstruction accuracy for a 12 V LiFePO4 pack, obtained with the 12-bit converter of the ESP32 and the AVS 725 divider, is reported in Table 9. The absolute error remains within 0.01 V across the tested range.

Table 9: Representative Voltage-Reconstruction Accuracy with the 12-Bit Converter

True Voltage (V)	12-Bit Count	Estimated Voltage (V)	Absolute Error (V)
10.50	1074	10.49	0.01
11.95	1222	11.94	0.01
12.80	1309	12.79	0.01
13.45	1376	13.44	0.01
14.40	1473	14.39	0.01

6.2 Bluetooth Access Control and Relay Gating

The response of the password-matching logic and of the relay is summarised in Table 10. A correct password string produced a successful authorisation in every trial.

Table 10: Representative Bluetooth Authentication Performance

Metric	Minimum	Typical	Maximum
Pair and connect (s)	1.2	2.5	5.0
Password match (ms)	5	12	25
Relay gating delay (ms)	15	30	55
Authorisation success	100 percent for a correct password		

6.3 Current Filtering and Non-Volatile Persistence

The behaviour of the cycle counter and of the non-volatile storage is summarised in Table 11. Short interruptions and surges were rejected, while genuine charge events were counted and preserved across a reset.

Table 11: Representative Cycle Monitoring and Flash Persistence

Test Condition	Outcome	Observation
Power failure under 10 s	Ignored	No false increment
Charge event over 2 min	Passed	Count incremented and stored
Reset after 50 cycles	Passed	Count restored from storage
Surge above 1.5 A	Filtered	Temporal logic prevented a false hit

6.4 Anti-Theft Response Latency

The delay between a detected tamper event and the receipt of a location alert is reported in Table 12, together with the positioning precision.

Table 12: Representative Security-Subsystem Latency

Security Event	Detection to Message	Positioning Precision
Battery removal	3.5 to 5.0 s	plus or minus 2.8 m
Geofence breach	4.0 to 6.2 s	plus or minus 3.1 m
Unauthorised movement	3.8 to 5.5 s	plus or minus 3.0 m

6.5 Integrated Pricing with a Health Penalty

For a pricing policy in which the fixed charge is 50, the electricity charge is 0.8 times the quantity (100 minus the state of charge), and the penalty is 0.5 times the quantity (100 minus the state of health), the computed decisions and health-aware prices are given in Table 13.

Table 13: Integrated Decisions and Computed Health-Aware Swap Price (INR)

Authentication	Voltage (V)	Life Cycles	SOC (%)	SOH (%)	Decision and Total Price
Fail	12.8	56	75	93	Deny (unauthorised)
Pass	10.2	10	2	99	Deny (under-voltage)
Pass	12.1	800	45	25	Deny (health reject)
Pass	12.8	200	75	75	Approve, 50 + 20 + 12.5 = 82.5
Pass	13.2	40	90	95	Approve, 50 + 8 + 2.5 = 60.5

7 Discussion: Scaling the Framework Toward Industrial Stations

7.1 From Relay to Contactor and Safety Interlocks

At industrial power levels the prototype relay must be replaced by a direct-current contactor together with a pre-charge circuit and interlock loops. The decision chain is unchanged, since M2 authorises, M6 gates, and M3 logs; only the switching device is upgraded.

7.2 Measurement Hardening and Isolation

Industrial systems require galvanically isolated voltage and current sensing, wiring and filtering that resist electromagnetic interference, and sensing that is calibrated across temperature. Modules M1 and M3 nevertheless remain valuable as design patterns for scaling, reconstruction, and persistence.

7.3 Improving State-of-Charge and State-of-Health Accuracy

Voltage-based state of charge is sensitive to load and chemistry. Industrial scaling should therefore incorporate telemetry from the pack management system, coulomb counting, model-based estimation such as a Kalman filter, and temperature compensation. Module M4 still serves as an operator-facing estimator and as a point at which policy is enforced.

7.4 Security Beyond Simple Identifier Matching

Matching a stored identifier is adequate for a controlled prototype but not for a hostile setting. Suitable upgrades include cryptographic tags, rolling-token schemes, server verification with an offline fallback, and tamper detection accompanied by signed logs. Module M2 provides the physical enforcement point, while the credential method can be strengthened independently.

7.5 Lifecycle Logging at Scale

EEPROM logging can be replaced by higher-endurance storage such as ferroelectric memory, a journalled memory card, or a secure cloud-backed log with local buffering. The edge-triggered write policy retained from the modules provides a baseline for event logging.

7.6 The Alert Layer as an Independent Safety Channel

Keeping the alert function on a minimal, separate node improves resilience. In an industrial station this layer can be extended to stack lights and sirens, to fail-safe shutdown signals, and to watchdog supervision that operates independently of the interface.

8 Limitations and Engineering Constraints

State of charge inferred from voltage is an approximation whose accuracy depends on chemistry and rest time. State of health inferred from cycle count is a proxy that does not capture the variability of depth of discharge or the influence of temperature. EEPROM endurance demands a careful write policy, because frequent writes without buffering shorten its life. Prototype wiring must also be hardened for field deployment through strain relief, insulation, and control of electromagnetic interference.

9 Conclusion

A technically dense, module-integrated, and low-cost embedded framework has been presented for mitigating critical operational threats in next-generation battery swapping stations. The contribution is a complete chain from threat to implementation: calibrated voltage acquisition (M1) supports robust state-of-charge inference; Bluetooth access control (M2) enforces physical authorisation through password verification; and migration to the ESP32 (M3) provides high-speed dual-core processing together with native connectivity. Cycle monitoring based on the ACS712 (M4) delivers auditable life-cycle tracking with flash persistence, the combined positioning, cellular, and ultrasonic suite (M5) provides proactive anti-theft protection and live tracking, and the fusion of a MATLAB interface with ThingSpeak (M6) converts raw measurements into operator-grade decision variables for transparent, health-aware pricing. By supplying pin-level detail, mathematical derivations for interpolation-based sensing, and algorithmic flows for real-time security, the framework narrows the gap between conceptual literature and a deployable prototype. Future work will pursue predictive health analytics through artificial intelligence, an upgrade of the Bluetooth layer toward cryptographic mesh networking, and validation under high-current industrial charging with formal hardware-in-the-loop safety interlocks.

Acknowledgement

The authors thank Gujarat Technological University and Vishwakarma Government Engineering College for the laboratory infrastructure and support provided for this research.

References

1. M.H.S.M. Haram, J.W. Lee, G. Ramasamy, E.E. Ngu, S.P. Thiagarajah, Y.H. Lee, "Feasibility of Utilising Second Life EV Batteries: Applications, Lifespan, Economics, Environmental Impact, Assessment, and Challenges", Alexandria Engineering Journal, October 2021, 60 (5), 4517.
2. M.H.S.M. Haram, M.T. Sarker, G. Ramasamy, E.E. Ngu, "Second Life EV Batteries: Technical Evaluation, Design Framework, and Case Analysis", IEEE Access, 2023, 11, 138799-138812.
3. C. Madden, J. Peskar, K. Sado, A.R.J. Downey, J. Khan, "Electro-Thermal Hardware-in-the-Loop Ba-

ttery Emulator for Shipboard Systems Testing", 2025 IEEE Electric Ship Technologies Symposium, 2025, 47-52.
4. S. Lale, M. Basic, S. Lubura, B. Popovic, M. Ikic, "Current-Mode Controlled Battery Emulator", Processes, 2025, 13 (10), 3281.
5. A. Verani, R. Di Rienzo, N. Nicodemo, F. Baronti, R. Roncella, R. Saletti, "Open Hardware and Software Modular Battery Emulator for Battery Management Systems Development and Functional Testing", IEEE Access, 2024, 12, 84488-84497.
6. J. Linru, Z. Yuanxing, L. Taoyong, D. Xiaohong, Z. Jing, "Analysis on Charging Safety and Optimization of Electric Vehicles", 2020 IEEE 6th International Conference on Computer and Communications, 2020, 2382-2385.
7. Q. Li and others, "Optimizing Charging Strategies for Electric Vehicles in Residential Areas Based on a Leaped Halton Sequence Initialized Flood Algorithm", 2025 IEEE 8th Information Technology and Mechatronics Engineering Conference, 2025, 1513-1517.
8. Y. Deng and others, "Feature Parameter Extraction and Intelligent Estimation of the State of Health of Lithium-Ion Batteries", Energy, 2019, 176, 91-102.
9. Y. Zheng and others, "An Accurate Parameter Extraction Method for a Novel On-Board Battery Model Considering Electrochemical Properties", Journal of Energy Storage, 2019, 24, 100745.
10. O. Theliander, A. Kersten, M. Kuder, W. Han, E.A. Grunditz, T. Thiringer, "Battery Modeling and Parameter Extraction for Drive-Cycle Loss Evaluation of a Modular Battery System Based on a Cascaded H-Bridge Multilevel Inverter", IEEE Transactions on Industry Applications, November 2020, 56 (6), 6968-6977.
11. M. Bayati, N. Tashakor, P.P. Abkenar, S. Goetz, "Highly Compact Charging with Power-Factor Correction for Electric Vehicles Through Functional Integration into a Dynamically Reconfigurable Battery", IEEE Transactions on Transportation Electrification, 2025, 11 (4), 10276-10285.
12. R. Di Fonso, R. Teodorescu, P. Bharadwaj, "Aging-Aware Equivalent Circuit Model for State-of-Health Estimation in Lithium-Ion Batteries", 2024 IEEE International Telecommunications Energy Conference, 2024, 1-5.
13. N. Tashakor, P. Pourhadi, M. Bayati, M.H. Samimi, J. Fang, S.M. Goetz, "Modular Reconfigurable Mixed Battery System with Heterogeneous Modules", IEEE Transactions on Transportation Electrification, 2024, 10 (4), 8486-8497.
14. N. Blasuttigh, H. Beiranvand, T. Pereira, S. Castellan, M. Liserre, "Efficiency Trade-off-Oriented Analysis for the Integration of a DC-DC Converter and Battery Pack in Vehicle-to-Grid Applications", 2022 IEEE Energy Conversion Congress and Exposition, 2022, 1-7.
15. J. Brand, Z. Zhang, R.K. Agarwal, "Extraction of Battery Parameters of the Equivalent Circuit Model Using a Multi-Objective Genetic Algorithm", Journal of Power Sources, 2014, 247, 729-737.
16. S. Ali and others, "Advancing Lithium-Ion Battery Management: A Portable Arduino-Based Health-Monitoring Solution", Kashf Journal of Multidisciplinary Research, December 2024, 1 (12), 1-10.
17. T. Sathapornbumrungpao and others, "Battery Management System Using a Relay Contactor by an Arduino Controller for a Lithium-Ion Battery", Proceedings of the 5th International Conference on Clean Energy and Electrical Systems, Springer, Singapore, 2023, 1058, 113-124.
18. M. Aydin, I. Gurbuz, "IoT-Based Low-Cost Battery Monitoring System Using the ESP8266 and the Arduino IoT Cloud Platform", International Journal of Automotive Engineering and Technologies, 2024, 13 (4), 170-179.