IOT Based Battery Management System for Electrical Vehicles Using Raspberry Pi Pico

K. Kartheek[1], K.Sreeja[2], M.Nikhil[3], G. Hima Bindu[4]

[1]Assistant Professor, Dept. of EEE, Matrusri Engineering College, Hyderabad.
[2,3,4]Students, Dept. of EEE, Matrusri Engineering College, Hyderabad

Abstract

This paper presents the design and implementation of an IoT-based Battery Management System (BMS) for monitoring and controlling a 12 V battery using a Raspberry Pi Pico W. The system measures key parameters such as voltage, current, and temperature in real time using appropriate sensors and processes the data through an embedded controller. An IoT platform is integrated to enable remote monitoring, data logging, and visualization of battery performance. The proposed system incorporates protection mechanisms against overvoltage, overcurrent, and over temperature conditions to ensure safe and reliable operation. Additionally, State of Charge (SoC) and State of Health (SoH) are estimated to provide deeper insight into battery status and performance. Experimental results demonstrate accurate monitoring, efficient control, and reliable communication with minimal delay. The developed system is low-cost, scalable, and suitable for applications in electric vehicles and energy storage systems, offering an effective solution for intelligent battery management.

Keywords: Battery Management System (BMS), Internet of Things (IoT), Raspberry Pi Pico W, State of Charge (SoC), State of Health (SoH), Voltage Monitoring, Current Sensing, Temperature Monitoring, ThingSpeak, Electric Vehicles (EVs)

1. Introduction

The growing demand for green energy solutions has positioned electric vehicles (EVs) as a key technology for sustainable transportation. Lithium-ion batteries are widely used in EV applications due to their high energy density and current carrying capability. However, these batteries must operate within a defined Safety Operating Area (SOA) to prevent issues such as overheating, degradation, and potential failure. Therefore, an efficient Battery Management System (BMS) integrated with Internet of Things (IoT) technology is essential for ensuring safe and optimal battery performance. This work presents the design and implementation of an IoT-based BMS using the Raspberry Pi Pico W as the central controller. The system integrates voltage, current, and temperature sensors along with relays, a buzzer, cooling fan, LCD module, and a 12 V battery pack with a charging setup. It continuously monitors critical battery parameters including voltage, temperature, State of Charge (SoC), and State of Health (SoH). When the temperature exceeds a predefined threshold, the system activates a cooling fan and buzzer to ensure safety. The measured data is transmitted to the ThingSpeak cloud platform for real-time monitoring and analysis. Additionally, the system provides flexibility in charging by allowing users to select different charging modes through switches. The use of relays enables controlled charging and discharging, improving battery efficiency and lifespan. Both simulation and experimental results validate the effectiveness of the proposed system. The developed IoT-enabled BMS enhances safety, reliability, and performance, making it suitable for applications in electric vehicles and energy storage systems. Furthermore, the system provides a foundation for future advancements in intelligent battery monitoring and control.

2. Literature Review

T. Sirisha et al. [1] emphasize the importance of battery monitoring in EVs and propose a BMS that ensures safety and performance through continuous monitoring of parameters such as voltage and temperature. The study employs Coulomb counting for State of Charge (SoC) estimation and CCCV methods for State of Health (SoH), along with IoT-based data logging using ThingSpeak.

Raj Patel et al. [2] provide a comprehensive overview of BMS technologies, highlighting key functions such as monitoring, charge control, thermal management, and safety mechanisms. The study identifies core BMS components including sensing units, charging algorithms, and balancing circuits for reliable operation.

Muhammad Nizam et al. [3] review BMS design challenges for lithium-ion batteries and discuss various charging strategies and advanced techniques such as neural networks for battery life prediction, focusing on improving performance and durability.

Harish N. et al. [4] propose an IoT-based BMS that monitors battery parameters and disconnects the system under abnormal conditions, demonstrating improved safety and reliability.

S. Prabakaran et al. [5] develop an IoT-enabled battery monitoring system with real-time alerts and cloud-based analytics, enhancing fault detection, safety, and maintenance efficiency.

3. System Architecture / Block Diagram Description

The proposed system is an IoT-based Battery Management System (BMS) designed for real-time monitoring, protection, and control of a 12 V battery using a Raspberry Pi Pico W. It consists of power supply, sensing, control, and communication units.

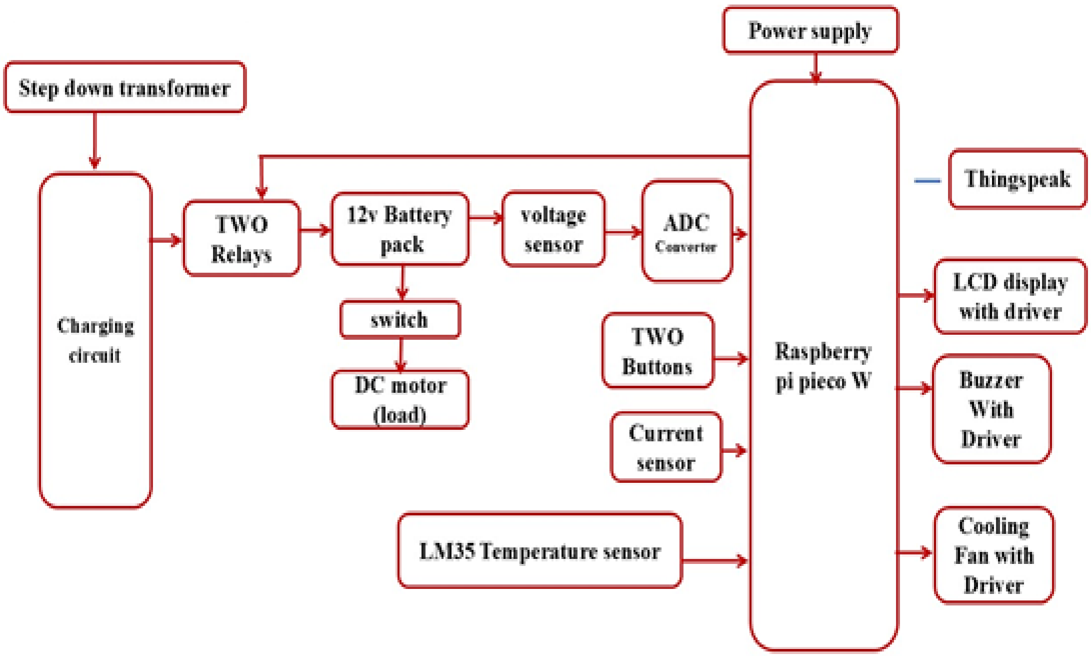


Figure 1: Block Diagram

3.1 Power Supply and Charging Unit

The AC input is stepped down, rectified, filtered, and regulated to obtain a stable DC output for battery charging. Relays are used to control charging and provide protection against overcharging.

3.2 Battery and Load Section

A 12 V battery serves as the energy source and is connected to a DC motor load through a switch, enabling controlled charging and discharging.

3.3 Sensing and Data Acquisition Unit

Voltage, current, and temperature sensors are used to monitor battery parameters. An ADC converts analog signals into digital form for processing by the controller.

3.4 Control Unit

The Raspberry Pi Pico W processes sensor data and performs control actions based on predefined limits, including relay operation, cooling activation, and fault alerts. Push buttons enable manual control.

3.5 Output and Protection Unit

An LCD displays real-time parameters (voltage, current, temperature, SoC, SoH). A buzzer provides alerts, and a cooling fan maintains safe operating temperature.

3.6 IoT Communication Unit

The system uses Wi-Fi to transmit data to the ThingSpeak cloud for real-time monitoring, visualization, and data logging.

3.7 Overall Operation

The system continuously monitors battery parameters, processes data, and executes control actions to ensure safe operation. Data is displayed locally and uploaded to the cloud for remote access, enhancing reliability and performance.

4. Hardware Design

4.1 Raspberry Pi Pico W

The Raspberry Pi Pico W acts as the central controller, providing processing capability and built-in Wi-Fi for IoT communication. It interfaces with sensors, control devices, and the cloud platform for real-time monitoring and control.

Specifications:

- Processor: ARM Cortex-M0+
- Operating Voltage: 5 V
- GPIO Pins: 40
- Connectivity: Wi-Fi (802.11 b/g/n)
- Operating Temperature: –20°C to 70°C

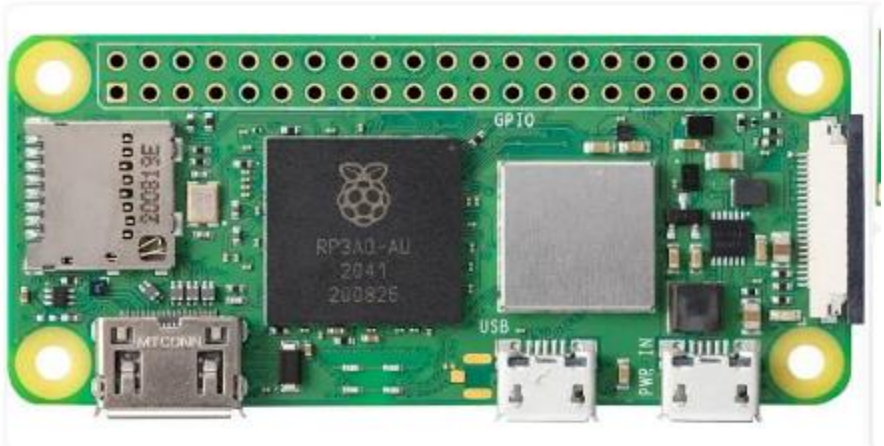


Figure 2: Raspberry pi pico W

4.2 Sensors Used

The system employs multiple sensors for accurate battery monitoring:

- **Voltage Sensor:** Uses a voltage divider to measure battery voltage for SoC estimation and protection.
- **Current Sensor:** A Hall-effect sensor (e.g., ACS712) measures charging/discharging current and detects overcurrent conditions.
- **Temperature Sensor (LM35):** Monitors battery temperature and ensures thermal safety through cooling or shutdown mechanisms.

4.3 Power Supply and Charging Circuit

A step-down transformer, rectifier, filter, and voltage regulator are used to convert AC supply into regulated DC for battery charging and system operation. This ensures stable and reliable power delivery.

4.4 Control and Switching Components

- **Relays:** Used to control charging/discharging and provide protection against overvoltage and fault conditions.
- **Relay Driver Circuit (Transistor-based):** Interfaces relays with the microcontroller.
- **Switches/Push Buttons:** Enable manual control for system operation and mode selection (e.g., charging modes).

4.5 Output and Protection Components

- **LCD Display (16×2) with Driver:** Displays real-time parameters such as voltage, current, temperature, SoC, and SoH.
- **Buzzer with Driver:** Provides audible alerts during abnormal conditions.
- **Cooling Fan:** Maintains safe temperature levels by dissipating heat.

4.6 Data Acquisition and Interface

- **Analog-to-Digital Converter (ADC):** Converts analog sensor signals into digital form for processing.
- **Signal Conditioning Circuits:** Ensure accurate and noise-free sensor readings.

4.7 Communication Module

- **Wi-Fi Communication:**
  The built-in Wi-Fi of the Raspberry Pi Pico W enables real-time data transmission.
- **IoT Platform Integration:**
  ThingSpeak is used for cloud-based data storage, visualization, and analysis, allowing remote monitoring and performance tracking.

4.8 Battery and Load

A 12 V battery pack is used as the energy source, connected to a DC motor load. This setup enables testing under real charging and discharging conditions.

5. Software Design

5.1 Programming (MicroPython / Python)

The proposed BMS is implemented using MicroPython on the Raspberry Pi Pico W, enabling efficient hardware interfacing, rapid development, and real-time control.

- **Development Environment:**

  Thonny IDE is used for coding, debugging, and firmware uploading.

- **Data Acquisition:**

  Sensor signals (voltage, current, temperature) are read via ADC and converted into calibrated digital values.

- **Control Algorithm:**

The controller compares sensor values with predefined limits and performs actions such as relay control, cooling fan activation, and buzzer alerts to ensure safe operation.

- **User Interface:**

  Push buttons allow manual control, while an LCD displays real-time parameters.

- **IoT Communication:**

  Data is transmitted to the cloud using Wi-Fi via HTTP/MQTT protocols for remote monitoring.

- **Program Flow:**

  Initialization → Wi-Fi setup → Data acquisition → Processing → Control actions → Display → Cloud upload (continuous loop).

- **Advantages:**
  Simple programming, fast development, low resource usage, and easy IoT integration.

5.2 IoT Platform (ThingSpeak, Blynk, etc.)
IoT platforms enable real-time monitoring, data logging, and remote access of battery parameters.

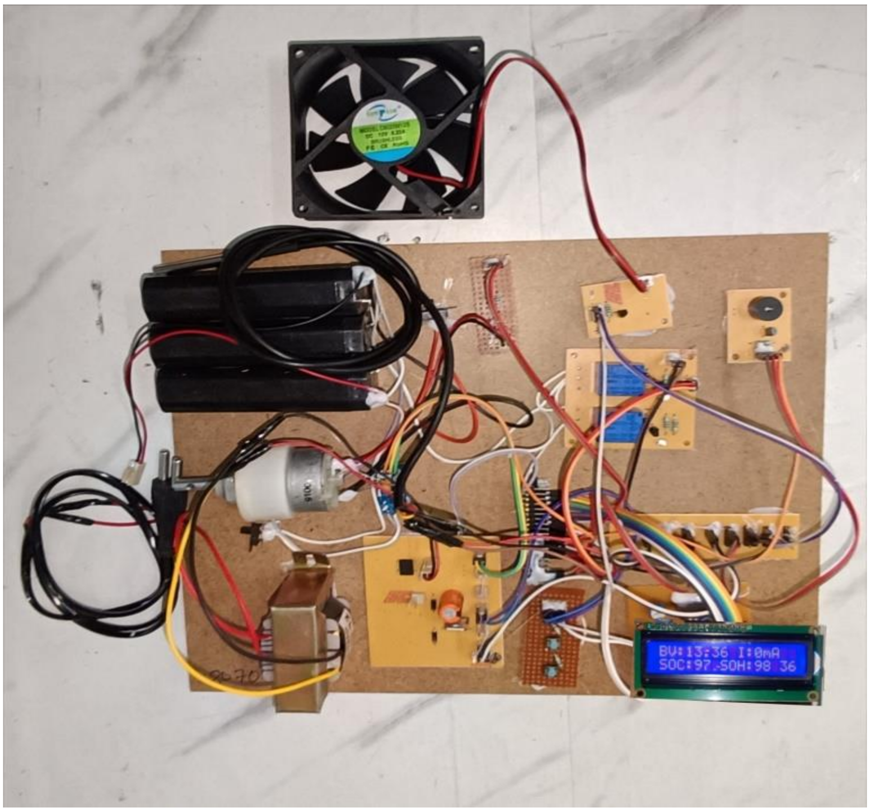
- **ThingSpeak:**
  Used for cloud-based data storage and visualization. Battery parameters are uploaded via HTTP and displayed as time-based graphs. It also supports MATLAB analytics for advanced analysis.
- **Blynk (Optional):**

  Provides a mobile-based interface for real-time monitoring and control with customizable dashboards and alerts.

- **Key Features:**

  Real-time visualization, historical data storage, remote access, and support for predictive analysis.

6 Prototype setup



7. Results and Discussion

The proposed IoT-based Battery Management System (BMS) was implemented and tested under charging, discharging, and fault conditions to evaluate monitoring accuracy, protection capability, and communication performance.

7.1 Real-Time Monitoring

The system successfully monitored voltage, current, temperature, State of Charge (SoC), and State of Health (SoH), with data displayed locally and transmitted to the IoT platform with minimal delay.

- **Voltage:** Maintained within 10–14.5 V, showing expected charging and discharging behavior.
- **Current:** Accurately reflected load variations during operation.
- **Temperature:** Slight increase under load, confirming effective thermal monitoring.
- **SoC:** Decreased during discharge and increased during charging, following expected trends.
- **SoH:** Remained relatively stable, indicating healthy battery condition with minor degradation over time.

7.2 IoT Data Visualization

All parameters were visualized as time-based graphs on the IoT platform. Voltage and SoC showed consistent charge/discharge patterns, current reflected load variations, and temperature remained within safe limits. SoH trends indicated gradual aging. Historical data storage enabled performance analysis and monitoring.

7.3 Protection Mechanisms

The system effectively handled abnormal conditions:

- **Overvoltage:** Charging was disconnected using relays.
- **Overcurrent:** Protective action prevented circuit damage.
- **Overtemperature:** Cooling fan and buzzer were activated.

Overall, the system demonstrated accurate monitoring, reliable protection, and effective SoC and SoH estimation, validating its suitability for intelligent battery management applications.

8. Advantages of the Proposed System

- Real-time monitoring of voltage, current, temperature, SoC, and SoH
- Enhanced safety with overvoltage, overcurrent, and thermal protection
- IoT-based remote monitoring and data visualization
- Cloud data logging for analysis and diagnostics
- Low-cost and compact design
- Scalable for advanced and larger applications

- Simple implementation using MicroPython
- Extended battery life through controlled operation
- Intelligent management using SoC and SoH estimation
- User-friendly interface via LCD and IoT dashboards

9. Conclusion

The proposed IoT-based BMS using Raspberry Pi Pico W enables real-time monitoring, protection, and control of a 12 V battery system. It accurately measures key parameters and integrates IoT for remote access and visualization. Protection mechanisms ensure safe operation, while SoC and SoH estimation enhance system intelligence.

Experimental results confirm reliable performance, fast response, and accurate monitoring. The system is low-cost, scalable, and suitable for electric vehicles and energy storage applications.

References

1. T. Sirisha, et al., “Battery Management System for Electric Vehicles: A Review,” International Journal of Engineering Research & Technology (IJERT), vol. X, no. X, pp. XX–XX, 2023.
2. P. Ramadass, B. Haran, R. White, and B. N. Popov, “Performance study of commercial Li-ion batteries,” Journal of Power Sources, vol. 123, no. 2, pp. 230–240, 2003.
3. H. He, R. Xiong, and J. Fan, “Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation,” Energies, vol. 4, no. 4, pp. 582–598, 2011.
4. Texas Instruments, “ACS712 Hall-Effect Current Sensor Datasheet,” 2019.
5. Analog Devices, “LM35 Precision Temperature Sensor Datasheet,” 2018.
6. Raspberry Pi Foundation, “Raspberry Pi Pico W Datasheet,” 2022.
7. MathWorks, “ThingSpeak IoT Analytics Platform,” [Online]. Available: https://thingspeak.com
8. Blynk Inc., “Blynk IoT Platform Documentation,” [Online]. Available: https://blynk.io
9. A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu, “A review on lithium-ion battery ageing mechanisms and estimations for automotive applications,” Journal of Power Sources, vol. 241, pp. 680–689, 2013.
10. S. Piller, M. Perrin, and A. Jossen, “Methods for state-of-charge determination and their applications,” Journal of Power Sources, vol. 96, no. 1, pp. 113–120, 2001.