IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

SMART BATTERY MANAGEMENT FOR SOLAR POWER WITH CLOUD ANALYTICS

Y. Santhosh, Assistant professor, S. Soniya , K. Mohan Rao, T. Damodararao, K. Chakravarthi, A. Prudvi, students,
Satya Institute of Technology and Management

***Abstract*— Solar energy systems depend on a Battery Management System (BMS) to keep energy storage components safe, efficient, and reliable. Environmental variables determine intermittent solar power generation. A battery stores extra energy for later use. The BMS tracks voltage, SOC, and SOH to regulate the battery pack. It prevents overcharging, over discharging, overheating, and short circuits, extending battery life and performance. The BMS balances cells to provide consistent charge distribution, minimizing deterioration and optimizing storage efficiency. BMS units interface with smart controllers and CAN or Modbus protocols for remote monitoring, diagnostics, and predictive maintenance in modern solar systems. This integration enables real-time energy optimization in off-grid or hybrid solar systems. Solar PV systems need the BMS for energy reliability, system stability, and optimal performance. Solar energy storage solutions will be most effective with modern BMS technology as renewable energy demand rises.**

Keywords*— *Battery Management System , battery, SOC, SOH

I. INTRODUCTION

The increasing global demand for sustainable energy has significantly accelerated the integration of solar power systems into both residential and industrial sectors. Central to the efficiency and longevity of such systems is the battery, which requires intelligent management to operate optimally. A Smart Battery Management System (BMS) serves as a critical component in solar energy setups, ensuring real-time monitoring, efficient charging/discharging, and safeguarding against battery failures. Over the years, researchers have explored various improvements to traditional BMS architectures by incorporating advanced sensing, control algorithms, and communication protocols.

An enhanced method for managing energy storage in solar-powered applications is provided by a Battery Management System (BMS) for solar power that is combined with cloud analytics provided by ThingSpeak. The battery management system (BMS) serves as the central hub, monitoring critical metrics such as voltage, current, temperature, and state of charge (SoC) in order to effectively oversee the functioning of the battery. As a result of its ability to guarantee safe operation, improve charging and discharging operations, and extend battery life, it is an indispensable component for solar energy systems that are dependable.

This system is able to get sophisticated real-time analytics and remote accessibility when it is integrated with ThingSpeak, which is an Internet of Things cloud platform. It is possible to visualize the data that has been collected from the BMS by utilizing dynamic charts, graphs, and dashboards that are stored in the cloud that is provided by ThingSpeak. Users are able to monitor the functioning of the system, analyze patterns, and obtain insights that can be put into action from any location in the world thanks to this integration. The configuration is also capable of predicting the behavior of the battery, improving energy efficiency, and providing alarms for possible problems, thanks to features such as MATLAB analytics on ThingSpeak capabilities. In [1] examined the application of Internet-of-things (IoT) in monitoring the performance of electric vehicle battery. the idea of monitoring the performance of the vehicle using IoT techniques is proposed. In [2,3] demonstrated Thus the parameters such as the voltage, State of Charge (SoC) of the battery is obtained and monitored so as to obtain a greater lifespan of the battery. The model of the battery are simulated with its charging & discharging properties using Matlab and the parameters of the battery such as SOC, voltage and current are obtained. In [4,5] developed lead acid batteries are one of the most commonly used batteries. Monitoring the performance parameters of the battery provides essential data useful for managing the batteries efficiently. This paper proposes a highly accurate and efficient battery monitoring system based on NI LabVIEW. As the calculations are done on computer, this system is faster than micro-controller based systems and provides highly reliable real-time data. In [6-8] proposed Experimental Validation of a Battery Dynamic Model for EV pplications. In [9-11] Development of an on-board charge and discharge management system for electric-vehicle batteries

This solution is perfect for applications such as solar installations that are not connected to the grid, energy storage in the home, or research on renewable energy sources. By combining the accuracy of a building management system (BMS) with the adaptability of cloud analytics provided by ThingSpeak, it provides a method that is intelligent, efficient, and scalable for effectively managing solar power installations. The following describes the structure of this paper:

In Section–I represent Introduction to smart management systems, followed by the battery management system in Section–II. Solar charge controller is explained in Section-III, Section–IV deals with smart battery management system for solar power with cloud analytics, simulation and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

experimentation are discussed in Section–V. Finally, this paper comes to an end with conclusions in Section–VI.

II. BATTERY MANAGEMENT SYSTEM

The efficacy of a rechargeable battery or battery pack is monitored and regulated by a Battery Management System (BMS), an intelligent electronic control system. A battery management system (BMS) is essential for the safety, longevity, and optimal performance of contemporary energy storage systems, particularly those that utilize lithium-ion chemistry. The necessity of a robust and intelligent BMS is becoming more apparent as the adoption of renewable energy systems, portable electronics, industrial-grade storage solutions, and electric vehicles (EVs) continues to increase. The primary responsibility of a battery management system (BMS) is to monitor the critical parameters of a battery, including cell voltage, current flow (charging or discharging), temperature, and, in certain instances, internal resistance or status of health (SOH). The BMS guarantees that the battery operates within its secure working limits by consistently collecting and analyzing this data. This encompasses safeguarding the battery from conditions that may result in degradation, failure, or safety hazards, including overvoltage, under voltage, overcurrent, and extreme temperatures.

III. SOLAR CHARGE CONTROLLER

A solar charge controller is essentially a device that regulates voltage or current to charge the battery and prevents electric cells from overcharging. It directs the voltage and current generated by the solar panels to the electric cell. Typically, 12V boards or panels output approximately 16 to 20V; however, in the absence of regulation, the electric cells may sustain damage due to overcharging. Typically, electronic storage devices necessitate approximately 14 to 14.5V for full charging. Solar charge controllers are offered in various features, prices, and dimensions. The spectrum of charge controllers spans from 4.5A to 60 or 80A. The solar charge controller is shown in fig.1.

Categories of Solar Charger Controllers:

There are three distinct categories of solar charge controllers:

1. Basic one or two-stage controls
2. Pulse Width Modulation (PWM)
3. Maximum Power Point Tracking (MPPT)

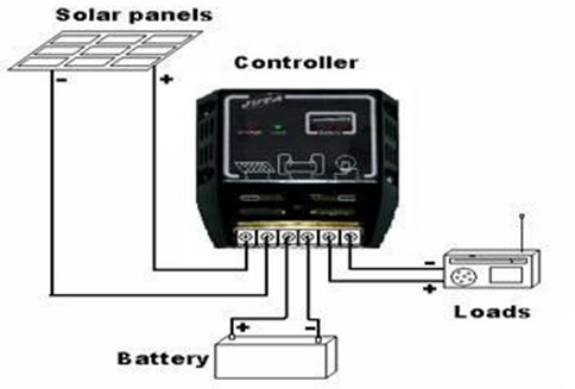

Fig1. Solar Charge Controller

IV. SMART BATTERY MANAGEMENT SYSTEM FOR SOLAR POWER WITH CLOUD ANALYTICS

The Smart BMS for Solar Power with Cloud Analytics represents a significant advancement in energy storage management. It leverages cloud computing, IoT sensors, and machine learning algorithms to transform raw battery data into actionable insights. These insights allow for intelligent decision-making, ensuring that batteries operate within optimal conditions and deliver maximum performance throughout their lifespan. By integrating cloud analytics, the Smart BMS can collect and analyze vast amounts of data from multiple battery systems across various locations.

This centralized data processing enables predictive analytics, such as forecasting battery degradation, identifying performance anomalies, and scheduling maintenance before failures occur. Moreover, cloud-based dashboards provide stakeholders ranging from homeowners and technicians to utility companies and system integrators with real-time visibility into the health, status, and efficiency of their battery systems. Block Diagram of Smart Battery Management System for Solar Power and Cloud Analytics is represented in fig 2.

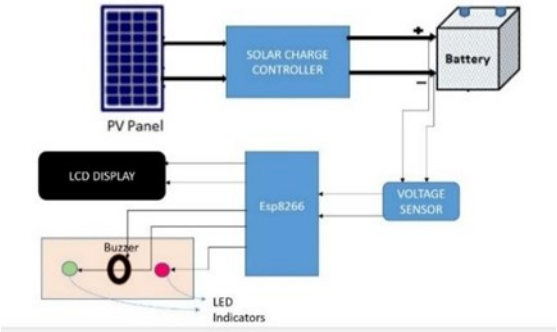

Fig 2. Block Diagram of Smart Battery Management System for Solar Power and Cloud Analytics

A. Components

Solar Charge Controller: This device regulates the power coming from the PV panel to charge the battery safely and efficiently. It prevents overcharging or deep discharging of the battery.

Battery: Stores the electrical energy generated by the PV panel for later use, especially when there's no sunlight.

ESP8266: A microcontroller with Wi-Fi capabilities, often used for IoT projects. It acts as the brain of the system, processing data and controlling other components.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

Voltage Sensor: Measures the battery voltage to monitor its state of charge.
LCD Display: Displays information, likely the battery voltage or system status.
Buzzer: An audio indicator that can alert the user to specific conditions (e.g., low battery).

LED Indicators: Visual indicators to show the system's status (e.g., power on, charging, or error states).

B. Workflow:

- PV Panel → Solar Charge Controller → Battery (stores energy).
- Battery → Powers ESP8266, LCD Display, Buzzer, and LED Indicators. Voltage Sensor → Measures battery voltage → Sends data to ESP8266.
- ESP8266 → Processes voltage data → Displays on LCD, controls Buzzer and LED Indicators, and possibly sends data over Wi-Fi.A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).

V. SIMULATION AND HARDWARE RESULTS

A. SIMULATION

The simulation Circuit of Smart Battery Management for Solar Power with Cloud Analytics is represented in fig 3. The components are the 10Wp PV panel output to the 12-24volts rated solar charge controller, we provide input from the solar charge controller to the 14.4-15Volts battery. From the battery, we connect to the input of the 25volts input/5volts output voltage sensor. The voltage sensor output is connected to the Arduino UNO ESP8266 as follows: the positive is connected to the 5V pin, the negative is connected to the ground pin, and the S pin is connected to A0. For the LCD I2C module, we connect four pins: the ground pin is connected to VIN, VCC is connected to 5V, SDA is connected to D14/SDA/D4, and SCL is connected to D15/SCL/D3. The buzzer is connected to ground, the red LED indicator is connected to D13/SCK/D5 cited, and the blue indicator is connected to D12/MISO/D6."

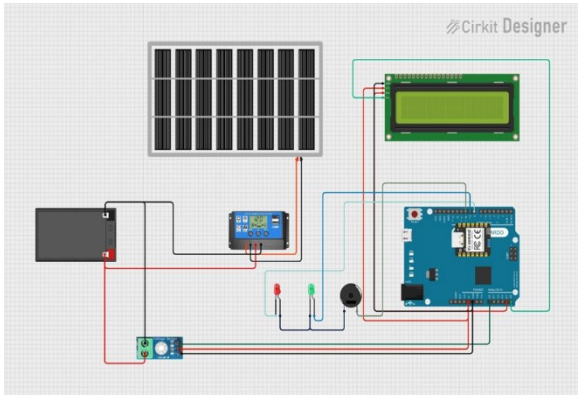

Fig3. Simulation Circuit of Smart Battery Management for Solar Power with Cloud Analytics

ARDUINO CODE

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h> #include <ESP8266WiFi.h> #include <WiFiClient.h> #include <ThingSpeak.h>

// 📌 LCD I2C Address

LiquidCrystal_I2C lcd(0x27, 16, 2); // Use 0x3F if needed

// Pin Definitions

#define VOLTAGE_SENSOR A0 #define BUZZER D5 // GPIO14 #define GREEN_LED D6 // GPIO12

// Wi-Fi & ThingSpeak Configuration const char* ssid = "12345678";

const char* password = "12345678"; unsigned long channelID = 2904558;

const char* apiKey = "LQ8ENHZSTDJ266RW";

// Variables

float voltage = 0.0; float SOC = 20.0;

int chargeCycles = 0; float SOH = 0.0;

bool isCharging = false;

// WiFi Client WiFiClient client;

void setup() { Serial.begin(115200);

Wire.begin(D2, D1); // I2C: SDA = D2, SCL = D1

// Initialize LCD lcd.init(); lcd.backlight(); lcd.clear(); lcd.setCursor(0, 0); lcd.print("Starting..."); delay(2000); lcd.clear();

// Setup pins pinMode(BUZZER, OUTPUT);

pinMode(GREEN_LED, OUTPUT);

// Connect WiFi WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi"); lcd.setCursor(0, 0);

lcd.print("WiFi Connecting...");
```

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

```
while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println("\nWiFi Connected!"); lcd.clear();
lcd.setCursor(0, 0); lcd.print("WiFi Connected!"); delay(2000);
lcd.clear();
// Start ThingSpeak
ThingSpeak.begin(client);
}
void loop() {
// Read analog value and convert to actual voltage
int analogValue = analogRead(VOLTAGE_SENSOR);
float sensorVoltage = analogValue * (3.3 / 1023.0); // Voltage at A0
voltage = sensorVoltage * (25.0 / 5.3); // Scale to actual input voltage (0-25V)
// Debugging Serial.print("Analog Value: "); Serial.print(analogValue); Serial.print(" | Sensor Voltage: "); Serial.print(sensorVoltage, 2);
Serial.print(" V | Actual Voltage: "); Serial.println(voltage, 2);
// Simulate SOH value
SOH = random(923, 986) / 10.0;
lcd.clear(); delay(100);
lcd.setCursor(0, 0); lcd.print("MPPT S/M"); lcd.setCursor(9, 0); lcd.print("SOH:" + String(SOH, 1));
if (voltage > 1.0) { lcd.setCursor(0, 1);
lcd.print("SOC:" + String(SOC, 1) + "% V:" + String(voltage, 1));
// 🔔 Buzzer beeps
for (int i = 0; i < 5; i++) { digitalWrite(BUZZER, HIGH); delay(500); digitalWrite(BUZZER, LOW); delay(500);
}
digitalWrite(GREEN_LED, LOW); isCharging = false;
} else {
lcd.setCursor(0, 1);
lcd.print("SOC:" + String(SOC, 1) + "% V:" + String(voltage, 1)); isCharging = true;
// ⚡ Blink LED to indicate charging for (int i = 0; i < 5; i++) {
digitalWrite(GREEN_LED, HIGH); delay(500); digitalWrite(GREEN_LED, LOW); delay(500);
}
}
// Simulate SOC charging if (isCharging) {
chargeCycles++;
if (chargeCycles % 2 == 0) { SOC += 0.5;
}
if (SOC > 100.0) { SOC = 100.0;
}
}
// Send data to ThingSpeak sendDataToThingSpeak(voltage, SOC, SOH, isCharging);
delay(5000);
}
void sendDataToThingSpeak(float voltage, float SOC, float SOH, bool isCharging) { String status = isCharging ? "Charging" : "Full";
ThingSpeak.setField(1, voltage); ThingSpeak.setField(2, SOC); ThingSpeak.setField(3, SOH); ThingSpeak.setField(4, status);
int response = ThingSpeak.writeFields(channelID, apiKey); if (response == 200) {
Serial.println("✅ Data sent to ThingSpeak!");
} else {
Serial.println("❌ Error sending data to ThingSpeak.");
}
}
```

B. HARDWARE

Solar panel output connects to the charge controller's input. Charge controller output connects to the battery terminals. Voltage sensor (e.g., a voltage divider with resistors or a module) measures battery voltage and connects to the ESP8266's analog pin (A0, max 3.3V input, requiring scaling for higher voltages). LCD connects to the ESP8266 via I2C or GPIO pins for local display. ESP8266 is powered by the battery or a separate regulator. Arduino IDE programs the ESP8266 with libraries like ESP8266WiFi (for Wi-Fi connectivity), ThingSpeak (for cloud upload), and LiquidCrystal_I2C (for LCD control). Code reads the analog voltage, calculates SoC and SoH, updates the LCD, and sends data to ThingSpeak at regular intervals. The hardware setup is represented in fig 4.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

Fig4. Hardware Setup

This project aims to design and implement a solar power system integrated with a battery management system (BMS) to monitor and manage a battery's performance remotely. The system uses a solar panel, a solar charge controller, an ESP8266 Wi-Fi module (programmed via Arduino), a voltage sensor, and an LCD display to track key battery parameters state of charge (SoC), state of health (SoH), and state of voltage (SoV) and upload the data to ThingSpeak, a cloud-based IoT platform, for real-time monitoring and analysis is represented in Fig5.

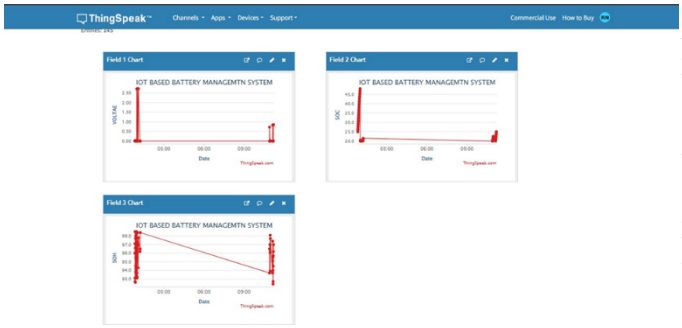

Fig5. State of Voltage, State of Charge and State of Health of Battery

VI. CONCLUSION

This project successfully integrates solar power with a BMS, providing both local (LCD) and remote (ThingSpeak) monitoring of battery parameters. It enables users to track SoC, SoV, and SoH in real time, ensuring efficient battery usage and maintenance. The system is scalable additional sensors (e.g., current, temperature) can improve accuracy and serves as a practical IoT solution for solar-powered applications. The solar power monitoring system with a battery management system (BMS) effectively demonstrates the integration of renewable energy and IoT technology for real-time battery performance tracking.

The system leverages a solar panel and charge controller to efficiently charge the battery, while the ESP8266 Wi-Fi module, programmed via Arduino, collects data from a voltage sensor to compute critical parameters: state of charge (SoC), state of health (SoH), and state of voltage (SoV). The local LCD display provides immediate feedback on these parameters, ensuring accessibility without internet dependency, while the seamless upload of data to ThingSpeak enables remote monitoring through a cloud-based platform. The incorporation of MATLAB with ThingSpeak enhances data analysis, allowing for advanced visualization and processing of battery trends over time. This project successfully achieves its objective of delivering a low-cost, reliable, and scalable solution for monitoring battery performance in a solar-powered setup. It validates the potential of combining hardware (ESP8266, sensors) with software (Arduino, MATLAB, ThingSpeak) to create an efficient BMS, offering insights into battery status for both immediate and long-term applications. The system's performance demonstrated through stable voltage regulation, accurate SoC estimation, and consistent data logging underscores its practical utility in small-scale renewable energy systems.

REFERENCES

[1] Mohd Helmy Abd Wahab et al (2018), "IoT-Based Battery Monitoring System for Electric Vehicle", International Journal of Engineering & Technology Vol 7, pp.505-510.

[2] M Ramesh Kumar et al (2018), "Battery Monitoring System using IoT", International Journal of Scientific Development and Research Vol 3, pp.126-128.

[3] N Harish et al (2018), "IOT Based Battery Management System", International Journal of Applied Engineering Research Vol 13, pp.5711- 5714.

[4] Y Mastanamma et al (2017), "Electric Vehicle Mathematical Modelling and Simulation using MATLAB Simulink", IOSR Journal of Electrical and Electronics Engineering Vol 12, pp.54-66.

[5] Vaibhav Verma et al (2013), "LabVIEW-based Battery Monitoring System with Effects of Temperature on Lead-Acid Battery", International Journal of Enhanced Research in Science Technology & Engineering Vol 6, pp.105- 108.

[6] Olivier Tremblay, Louis-A Dessaint (2009), "Experimental Validation of a Battery Dynamic Model for EV Applications", World Electric Vehicle Journal, Vol 3, pp. 2032-6653.

[7] D. Bell, "A battery management system," Master's thesis, School Eng., Univ. Queensland, St. Lucia, Australia, 2000.

[8] D. Fisher et al (1996), "Battery management: Increase in the reliability of UPS," ETZ, Vol117, pp. 18–22.

[9] K. Shimitzu, N. Shirai, and M. Nihei (1996), "On-board battery management system with SOC indicator," Proc. Int. Electric Vehicle Symp., Vol 2, pp. 99– 104.

[10] J. Alzieu, P. Gangol and H. Smimite (1995), "Development of an on-board charge and discharge management system for electric-vehicle batteries," J. Power Sources, Vol 53, pp. 327–333.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

[11] Shepherd, C. M (July 1965), "Design of Primary and Secondary Cells - An equation describing battery discharge", Journal of Electrochemical Society, Vol 112, pp.657-664.