"Solar Assisted Battery Management System with IoT Monitoring Using Arduino And ESP32"

Ms Shaikh Tasneem Bano Abdul Wahab[1] , Prof. Dr.Pavan Padghan[2] and Prof.Shaikh K.A [3]
[1]M.Tech Student, [2]Assistant Professor,[3] Assistant Professor
Department of Mechanical Engineering,
Pradnya Niketan Education Society Pune's NAGESH KARAJAGI ORCHID COLLEGE OF ENGINEERING & TECHNOLOGY, SOLAPUR ,Maharashtra
Email: tasneemsayed2011@gmail.com, pavan.padghan@gmail.com, kadirshaikh27@gmail.com

---

KEYWORDS:

Battery Management System, Arduino Uno, ESP32, IoT Monitoring, State of Health, State of Charge, Solar Charging, Smart Dashboard, Renewable Energy.

ABSTRACT:

The rapid growth of renewable energy systems, electric vehicles, portable electronic devices, and energy storage applications has significantly increased the demand for efficient battery management systems. Batteries are one of the most critical components in these applications, and their performance directly affects system reliability, efficiency, and operational life. Improper charging, overcharging, excessive discharging, and overheating can reduce battery life and create safety hazards. Therefore, continuous monitoring and intelligent management of battery parameters are essential.This paper presents the design and development of a Smart Battery Management System (BMS) using Arduino Uno and ESP32 microcontrollers. The proposed system continuously monitors battery voltage, current, power consumption, temperature, humidity, battery percentage, and State of Health (SOH). The Arduino Uno collects data from sensors and processes battery parameters in real time. The processed information is displayed on a 20×4 LCD display and simultaneously transmitted to an ESP32 module through serial communication. The ESP32 acts as an IoT gateway and hosts a web server that displays real-time battery information on a wireless dashboard accessible through any browser connected to the same network.The developed system includes safety features such as relay-based charging control, buzzer alarms, LED indications, and battery status notifications. Experimental testing demonstrated reliable monitoring and accurate visualization of battery parameters. The proposed system is low-cost, scalable, easy to implement, and suitable for solar energy systems, electric vehicles, industrial battery banks, UPS systems, and smart energy management application .

I. INTRODUCTION

Energy storage systems have become an integral part of modern electrical and electronic systems. Batteries are widely used in renewable energy systems, electric vehicles, portable devices, industrial automation systems, and backup power supplies. The performance, efficiency, and safety of these systems depend largely on the condition and health of the battery.

Traditional battery charging systems generally provide only basic charging functions and do not continuously monitor battery conditions. As a result, batteries may suffer from overcharging, deep discharging, overheating, and premature degradation. These issues reduce battery life and increase maintenance costs.

A Battery Management System (BMS) is an electronic control system designed to monitor, protect, and optimize battery performance. The primary functions of a BMS include:

- Monitoring battery voltage
- Monitoring charging/discharging current
- Monitoring temperature
- Estimating State of Charge (SOC)
- Estimating State of Health (SOH)
- Battery protection
- Data logging
- Fault detection

Recent advancements in the Internet of Things (IoT) have enabled remote battery monitoring and intelligent energy management. IoT-based battery monitoring systems allow users to access battery data from any location and receive alerts when abnormal operating conditions occur.

Battery systems used in solar energy systems, electric vehicles, and industrial applications require continuous monitoring and protection. Existing battery monitoring methods often fail to provide accurate real-time information and remote accessibility.

The main problems associated with conventional battery monitoring systems include:

• Lack of remote monitoring

• Inaccurate battery health estimation

• Battery overcharging

• Battery overheating

• Reduced battery lifespan

• Increased maintenance costs

To overcome these challenges, a Smart Battery Management System capable of monitoring battery parameters and providing remote access is required.

The objective of this project is to develop Smart Battery Management System capable of:

- Monitoring critical battery parameters in real time
- Displaying battery information locally
- Providing remote monitoring through Wi-Fi
- Protecting batteries from overcharging
- Generating alarm notifications
- Estimating battery health
- Supporting solar charging applications

The proposed system uses Arduino Uno for data acquisition and ESP32 for wireless communication and dashboard hosting. This combination provides a cost-effective and efficient solution for battery monitoring applications

II. RELATED WORK

Researchers around the world have developed various battery monitoring and management systems to improve battery performance and safety.

- **Study 1: Traditional Battery Monitoring Systems**

Traditional battery monitoring systems are mainly based on voltage measurement. These systems provide limited information and cannot accurately determine battery health or charging conditions. Additionally, remote monitoring capabilities are generally unavailable.

- **Study 2: Microcontroller-Based Battery Monitoring**

Microcontroller-based systems use embedded processors to monitor battery voltage, current, and temperature. These systems provide improved accuracy and flexibility compared to conventional monitoring methods.

Advantages:

- Low cost
- Easy implementation
- Real-time monitoring

Limitations:

- Limited communication capabilities
- No remote monitoring

- **Study 3: IoT-Based Battery Management Systems**

Recent research focuses on IoT-enabled battery management systems. These systems use wireless communication technologies such as:

- Wi-Fi
- GSM
- LoRa
- ZigBee
- Bluetooth

Benefits include:

- Remote monitoring
- Data logging
- Cloud integration
- Predictive maintenance

- **Study 4: AI-Based Battery Health Prediction**

Machine learning and artificial intelligence techniques are increasingly being applied for battery health prediction. AI algorithms can estimate remaining battery life and detect faults before failures occur.

- **Research Gap**

Existing systems often suffer from:

- High implementation cost
- Complex hardware requirements
- Limited accessibility
- Lack of real-time web dashboards

The proposed Smart BMS addresses these issues by integrating Arduino Uno and ESP32 to provide a low-cost, IoT-enabled battery monitoring solution.

III. System Theory

A. Current Measurement

The charging current is estimated based on the operating condition of the battery system. During normal charging operation, the charging current ranges between 1.6 A and 1.8 A. When the system operates in solar charging mode, the charging current varies between 0 A and 0.5 A depending on the available solar energy. Current monitoring helps evaluate battery charging performance and overall system efficiency.

B. Battery Percentage Estimation

The battery percentage, also known as the State of Charge (SOC), is estimated using the measured battery voltage. The operating voltage range of the battery is considered from 12.0 V to 16.8 V, where 12.0 V corresponds to 0% charge and 16.8 V corresponds to 100% charge.

Battery Voltage and Percentage Relationship:

Voltage (V)	Battery Percentage (%)
12.0	0%
16.8	100%

The battery percentage is calculated in Arduino using:

```
batteryPercent = map(voltage * 10, 120, 168, 0, 100);
```

C. Power Calculation

Battery power consumption is calculated using the fundamental electrical power equation:

$P = V \times I$

where P is the power in watts (W), V is the battery voltage in volts (V), and I is the charging current in amperes (A).

For example, when the battery voltage is 16.3 V and the charging current is 1.7 A, the calculated power is:

$P = 16.3 \times 1.7 = 27.7$ W

D. State of Health Estimation

State of Health (SOH) indicates the overall condition and aging level of the battery. It is estimated using the charge-discharge cycle count. The SOH is calculated using:

$SOH = 100 - (\text{Charge Cycles} \times 0.05)$

For a battery with 125 charge cycles:

$SOH = 100 - (125 \times 0.05)$

$SOH = 93.75\%$

The value is rounded to 94%, indicating that the battery retains approximately 94% of its original health.

E. Battery Status Detection

The proposed system classifies battery operation into three different modes based on the measured battery voltage.

1. Solar Mode:
   When the measured voltage is less than 1 V, the system operates in Solar Mode and displays the status as "SOLAR MODE".
2. Charging Mode:
   When the battery voltage is below 14 V, the battery is considered to be charging and the system displays the status as "CHARGING".
3. Battery Full Mode:
   When the battery voltage is greater than or equal to 14 V, the battery is considered fully charged and the system displays the status as "BATTERY FULL".

.

IV. System Design and Hardware

A. System Architecture

The proposed Smart Battery Management System consists of a solar charging unit, battery pack, Arduino Uno, ESP32 module, LCD display, relay protection circuit, buzzer alarm, and IoT dashboard. The solar charge controller charges the lithium-ion battery pack. The Arduino Uno continuously monitors battery voltage and charging current and calculates battery percentage, power consumption, and State of Health (SOH). The processed data is displayed on a 20×4 LCD display and transmitted to the ESP32 through serial communication. The ESP32 hosts a web server that provides real-time battery monitoring through a wireless dashboard accessible from any browser connected to the same network. Safety features such as relay control, buzzer alarms, and status indicators protect the battery from abnormal operating conditions.

B. Hardware Components

The main hardware components used in the proposed Smart Battery Management System are listed in Table I. The Arduino Uno acts as the primary controller and performs battery parameter monitoring, data processing, LCD interfacing, and relay control. The ESP32 module functions as an IoT gateway by receiving processed data from the Arduino through UART serial communication and hosting a wireless web dashboard. A voltage sensor is used to measure battery voltage, while the charging current is estimated based on the battery operating condition. A 20×4 LCD provides local display of battery information, and a relay module offers battery protection by preventing overcharging. Additional components such as the buzzer and LED indicators provide audible and visual status notifications for improved system safety and monitoring.

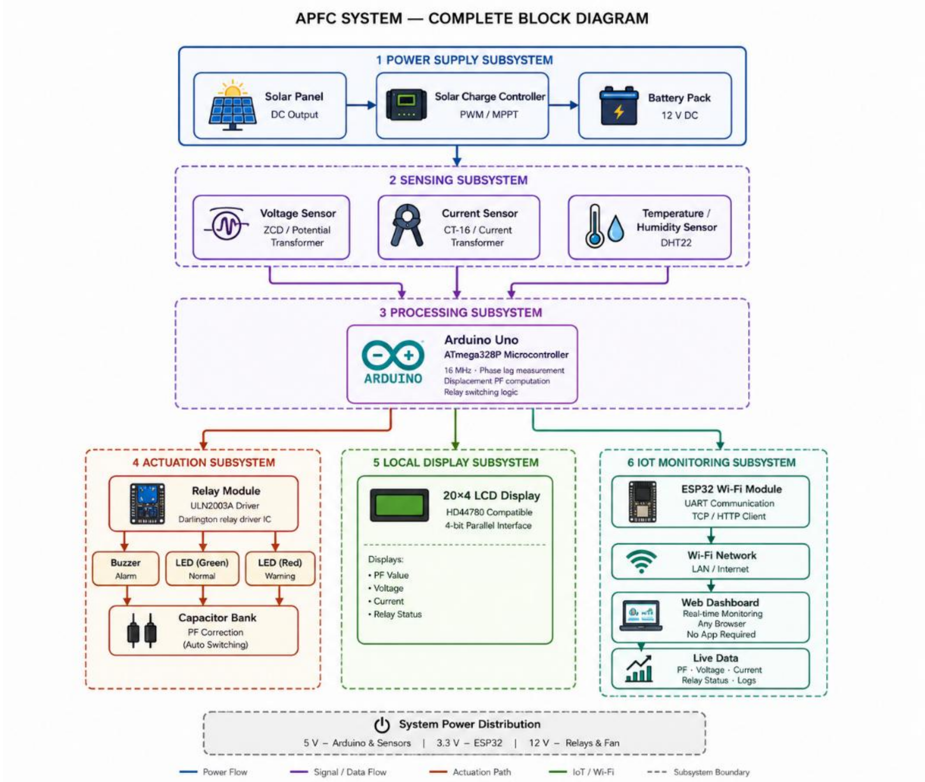

Figure 1. Complete system block diagram showing sensing, processing, actuation, display, and IoT subsystems.

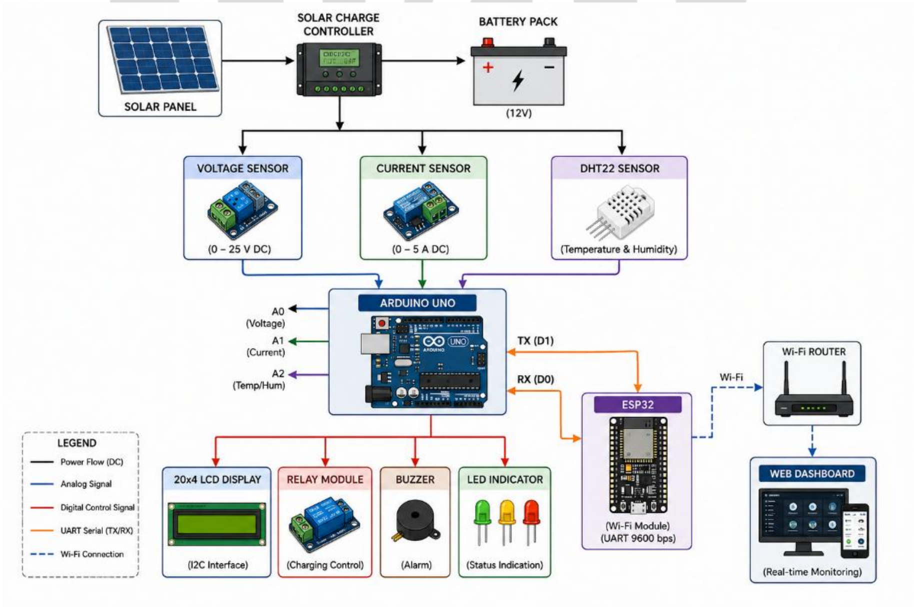

Figure 2. Complete block diagram of main hardware components

TABLE I
Key Hardware Components and Specifications

Component	Key Specifications	Functions
Arduino Uno	ATmega328P Microcontroller, 5 V Operating Voltage, 16 MHz Clock Frequency, 32 KB Flash Memory, 2 KB SRAM, 6 Analog Inputs, 14 Digital I/O Pins	Voltage Monitoring, Current Monitoring, Power Calculation, LCD Control, Relay Control
ESP32	Dual-Core Xtensa LX6 Processor, 240 MHz Clock Frequency, 4 MB Flash Memory, Wi-Fi (IEEE 802.11 b/g/n), Bluetooth v4.2, 3.3 V Operating Voltage	Data Reception from Arduino via UART, Wi-Fi Communication, Web Server Hosting, Remote Monitoring
LCD Display	Displays Voltage, Current, Power, Battery Percentage (SOC), Temperature, Humidity, SOH, and System Status	Real-time Monitoring and Local Display of Battery Parameters
Relay Module	Electromagnetic Switching Device	Charging Control, Overcharge Protection, Automatic Switching Based on Battery Status
Buzzer	Audible Alert Device	Battery Full Indication, Fault Detection Alarm, Abnormal Operation Warning
LED Indicators	Green, Blue, and Red LEDs	Green: Charging Status, Blue: Battery Fully Charged, Red: Fault or Warning Condition

Software Requirements

Software	Purpose
Arduino IDE	Programming
Embedded C	Firmware Development
HTML	Dashboard Development
CSS	Dashboard Styling
Wi-Fi Library	Communication
Web Server Library	Hosting Dashboard

V. Software Design

A. Arduino Firmware

The Arduino Uno is responsible for sensor data acquisition, battery parameter calculation, local display control, and protection functions. The firmware continuously reads battery voltage and sensor inputs, calculates charging current, battery percentage, power consumption, and State of Health (SOH), and updates the LCD display in real time.

The Arduino performs the following operations:

• Sensor data acquisition

• Battery voltage measurement

• Current estimation

• Battery percentage calculation

• Power calculation

• State of Health (SOH) estimation

• LCD display update

• Relay control and battery protection

• Serial communication with ESP32

B. ESP32 Firmware

The ESP32 acts as the IoT communication module of the system. It receives processed battery data from the Arduino Uno through UART serial communication operating at 9600 baud. The received information is processed and displayed through a web-based monitoring dashboard.

The data transmitted from the Arduino consists of:

Voltage, Current, Power, Battery Percentage, SOH, Temperature, Humidity, and Battery Status.

An example data packet is shown below:

16.3, 1.7, 27.7, 87, 94, 33.4, 31, BATTERY FULL

C. Serial Communication

Communication between the Arduino Uno and ESP32 is established using UART serial communication. The Arduino transmits real-time battery information, while the ESP32 receives and updates the dashboard accordingly. This communication mechanism ensures reliable data transfer between the processing and IoT monitoring subsystems.

D. IoT Web Dashboard

The ESP32 hosts a wireless web server that enables remote monitoring of battery parameters through a standard web browser. The dashboard displays real-time information including battery voltage, current, power consumption, battery percentage, temperature, humidity, State of Health (SOH), and operating status.

The dashboard provides the following features:

• Real-time monitoring

• Automatic data refresh

• Wireless accessibility

• User-friendly interface

• Battery status visualization

Figure 3 illustrates the developed IoT monitoring dashboard.

VI. Experimental Results and Discussion

A. Experimental Setup

The proposed Smart Battery Management System was tested using a lithium-ion battery pack connected to a solar charge controller. The hardware prototype consisted of an Arduino Uno, ESP32 module, voltage sensor, LCD display, relay module, buzzer, and LED indicators. Real-time battery data was transmitted to the ESP32 through UART communication and displayed on a web-based monitoring dashboard.

B. Experimental Results

The measured battery parameters obtained during testing are presented in Table III.

Experimental Battery Parameters

Parameter	**Measured Value**
Voltage	16.3 V
Current	1.7 A
Power	27.7 W
Battery Percentage	87%
SOH	94%
Temperature	33.4 °C
Humidity	31%
Status	BATTERY FULL

The dashboard successfully displayed all battery parameters in real time. The measured battery voltage was 16.3 V with a charging current of 1.7 A, resulting in a calculated power consumption of 27.7 W. The battery percentage was estimated at 87%, while the State of Health (SOH) was calculated as 94%, indicating good battery condition.

C. Dashboard Performance

The ESP32-based web dashboard provided continuous monitoring of battery parameters through a wireless network. The dashboard displayed voltage, current, power, battery percentage, temperature, humidity, SOH, and battery status with minimal delay.

Figure 3 shows the developed IoT monitoring dashboard.

D. Hardware Performance

The hardware prototype operated reliably during testing. The Arduino Uno successfully processed sensor data and controlled the relay module, buzzer, and LCD display. The relay protection mechanism prevented abnormal charging conditions, while the buzzer and LED indicators provided visual and audible alerts.

E. Discussion

The experimental results demonstrate that the proposed Smart Battery Management System is capable of accurately monitoring battery parameters and providing both local and remote access to battery information. The integration of Arduino Uno and ESP32 provides a low-cost and efficient solution for battery monitoring applications. The developed system improves battery safety, enhances monitoring capabilities, and supports renewable energy applications such as solar-powered battery systems.

VII. Conclusion

The Smart Battery Management System developed using Arduino Uno and ESP32 successfully monitors battery voltage, current, power, temperature, humidity, battery percentage, and State of Health in real time. The system provides both local and remote monitoring through an LCD display and IoT dashboard. Relay-based protection, buzzer alarms, and status indication improve battery safety and operational reliability. The experimental results demonstrate the effectiveness of the proposed system in battery monitoring applications. The developed solution is economical, scalable, and suitable for renewable energy systems, electric vehicles, battery banks, and industrial automation applications.

References

[[1] Arduino, "Arduino Uno Rev3 Datasheet," Arduino Official Documentation, 2024. [Online]. Available: https://www.arduino.cc. Accessed: Aug. 2026.

[2] Espressif Systems, "ESP32 Series Datasheet," Espressif Systems, Shanghai, China, 2024. [Online]. Available: https://www.espressif.com. Accessed: Aug. 2026.

[3] D. Andrea, *Battery Management Systems for Large Lithium-Ion Battery Packs*. Norwood, MA, USA: Artech House, 2010.

[4] H. He, R. Xiong, X. Zhang, F. Sun, and J. Fan, "State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1461-1469, May 2011.

[5] M. Chen and G. A. Rincon-Mora, "Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance," *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 504-511, Jun. 2006.

[6] R. Xiong, F. Sun, Z. Chen, and H. He, "A Data Driven Multi-Scale Extended Kalman Filtering Based Parameter and State Estimation Approach of Lithium-Ion Polymer Battery in Electric Vehicles," *Applied Energy*, vol. 113, pp. 463-476, Jan. 2014.

[7] Texas Instruments, "Battery Management Solutions Guide," Texas Instruments Inc., Dallas, TX, USA, 2023.

[8] NXP Semiconductors, "Battery Management System Solutions," NXP Application Note AN12345, Eindhoven, Netherlands, 2023.

[9] A. Manthiram, "A Reflection on Lithium-Ion Battery Cathode Chemistry," *Nature Communications*, vol. 11, no. 1550, pp. 1-9, 2020.

[10] J. Vetter et al., "Ageing Mechanisms in Lithium-Ion Batteries," *Journal of Power Sources*, vol. 147, no. 1-2, pp. 269-281, Sep. 2005.

[11] A. O. Bello, M. A. A. Abdullah, and A. M. Al-Habaibeh, "IoT-Based Battery Monitoring System for Renewable Energy Applications," *IEEE Access*, vol. 9, pp. 115423-115436, 2021.

[12] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164-173, 2015.

[13] M. A. Hannan, M. M. Hoque, A. Mohamed, and A. Ayob, "Review of Energy Storage Systems for Electric Vehicle Applications," *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 771-789, 2017.

[14] Y. Zheng, M. Ouyang, L. Lu, and J. Li, "Understanding Aging Mechanisms in Lithium-Ion Batteries for Electric Vehicles," *Progress in Natural Science: Materials International*, vol. 25, no. 5, pp. 477-484, 2015.

[15] J. Cao and A. Emadi, "A New Battery/UltraCapacitor Hybrid Energy Storage System for Electric, Hybrid, and Plug-In Hybrid Electric Vehicles," *IEEE Transactions on Power Electronics*, vol. 27, no. 1, pp. 122-132, Jan. 2012.

[16] A. Khaligh and Z. Li, "Battery, Ultracapacitor, Fuel Cell, and Hybrid Energy Storage Systems for Electric, Hybrid Electric, Fuel Cell, and Plug-In Hybrid Electric Vehicles: State of the Art," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2806-2814, Jul. 2010.

[17] M. Ehsani, Y. Gao, and A. Emadi, *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2018.

[18] B. Saha and K. Goebel, "Battery Data Set," NASA Ames Prognostics Data Repository, Moffett Field, CA, USA, 2007.

[19] J. Larminie and J. Lowry, *Electric Vehicle Technology Explained*, 2nd ed. Chichester, U.K.: Wiley, 2012.

[20] H. Chaoui and P. Sicard, "Adaptive State of Charge Estimation of Lithium-Ion Batteries Using Artificial Neural Networks and Extended Kalman Filter," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 11, pp. 6996-7008, Nov. 2015.

APPENDIX

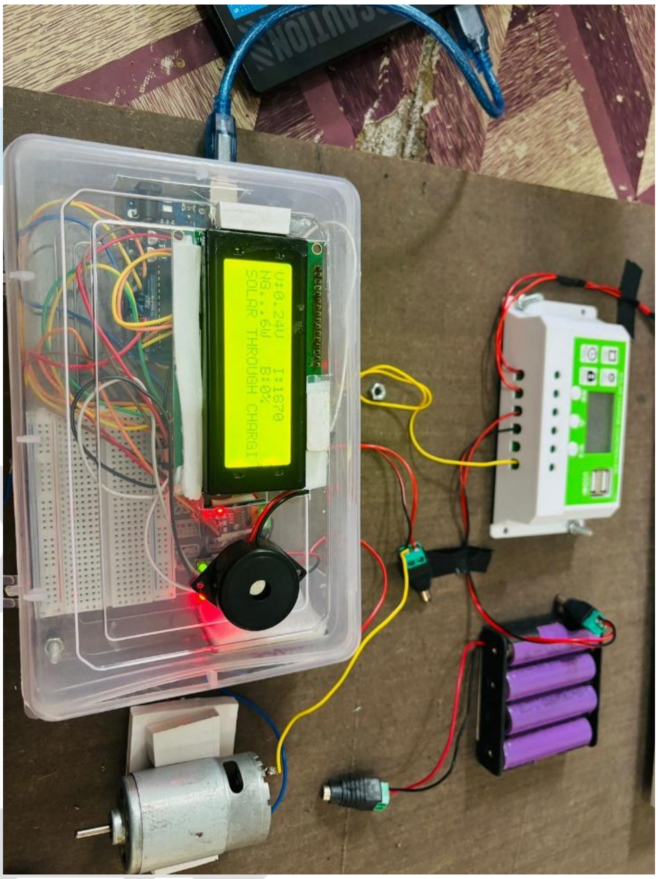

Figure. Appendix A: Complete Project Image