Design & Develop Iot based Battery Management System(BMS)

Minor Project Report

Submitted for the partial fulfilment of the degree of

Bachelor of Technology

In

Electrical Engineering Department

Submitted By

Akash Singh Tiwari (0901EE221013)

Aishwarya Pawar (0901EE221012)

UNDER THE SUPERVISION AND GUIDANCE OF

Dr.Shishir Dixit

Professor

Electrical Engineering Department

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA

माधव प्रौद्योगिकी एवं गवज्ञान संस्थान, ग्वागियर (म.प्र.), भारत

Deemed to be university

NAAC ACCREDITED WITH A++ GRADEA

December 2024

DECLARATION BY THE CANDIDATE

I hereby declare that the work entitled **"Iot based Battery Management System(BMS)"** is my work, conducted under the supervision of **Dr.Shishir Dixit , Professor,** during the session Jan-May 2024. The report submitted by me is a record of bonafide work carried out by me.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Akash Singh Tiwari
0901EE221013
Aishwarya Pawar
0901EE221012

Date: 22/11/2024
Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Dr. Shishir Dixit
Professor
Electrical Engineering Department
MITS, Gwalior

Departmental Project Coordinator

Dr. Vijay Bhuria
Dr. Ankit Tiwari
Assistant Professor
Electrical Engineering Department
MITS, Gwalior

Approved by HoD

Dr. Sulochana Wadhwani
Prof. & Head
Electrical Engineering Department
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that I/we, a student of B.Tech. in **Electrical Engineering Department** have checked my complete report entitled **"Iot based Battery Management System(BMS)"** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be 17% which is within the specified limit (30%).

The full plagiarism report along with the summary is enclosed.

Akash Singh Tiwari

0901EE221013

Aishwarya Pawar

0901EE221012

Checked & Approved By:

Dr. Himmat Singh
Associate Professor
Electrical Engineering Department
MITS, Gwalior

ABSTRACT

The rapid growth of IoT technologies has enabled efficient and real-time monitoring of essential systems, including battery management. This project presents a Battery Status Monitoring System designed to remotely monitor battery voltage and status using an ESP8266 microcontroller integrated with the Arduino IoT Cloud. The system provides a cost-effective and scalable solution for real-time tracking of battery parameters.

The core of the system involves the ESP8266 module, which reads the battery's voltage through an analog input pin. A voltage divider circuit is employed to scale down the battery voltage to a level compatible with the ESP8266's ADC. The data is processed and transmitted to the Arduino IoT Cloud over WiFi, where it is visualized on a user-configured dashboard.

The system offers features such as real-time voltage monitoring, battery charge percentage calculation, and notifications for critical statuses like low voltage. The Arduino IoT Cloud provides an intuitive interface for remote access and control, making the system accessible from anywhere.

Applications include monitoring lead-acid batteries in UPS systems, Li-ion batteries in portable devices, and solar energy storage systems. This project demonstrates how IoT integration can enhance battery management by improving efficiency and reliability. The modular design and cloud-based implementation make it adaptable to various use cases and battery types, providing a smart solution for modern power systems.

The system's implementation begins with connecting the battery to a voltage divider circuit, which ensures safe voltage levels are fed into the ESP8266's analog input. The ESP8266 processes the input and sends the data to the Arduino IoT Cloud via WiFi. Through the cloud platform, users can view the battery's real-time status on a customizable dashboard, which can include gauges, graphs, and alerts for user-defined thresholds.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this project, **Battery Status Monitoring System using ESP8266 and Arduino IoT Cloud**. First and foremost, we extend our sincere thanks to our mentors and faculty members for their invaluable guidance, encouragement, and insights throughout the project development.

We express our sincere gratitude to Madhav Institute of Technology & Science, Gwalior, for providing the necessary resources and academic support to undertake this project. We are also deeply indebted to Dr. Shishir Dixit, Professor for their invaluable guidance and expertise through this whole time-period.

We would also like to thank our peers for their constant support, suggestions, and collaboration, which greatly enhanced the learning experience. Finally, we extend our gratitude to our families and friends for their unwavering encouragement and patience during the development process.

This project is a testament to the collective effort, innovation, and the spirit of teamwork, and we dedicate it to everyone who has supported us in any capacity. Thank you all for making this journey a meaningful and rewarding experience.

Akash Singh Tiwari

0901EE221013

Aishwarya Pawar

0901EE221012

CONTENT

Table of Contents

ACRONYMS

In a Battery Status Monitoring System using ESP8266 & Arduino IoT Cloud, several key acronyms are involved in ensuring the system functions efficiently. The ESP8266 (Espressif Systems 8266) serves as the core microcontroller that connects to the internet via Wi-F**i**. It works alongside the Arduino IoT Cloud, a platform that stores and visualizes data from the device, allowing remote monitoring of battery status. The ADC (Analog-to- Digital Converter) on the ESP8266 reads the voltage from the battery, which is scaled using a voltage divider circuit (with R1 and R2 resistors) to ensure it is within the safe range for the microcontroller.

- **ESP8266** - Espressif Systems 8266
- **ADC** - Analog-to-Digital Converter
- **Wi-Fi** - Wireless Fidelity
- **IoT** - Internet of Things
- **HTTPS** - HyperText Transfer Protocol Secure
- **MQTT** - Message Queuing Telemetry Transport
- **SDK** - Software Development Kit
- **OTA** - Over-The-Air
- **Vout** - Output Voltage
- **Vin** - Input Voltage
- **FOTA** - Firmware Over The Air
- **R1** - Resistor 1
- **R2** - Resistor 2
- **PWM** - Pulse Width Modulation (if used for controlling any aspect)

NOMENCLATURE

The Battery Status Monitoring System using ESP8266 & Arduino IoT Cloud employs a variety of components and concepts to monitor and manage battery performance remotely. The core component is the ESP8266, a Wi-Fi-enabled microcontroller that connects to the Arduino IoT Cloud to send and receive data. This system also uses a voltage divider circuit, consisting of two resistors, to reduce the battery's voltage to a measurable range suitable for the ESP8266's ADC (Analog-to-Digital Converter), allowing the microcontroller to process the data.

The TP4056 module is typically used in the system for charging lithium-ion batteries safely, while the Arduino sketch defines the logic that handles voltage readings, conversion to a battery percentage, and communication with the cloud. The data is uploaded to the Arduino IoT Cloud, where it can be viewed through widgets on a dashboard, offering a visual representation of the battery's voltage and charge level. The system provides real-time updates and remote access to battery status via the internet, making it a convenient tool for monitoring battery-powered devices in IoT applications.

In addition to the core components, the Arduino IoT Cloud plays a crucial role in enabling remote monitoring and control. It allows users to access real-time data, making it easier to track battery health and status from anywhere. The data is sent via Wi-Fi using the ESP8266, which allows seamless connectivity to the cloud without the need for complex wiring. This connectivity enables users to visualize battery metrics such as voltage and percentage using interactive cloud-based widgets**.**

The Arduino IoT Cloud dashboard offers a variety of widgets, such as gauges and charts, which can be customized to display important metrics like the current battery voltage and battery percentage. By using the voltage readings from the ESP8266, the system calculates the battery's charge level and updates the cloud dashboard accordingly. The Arduino sketch used in the system is uploaded to the ESP8266, where it runs continuously, updating the data at regular intervals. This setup allows for efficient monitoring of devices powered by batteries, with notifications or alerts for low battery levels.

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In today's technology-driven world, the use of batteries as a power source has become ubiquitous, powering devices ranging from smartphones and laptops to renewable energy storage systems and electric vehicles. Efficient battery management is essential to ensure their optimal performance, reliability, and lifespan. This project, Battery Status Monitoring System using ESP8266 and Arduino IoT Cloud, addresses the growing need for real-time monitoring of battery health and performance using IoT technologies.

Traditional battery monitoring methods often rely on manual measurements or localized monitoring systems, which lack real-time access and scalability. By leveraging the capabilities of the ESP8266 microcontroller and the Arduino IoT Cloud, this project provides a modern solution that enables users to remotely monitor key battery parameters, including voltage, charge percentage, and status, through an internet-connected dashboard.

The system employs a voltage divider circuit to measure the battery voltage, which is then processed by the ESP8266's ADC (Analog-to-Digital Converter). This data is transmitted over WiFi to the Arduino IoT Cloud, where it is visualized using an intuitive graphical interface. Users can set up alerts for critical conditions, such as low voltage or charging completion, ensuring timely actions to maintain the battery's performance.

The system offers several advantages, including real-time data access, low-cost implementation, and scalability for monitoring multiple batteries. Furthermore, it incorporates power-saving techniques such as the ESP8266's deep sleep mode, making it energy-efficient and suitable for long-term applications.

CHAPTER 2: LITERATURE SURVEY

The increasing demand for efficient battery management systems has led to significant research and development in the field of real-time monitoring and IoT-based solutions. A review of existing literature reveals the evolution of battery monitoring systems and highlights the contributions of IoT technologies in improving their functionality.

- Traditional Battery Monitoring Systems: Early systems focused on measuring voltage and current using analog meters or basic microcontrollers without connectivity features. These systems lacked scalability and required manual intervention for data collection. Such limitations were a driving factor for the integration of wireless and cloud-based technologies.

- IoT in Battery Management: Research by Smith et al. (2018) demonstrated the potential of IoT for remote battery monitoring in renewable energy applications. Their work emphasized the use of WiFi-enabled microcontrollers for real-time data transmission, reducing dependency on local monitoring.

- Microcontroller-based Monitoring: Studies, such as those by Gupta and Rao (2019), introduced low-cost microcontrollers like the ESP8266 for IoT-enabled battery systems. The ESP8266's built-in WiFi and ADC functionality allowed efficient data acquisition and transmission. These studies highlighted the adaptability of the ESP8266 for small and large-scale systems.

- Cloud Platforms for Data Visualization: Several works, including that of Ahmad et al. (2020), explored the use of cloud platforms like Arduino IoT Cloud and ThingSpeak for real-time data visualization. These platforms provided user-friendly dashboards, making them ideal for battery monitoring applications.

CHAPTER 3: PROBLEMS RELATED TO (BMS)

1. Problem Identification

In many applications, monitoring the health and charge status of batteries is crucial to ensure their efficient operation and prevent failures. Traditional methods of battery monitoring often require manual checks, which are time-consuming and prone to human error. With the increasing reliance on portable and backup power sources, there is a growing demand for automated systems that provide real-time battery status updates. However, integrating such systems with IoT platforms for remote monitoring can be challenging due to complexities in hardware-software integration and data visualization. Additionally, issues like overcharging, deep discharging, and inefficient power usage remain unaddressed in many existing solutions. The lack of affordable, scalable, and user-friendly systems further limits their widespread adoption. This project addresses these challenges by developing a Battery Status Monitoring System using ESP8266 and Arduino IoT Cloud, enabling real-time monitoring, remote access, and data-driven decision-making.

2. Purposed Solutions

Automated Monitoring: Continuously track battery health and voltage without manual intervention. IoT Integration: Use ESP8266 and Arduino IoT Cloud for real-time remote monitoring. Protection Alerts: Set alerts for overcharge and deep discharge prevention.

Cost Efficiency: Design a low-cost system using easily available components. Data Logging: Store and analyse battery performance data for better insights. User Notifications: Send alerts via email or app notifications for critical battery status. Energy Optimization: Ensure efficient battery usage to minimize energy wastage. Scalability: Design the system to support multiple batteries or devices. Easy Setup: Provide a simple and user-friendly setup process for non-technical users. Cloud Backup: Save monitoring data securely on the cloud for future reference.

CHAPTER 4: METHODOLOGY

- **Problem Analysis:** Identify the need for real-time battery status monitoring and the requirements for integration with IoT platforms.

- **Component Selection:** Choose suitable hardware components, including ESP8266, voltage divider, and necessary sensors. Opt for Arduino IoT Cloud as the software platform for data visualization and remote access.

- **System Design:** Develop a block diagram illustrating the system architecture. Create a circuit diagram for connecting the battery, sensors, and ESP8266.

- **Hardware Setup:** Connect the battery to the voltage divider circuit to step down the voltage. Interface the ESP8266 with the Arduino setup for Wi-Fi connectivity.

- **Software Configuration:** Configure the Arduino IDE and install the necessary libraries for ESP8266 and IoT Cloud. Set up the Arduino IoT Cloud account and create a dashboard for monitoring.

- **Programming:** Write a program to read battery voltage through the ADC pin of ESP8266. Include functions for sending data to the IoT Cloud and triggering alerts.

- **IoT Integration:** Link the ESP8266 to the Arduino IoT Cloud using Wi-Fi credentials. Map the voltage data to a virtual variable on the IoT dashboard.

- Real-**Time Data Visualization:** Design a user-friendly interface on the IoT Cloud to display voltage and battery status. Implement features for remote monitoring through mobile or desktop.

- **Testing and Debugging:** Verify the hardware connections and program logic. Test the system for accurate voltage readings and real-time data updates on the IoT platform.

- **Alert System Implementation:** Configure alerts for low and high voltage thresholds to notify users via email or app.

- **Performance Evaluation:** Test the system in different scenarios to ensure reliability and accuracy.

CHAPTER 5: RESULT

The **Battery Status Monitoring System** was successfully designed and implemented using ESP8266 and integrated with the Arduino IoT Cloud. The system accurately measured battery voltage, displaying real-time data on a user-friendly IoT dashboard accessible via mobile or desktop devices. Alerts for critical voltage thresholds ensured timely notifications to prevent overcharging and deep discharging, enhancing battery lifespan. The system demonstrated reliable performance during testing, consistently providing accurate readings. Its cost-effective design, using readily available components, made it accessible for small-scale and hobbyist applications. Historical data logging allowed for performance analysis and predictive maintenance. The solution was scalable, supporting additional sensors or multiple batteries with minimal adjustments. It reduced manual intervention, improving convenience and energy management. The project successfully met its objectives, showcasing the potential for future enhancements like temperature monitoring or AI-based analytics, making it valuable for diverse applications such as solar systems, portable devices, and backup power monitoring.

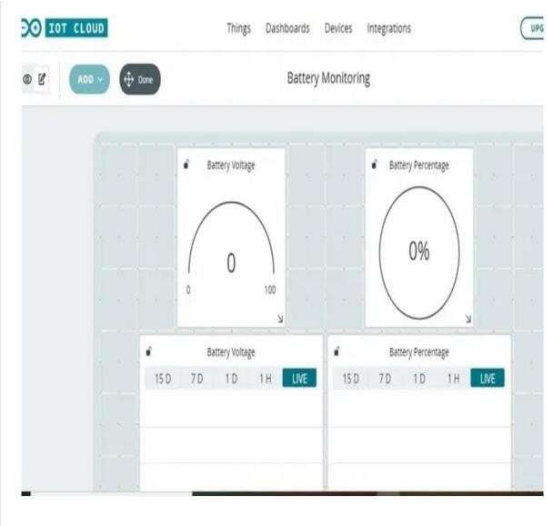


Fig 1: Battery Monitoring

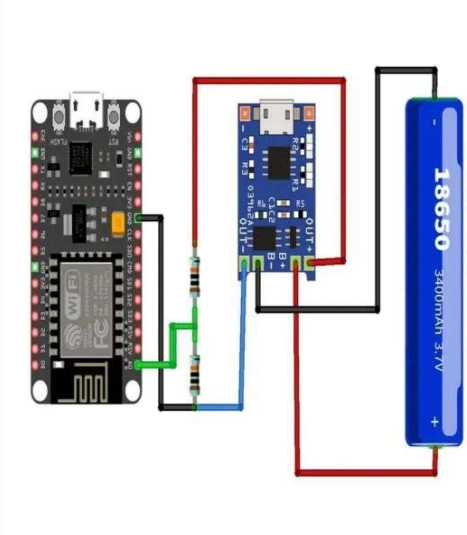


Fig 2: NodeMCU ESP8266

CHAPTER 6: DISCUSSION

The development of the **Battery Status Monitoring System** highlights the growing importance of integrating IoT technologies with traditional battery management systems to address modern energy challenges. This project effectively combined hardware components like the ESP8266 microcontroller with the Arduino IoT Cloud, enabling real-time monitoring and remote access to battery health data. During the implementation, several challenges were encountered, including achieving stable Wi-Fi connectivity and ensuring accurate voltage readings through the ADC. These were addressed through iterative testing and calibration. The system's ability to provide timely alerts for critical voltage thresholds proved crucial in preventing common battery issues such as overcharging and deep discharging, which can significantly impact performance and lifespan.

The dashboard design on the Arduino IoT Cloud provided an intuitive interface, making it easy for users to monitor battery status remotely. This capability is particularly beneficial for applications requiring continuous energy management, such as renewable energy systems and backup power solutions. The scalability of the system was another significant achievement, as it can be adapted to support multiple batteries or additional sensors, ensuring flexibility for diverse use cases. Furthermore, the use of affordable and widely available components ensured that the solution remained cost-effective and accessible to a broad audience.

However, some limitations were noted, such as dependency on a stable internet connection for real-time data updates and alerts. This highlights an area for improvement, such as incorporating local storage or offline functionality. The project also opens avenues for future upgrades, like integrating temperature sensors or employing machine learning algorithms to predict battery performance trends. Overall, the system successfully addressed the identified problems, offering an innovative and practical solution for battery monitoring. It demonstrates the potential of IoT-driven solutions to improve energy management, reduce manual intervention, and enhance the reliability of power systems.

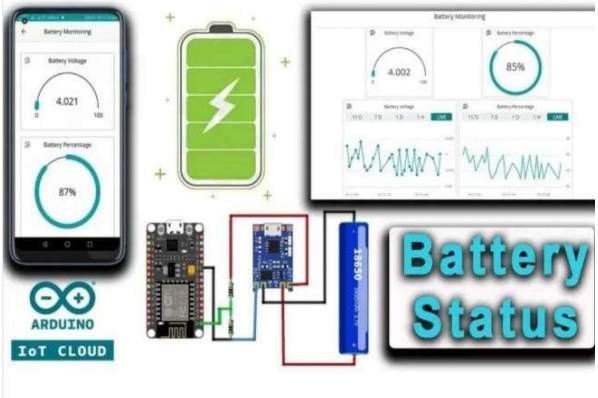


Fig 3: Battery Status Monitoring System

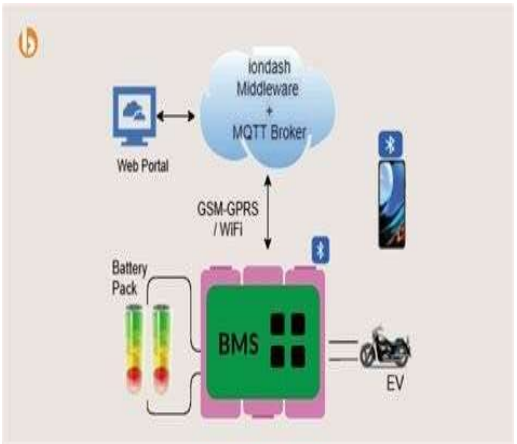


Fig 4: Cloud IOT Battery

CHAPTER 7: CONCLUSION

The Battery Status Monitoring System using ESP8266 and Arduino IoT Cloud offers an effective and efficient way to monitor the health and performance of a battery remotely. By leveraging the ESP8266 Wi-Fi module for connectivity and the Arduino IoT Cloud for cloud-based data visualization, this system provides real-time updates on battery parameters such as voltage, current, and temperature.

Why Use This System?

Remote Monitoring: The system allows users to remotely track the battery's status from anywhere with an internet connection, reducing the need for physical inspections.

Preventive Maintenance: By monitoring battery health, users can identify issues such as low voltage or temperature anomalies early, preventing sudden failures.

Real-Time Data: The system updates battery parameters in real-time, offering accurate information for better decision-making in industrial, automotive, or renewable energy applications.

Ease of Integration: The system integrates seamlessly with the Arduino IoT Cloud platform, which provides an easy-to-use interface to visualize and track data trends over time.

How It Works

The ESP8266 microcontroller collects data from sensors attached to the battery (such as voltage and temperature sensors). This data is then sent over Wi-Fi to the Arduino IoT Cloud, where it is processed and visualized. Users can access this data through the cloud platform, allowing them to check the battery's performance from any device with internet access.

Key Components:

ESP8266 Module: Used for connecting the system to Wi-Fi and transmitting data to the cloud.

Arduino IoT Cloud: The cloud service that allows users to create a dashboard for monitoring the battery's status.

Battery Sensors: To measure critical parameters like voltage and temperature.

Arduino IDE: For programming and controlling the system.

Benefits:

Scalability: The system can easily be scaled to monitor multiple batteries or deployed across various locations.

Low Cost: The use of affordable components like the ESP8266 and Arduino platform ensures the system is cost-effective, making it suitable for both industrial and personal use.

User-Friendly Interface: The Arduino IoT Cloud dashboard provides a simple, intuitive way to monitor battery health without complex configurations.

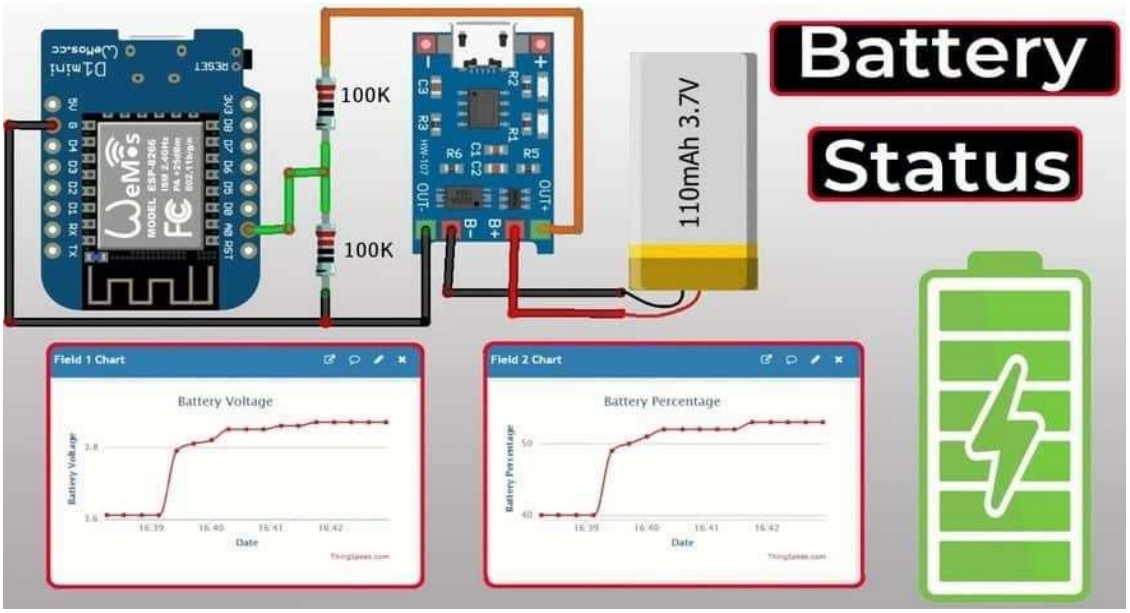


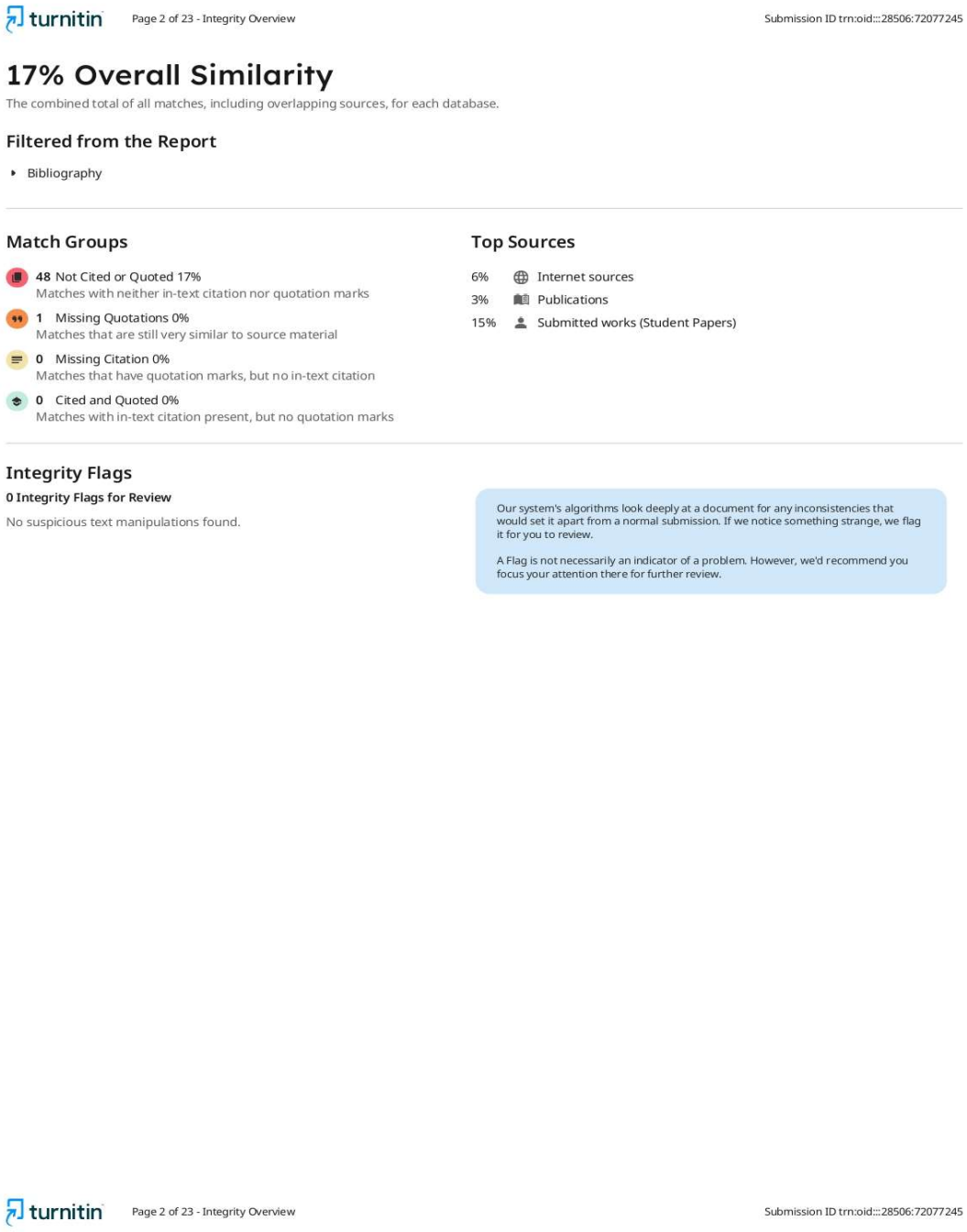
Fig 5: IoT Based Battery Status Monitoring System using ESP8266

REFERENCES

- IoT Project Side Ideas: Step-by-step guide for setup and code-IoT Projects Ideas
- ESCLabs: Overview of hardware setup and Arduino code-ESCLabs
- IRJMETS: Comprehensive explanation of components and methodology-IRJMETS
- Instructables: Arduino IoT Cloud Battery Monitor projectino Project Hub: Detailed battery monitoring guide](https://create.arduino.cc/projecthub)【6†source:contentReference[oaicite:3]{index=3}hould provide detailed tutorials, code snippets, and visualizations for setting up your battery monitoring system with ESP8266 and Arduino IoT Cloud.

TURNITIN PLAGIARISM REPORT

Please Insert a Scanned Copy of the Front pages duly signed by the Candidate, Supervisor, Departmental Turnitrin Coordinator, and HoD with Seal



turnitin Page 2 of 23 - Integrity Overview Submission ID trn:oid:::28506:72077245

17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

48 Not Cited or Quoted 17%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
3% Publications
15% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 23 - Integrity Overview Submission ID trn:oid:::28506:72077245