Implementation of RemRover: A Remotely Controlled Rover Using IOT and Websockets

Mrs. P. Shalini[1], G. Nandini[2], M. Bommaiah[3], S. Harsha Vardhan[4], K. Sree Nithya[5], P. Charan Teja[6]

[1]Assistant Professor, Dept of ECE, PBR VITS, Kavali, Andhra Pradesh, India.
[23456]Department of Electronics and Communication Engineering,
PBR Visvodaya Institute of Technology & Science, Kavali (Autonomous),
SPSR Nellore (Dt.), Andhra Pradesh – 524201, India

---***---

ABSTRACT - In this paper, The rapid expansion of Internet of Things (IoT) technologies has significantly improved the capability of robotic systems used for remote monitoring and control. The objective of this project is to design and implement RemRover, a remotely controlled rover that uses IoT and WebSocket communication to provide real-time monitoring and control. The system aims to enable remote navigation, obstacle detection, live video streaming, and environmental data monitoring while ensuring efficient data transmission.

Keywords: Internet of Things (IoT), Raspberry Pi, Web Sockets, WebSocket-based communication, Remote Monitoring

1. INTRODUCTION

With the increasing demand for automation and remote operation systems, IoT-based solutions have gained significant importance in recent years. Remote-controlled robotic systems are widely used in areas where human intervention is risky or inefficient, such as hazardous environments, military operations, and industrial monitoring.

The RemRover system is designed to provide a user-friendly platform for controlling a rover remotely using internet connectivity. Unlike traditional systems that rely on short-range communication technologies like Bluetooth or RF modules, this system uses IoT and WebSockets to enable global accessibility. WebSockets provide continuous, full-duplex communication between the client and server, allowing real-time control of the rover.

The main objective of this project is to develop a cost-effective and efficient rover that can be controlled from anywhere using a web-based interface, ensuring low latency and reliable communication.

2. LITERATURE SURVEY

Wannewar et al. (2021) provided a comprehensive survey of AI-based rescue robots, highlighting the integration of machine learning and computer vision techniques in modern robotic systems. The authors discussed various algorithms used for object detection, path planning, and victim identification. Their work emphasizes that AI significantly improves decision-making capabilities. Nevertheless, the survey indicates that many AI-based systems still face challenges related to computational complexity, real-time processing, and robustness in unpredictable disaster scenarios.

Kunal et al. (2018) introduced a robotic surveillance rover designed for monitoring applications. Their system focuses on real-time video transmission and remote operation, enabling continuous observation in security-sensitive areas. The study demonstrates how mobility and wireless communication enhance surveillance capabilities. However, the system largely relies on manual control, limiting its autonomy and adaptability in dynamic environments.

3. EXISTING METHOD

Existing remotely controlled robotic systems generally rely on basic wireless communication methods such as RF or Bluetooth and offer limited monitoring capabilities. Many traditional models provide only manual control with live video streaming, without integrating environmental sensors or cloud connectivity. Sensor data, if available, is often continuously transmitted, leading to unnecessary bandwidth usage and increased power consumption. Additionally, most existing systems lack real-time web-based control, selective data uploading, and centralized cloud storage, making remote monitoring inefficient and less scalable. These limitations reduce the effectiveness of traditional robotic rovers in applications that require intelligent control, efficient data management, and real-time environmental awareness.

In many of these systems, sensor data (if available) is continuously transmitted to the monitoring system. This constant data transmission results in high bandwidth usage, increased power consumption, and reduced efficiency. Additionally, most existing models lack cloud connectivity, which limits remote accessibility and centralized data storage.

Some systems also rely on Android-based control applications, meaning that they can only be operated from specific devices. This reduces flexibility and scalability.

4. PROPOSED METHOD

The proposed system is designed to automatically detect whether an egg is fertile or infertile using a camera-based monitoring system and image processing techniques. The system reduces human effort and improves accuracy compared to traditional manual candling methods.

In this method, eggs are placed on a tray inside a controlled environment with proper lighting. A bright LED light source is positioned behind or below the eggs to illuminate their internal structures. When the light passes through the egg, a web camera captures clear images of the egg interior.

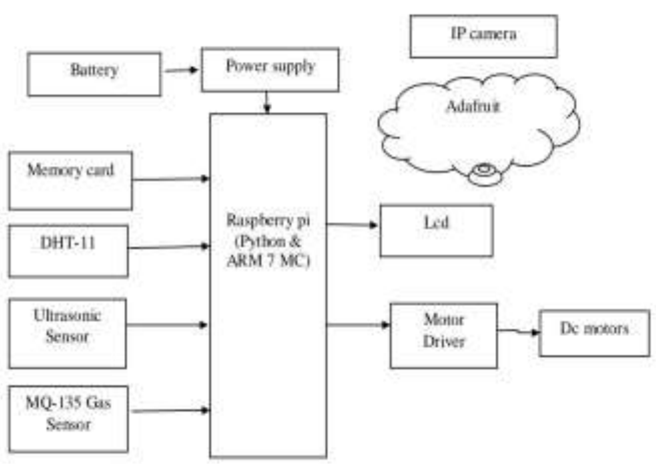

Fig 1: Block Diagram of Proposed system

5. RESULTS

The use of WebSocket protocol enabled real-time, bidirectional communication between the user interface and the rover. The system exhibited low latency, with command response times typically ranging between 50–150 ms under stable internet conditions.

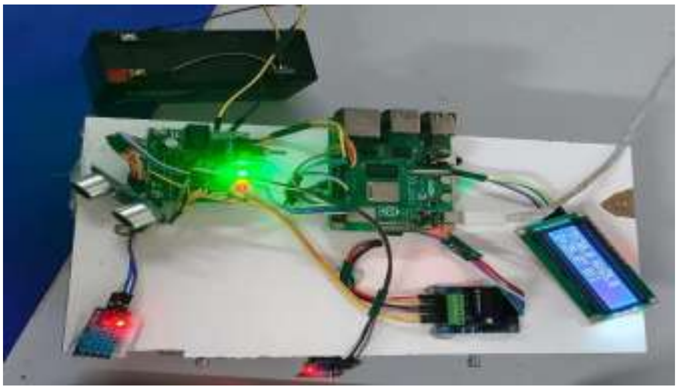

Fig 2: complete detecting system

The above Fig 2 tells about a DHT11 sensor successfully monitored environmental conditions by measuring temperature and humidity inside the system. The rover was successfully controlled from different locations using a web-based interface. The rover was successfully controlled from different locations using a web-based interface. The IoT integration allowed seamless access through internet-enabled devices such as laptops and smartphones.

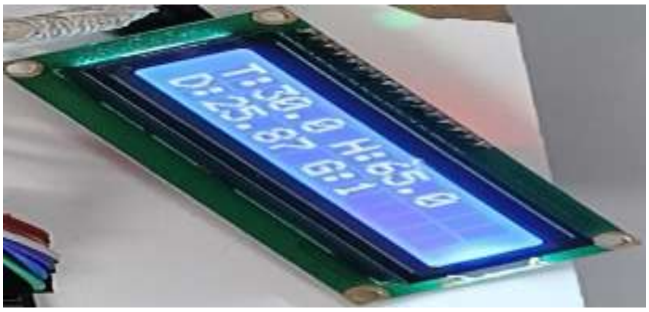

Fig 3: LCD Display

The Sensors integrated into the system, such as obstacle detection modules, performed effectively. The rover was able to detect obstacles in real time and provide feedback to the user interface. This enhances safety and usability in practical environments. The IoT integration allowed seamless access through internet-enabled devices such as laptops and smartphones. This demonstrates the

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

system's capability for global remote operation without range limitations.

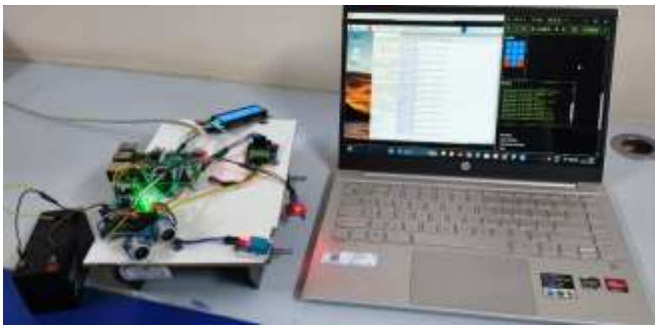

Fig 4: Results

The ultrasonic sensor reliably detected obstacles within 80 cm range and the Raspberry Pi automatically halted forward motion to prevent collisions.

Temperature & Humidity Monitoring: The DHT11 sensor provided temperature readings accurate to ±2°C and humidity readings accurate to ±5% RH, displayed both on the LCD and Adafruit IO dashboard upon request.

Gas Detection: The MQ-135 gas sensor successfully detected elevated concentrations of harmful gases including ammonia and smoke, triggering digital HIGH output and uploading readings to the cloud on demand.

The 16x2 LCD correctly displayed real-time sensor readings (temperature, humidity, gas status) during rover operation for immediate on-site reference.

6. DISCUSSION

The RemRover project focuses on the development of a remotely controlled robotic rover using Internet of Things (IoT) technology and WebSocket communication. The main objective of the project was to design a system capable of performing remote monitoring, environmental sensing, and real-time surveillance through a cloud-based interface.

During the implementation of the project, several hardware and software components were integrated to achieve the desired functionality. The Raspberry Pi was used as the main controller due to its ability to handle complex tasks such as sensor data processing, internet communication, and video streaming. The use of sensors such as the ultrasonic sensor, DHT11 temperature and humidity sensor, and MQ-135 gas sensor enabled the rover to collect environmental data from its surroundings.

The integration of the L293D motor driver allowed the Raspberry Pi to control the DC motors effectively, enabling the rover to move in different directions based on commands received from the remote user. The WebSocket communication protocol provided a reliable and efficient method for real-time communication between the rover and the cloud platform. This ensured that the user could control the rover with minimal delay.

CONCLUSION

In conclusion, the RemRover system demonstrates an efficient and practical implementation of an IoT-enabled remotely controlled rover designed for real-time surveillance and environmental monitoring. By integrating a Raspberry Pi as the central controller with sensors such as the ultrasonic sensor, DHT11, and MQ-135, along with a wireless IP camera for live streaming, the system ensures safe navigation and continuous environmental awareness.

The use of WebSocket-based communication through the Adafruit IO cloud platform enables reliable, low-latency remote control and selective data transmission, optimizing bandwidth usage while enhancing privacy and system reliability. Overall, the proposed RemRover is a cost-effective, scalable, and flexible solution suitable for applications including remote surveillance, hazardous environment monitoring, and exploratory robotic operations.

The implementation of RemRover, a remotely controlled rover using IoT and WebSockets, demonstrates an efficient and reliable system for real-time monitoring and control. By integrating a microcontroller with WiFi connectivity, the rover can receive commands instantly and respond with minimal delay, ensuring smooth and accurate movement.

Overall, the system proves to be cost-effective, scalable, and suitable for applications like surveillance, disaster management, and remote monitoring in hazardous environments. Future improvements can include advanced automation, AI-based decision making, and enhanced security features for better performance and wider applicability.

ACKNOWLEDGEMENT

The authors sincerely thank Mrs. P. Shalini (Assistant Professor, ECE, PBR VITS Kavali) for her guidance, Dr. R. Sravanthi (Professor & HoD, ECE) for providing facilities, and Dr. V. Anil Kumar (Principal, PBR VITS Kavali) for the academic environment that enabled this work.

REFERENCES

[1] M. Wannewar, V. Pandhare, K. Choubey, V. Gajbhiye, and V. Pandey, “AI Based Rescue Robot: A Literature Survey,” *Journal of University of Shanghai for Science and Technology (JUSST)*, 2021.

[2] K. Kunal, M. Suheab, K. S. Borkar, and K. H. S., “Robotic Surveillance Rover,” *IJARIIE*, 2018.

[3] J.-H. Kim, S. Jo, and B. Y. N. Lattimer, “Feature selection for intelligent fire-fighting robot classification of fire, smoke, and thermal reflections using thermal infrared images,” *Journal of Sensors*, vol. 2016, pp. 1–13, 2016.

[4] J. Reich and E. Sklar, “Robot-Sensor Networks for Search and Rescue,” ResearchGate Paper, 2006.

[5] P. G., A. S., R. M. Varghese, N. A. Mathew, and G. C. S., “Alive Human Body Detection and Tracking System Using an Autonomous PC Controlled Rescue Robot,” *IJETAE*, 2015.

[6] C. G., A. Jain, H. Jain, and Mohana, “Real Time Object Detection and Tracking Using Deep Learning and OpenCV,” 2018.

[7] F. Matsuno and S. Tadokoro, “Rescue robot systems in Japan,” in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, 2008.

[8] X. Chen and J. Li, “Research on an Efficient Single-Stage Multi Object Detection Algorithm,” 2019.

[9] S. Kobayashi *et al.*, “Development of a Door Opening System on Rescue Robot for Search,” *UMRS*, 2007.

[10] M. Sighenza, D. Guillen, and D. Arroyo, “Mobile Robots Development: A Case Study from Robotics Competitions and Course Projects,” 2017.