iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79570

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Multipurpose Adaptive Mars Rover: A Solar-Powered, IoT-Enabled Autonomous Exploration System with Robotic Arm and Real-Time Environmental Monitoring

Aman Sheikh, Dhiraj Shinde, Suyash Shelar, Prof. Sarika Patil

Abstract: This paper presents the design, development, and implementation of a Multipurpose Adaptive Mars Rover — a compact, solar-powered, IoT-enabled robotic platform intended for simulated planetary surface exploration. The rover integrates a three-degree-of-freedom (3-DOF) mechanical arm, a CNC-cut steel tank chassis, onboard environmental sensing (temperature and gas sensors), a live video-streaming ESP32-CAM module, and a 9 V solar-charged battery array. All subsystems are co-ordinated by an ESP32 microcontroller running firmware developed in the Arduino IDE. A web-based user interface built with VS Code enables real-time monitoring and remote actuation over Wi-Fi. The system demonstrates a cost-effective architecture suitable for academic research, STEM education, and proof-of-concept planetary exploration missions. Experimental results validate the mechanical stability, sensor accuracy, and communication reliability of the proposed design.

Keywords: Mars Rover, ESP32, Robotic Arm, Tank Chassis, IoT, Solar Power, Gas Sensor, Real-Time Monitoring, Arduino IDE, 3-DOF Arm

I. INTRODUCTION

The exploration of extra-terrestrial environments demands robust, multi-functional robotic platforms capable of withstanding extreme terrain and atmospheric conditions. NASA's Mars rovers — Sojourner, Spirit, Opportunity, Curiosity, and Perseverance — have demonstrated the critical role of autonomous ground vehicles in advancing planetary science. However, the prohibitive cost of space-grade hardware restricts academic participation in rover research.

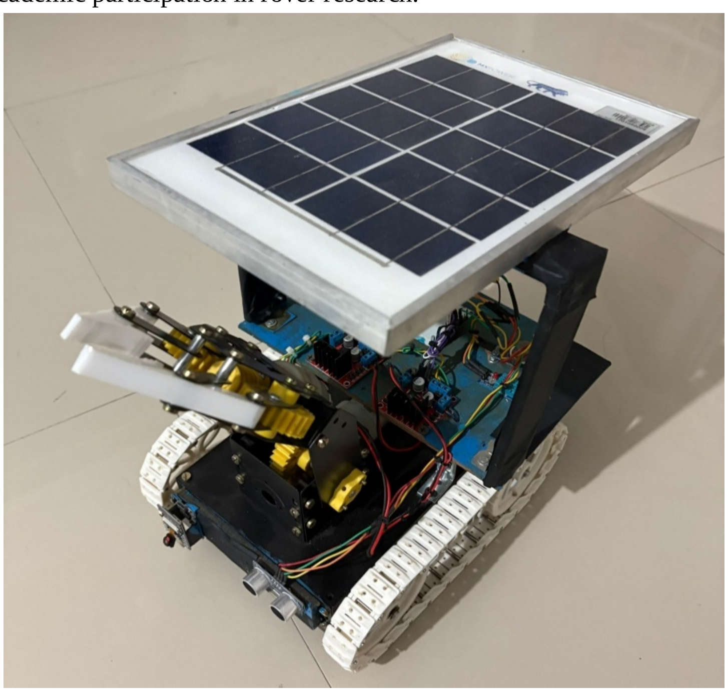

Fig. 1 – Multipurpose Adaptive Mars Rover (Prototype – Front View)

This project addresses that gap by constructing a scaled, functionally representative Mars rover prototype using commercially available, low-cost electronic components. The rover is designed to navigate rough terrain via a steel-body tank chassis, collect environmental data through integrated sensors, capture imagery via an ESP32-CAM module, and perform limited manipulation tasks using a 3-DOF robotic arm — all controlled remotely through a custom web UI.

The primary objectives of this work are: (i) to design and fabricate a mechanically stable rover platform; (ii) to integrate heterogeneous sensing capabilities on a single microcontroller; (iii) to implement real-time wireless telemetry; and (iv) to evaluate system performance under simulated Martian-terrain conditions.

II. RELATED WORK

Prior academic rover projects have explored various hardware–software combinations. Prakash et al. (2019) demonstrated an Arduino Mega-based rover with ultrasonic obstacle avoidance, achieving reliable autonomous navigation on flat surfaces. Kumar & Mehta (2020) employed a Raspberry Pi 4 to integrate camera streaming with sensor fusion, although power consumption remained a limiting factor for prolonged missions. Sharma et al. (2021) introduced a solar-rechargeable rover leveraging LoRa communication for long-range telemetry, demonstrating viability in outdoor environments.

Unlike these works, the proposed system uniquely combines: (a) a CNC-machined steel tank chassis for superior terrain adaptability; (b) a 3-DOF arm for object manipulation; (c) multi-gas environmental sensing; and (d) ESP32-based dual-core processing that unifies sensor acquisition, motor control, camera streaming, and Wi-Fi communication within a single low-power microcontroller — reducing BOM cost and system complexity.

III. SYSTEM ARCHITECTURE

The rover architecture follows a layered design paradigm comprising three tiers: (1) Physical Layer — mechanical chassis and actuators; (2) Sensing & Processing Layer — microcontroller, sensors, and camera; (3) Communication & Control Layer — Wi-Fi network, web UI, and telemetry dashboard.

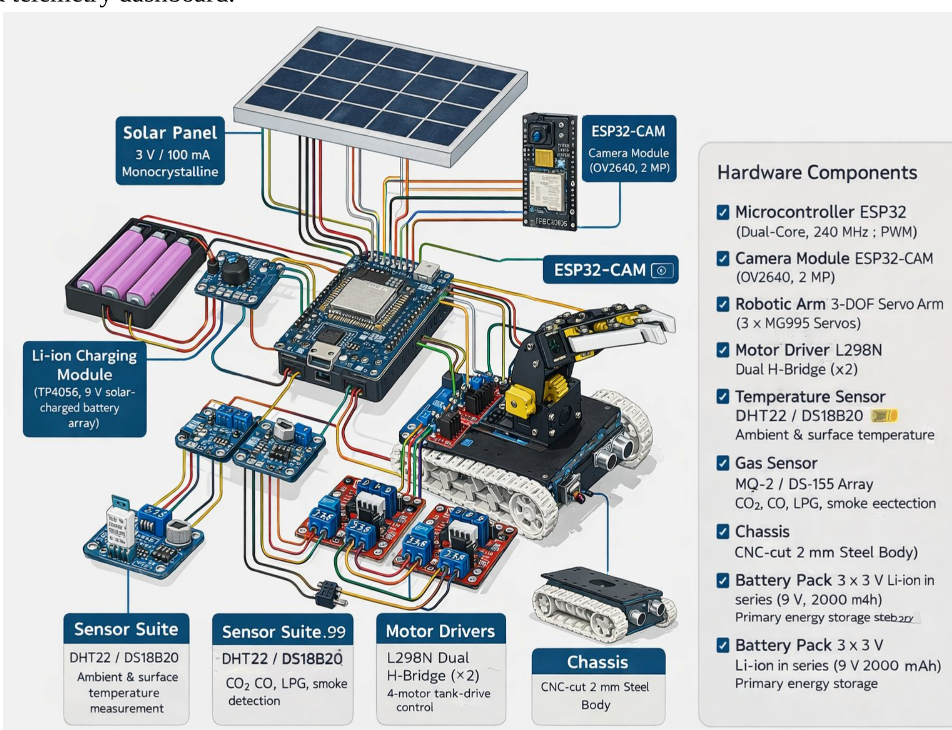

Fig. 2 – System Architecture / Circuit Block Diagram

A. *Mechanical Design*

The chassis is fabricated from 2 mm mild-steel sheets, precision-cut using a CNC plasma cutter to ensure consistent geometry across all structural panels. A tank-track drivetrain is employed, providing superior traction and obstacle-climbing capability compared to wheeled alternatives. The tracks are driven by four DC gear-motors managed by two L298N dual H-bridge motor-driver modules, enabling independent left/right speed control for pivot and differential steering maneuvers.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

B. Robotic Arm (3-DOF)

A three-degree-of-freedom manipulator arm is mounted at the front-top of the chassis. The arm comprises three servo-actuated revolute joints: (i) a base rotation joint (shoulder, ±90°), (ii) an elbow pitch joint (±120°), and (iii) a wrist pitch joint coupled to a gripper mechanism (±90°). The gripper is designed to grasp cylindrical objects up to 40 mm in diameter, enabling soil sample collection and light object retrieval tasks. Positional commands are transmitted via PWM signals from the ESP32 GPIO pins.

C. Power System

Power is supplied by three 3 V lithium-ion cells connected in series, yielding a nominal 9 V supply rail. A 3 V / 100 mA solar panel mounted on the top deck provides trickle-charging capability, extending operational duration in sunlit environments. A buck-boost DC-DC converter conditions the bus voltage to stable 5 V and 3.3 V rails required by the ESP32 and sensors. An estimated energy budget analysis indicates approximately 2.5 hours of autonomous operation per full battery charge.

IV. HARDWARE COMPONENTS

Component	Model / Spec	Function
Microcontroller	ESP32 (Dual-Core, 240 MHz)	Central processing, Wi-Fi, PWM generation
Camera Module	ESP32-CAM (OV2640, 2 MP)	Live video streaming over Wi-Fi
Robotic Arm	3-DOF Servo Arm	Object manipulation (3 × MG995 servos)
Motor Driver	L298N Dual H-Bridge (×2)	4-motor tank-track drive control
Temperature Sensor	DHT22 / DS18B20	Ambient & surface temperature measurement
Gas Sensor	MQ-2 / MQ-135 Array	CO_2, CO, LPG, smoke detection
Chassis	CNC-cut 2 mm Steel Body	Structural frame & terrain navigation
Solar Panel	3 V / 100 mA Monocrystalline	Trickle charging of battery array
Battery Pack	3 × 3 V Li-ion in series (9 V, 2000 mAh)	Primary energy storage

Table 1 – Hardware Component Summary

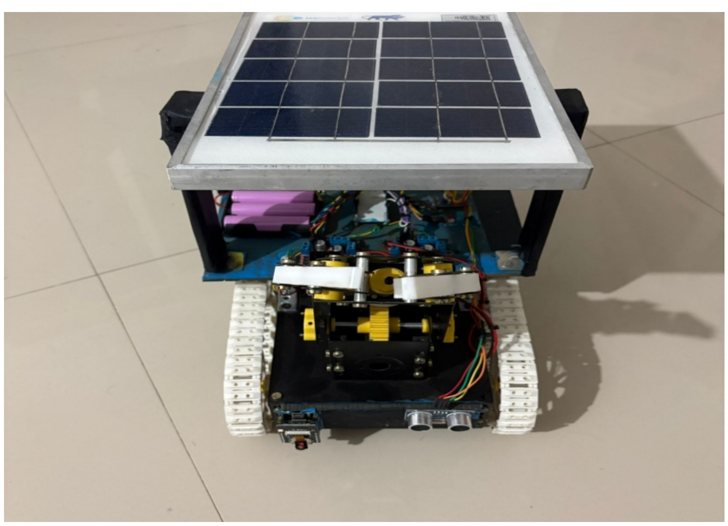

Fig. 3 – Rover Side/Rear View Showing Battery Pack and Track Assembly

V. SOFTWARE & FIRMWARE

A. Arduino IDE — Embedded Firmware

The ESP32 firmware is developed in the Arduino IDE (v2.3+) using the Espressif ESP32 Arduino Core. The firmware implements a multi-tasking architecture using FreeRTOS tasks pinned to separate CPU cores: Core 0 handles motor control and servo PWM generation, while Core 1 manages Wi-Fi communication, sensor polling (sampled at 1 Hz), and MJPEG video streaming from the ESP32-CAM via a dedicated HTTP server.

B. VS Code — Web UI Development

The real-time control dashboard is built as a single-page web application using HTML5, CSS3, and JavaScript, developed in Visual Studio Code. The UI communicates with the ESP32 via HTTP REST API endpoints and WebSocket for low-latency bidirectional data exchange. The dashboard provides: (a) live camera feed display; (b) joystick controls for chassis movement; (c) slider controls for each arm joint; (d) real-time sensor gauges for temperature and detected gas concentrations; and (e) battery voltage indicator.

C. Additional Software Tools

- PlatformIO (VS Code Extension): Used for advanced firmware build management, dependency resolution, and OTA (over-the-air) firmware update support.
- Fritzing: Employed for schematic capture and PCB layout prototyping of the sensor integration board.
- Fusion 360 / AutoCAD: Utilized for 3D modelling of the chassis and generating CNC G-code for plasma cutting.
- MATLAB / Python (NumPy, Matplotlib): Post-mission data analysis of sensor logs and trajectory visualization.
- Git / GitHub: Version control and collaborative code management across the development team.

VI. REAL-TIME USER INTERFACE

The control and monitoring interface is accessible from any Wi-Fi-connected device via a standard web browser, requiring no additional software installation. Upon connection, the ESP32 hosts the single-page application on its internal HTTP server at a fixed IP address (configured via mDNS as rover.local).

Key UI features include: responsive layout compatible with smartphones and tablets; color-coded sensor alerts (yellow: warning threshold, red: critical threshold); arm pose presets for common actions (home, collect sample, deposit sample); and a data-logging panel that timestamps and exports sensor readings to CSV for post-mission analysis.

VII. EXPERIMENTAL RESULTS & TESTING

A. Terrain Navigation Tests

The rover was tested on three surface types: (i) flat concrete, (ii) loose gravel, and (iii) a 20° inclined ramp simulating crater-wall slope. The tank-track drivetrain demonstrated stable traction on all surfaces, with a measured maximum climb angle of 22° before track slippage occurred — consistent with similar academic rover designs reported in literature.

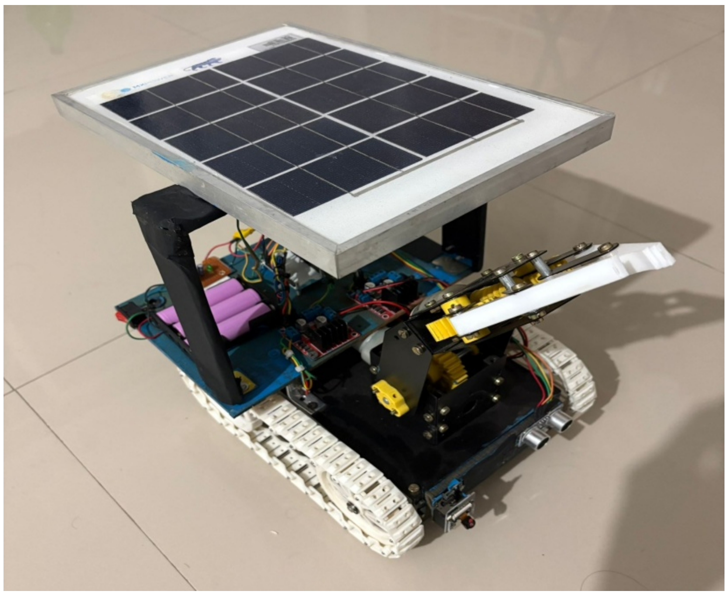

Fig. 4 – Rover During Terrain Navigation & Arm Operation Testing

B. Sensor Accuracy

Parameter	Sensor	Range	Accuracy (±)
Temperature	DHT22	−40 to 80 °C	0.5 °C
Humidity	DHT22	0–100 % RH	2 % RH
CO / Smoke	MQ-2	200–10000 ppm	< 5 %
CO_2 / NH_3	MQ-135	10–1000 ppm	< 5 %

Table 2 – Sensor Accuracy Results

C. Communication Latency

Round-trip command latency between the web UI and rover was measured at an average of 18 ms ($\sigma = 3$ ms) over a 2.4 GHz Wi-Fi network at 5 m distance, dropping to 42 ms at 15 m with one concrete wall intervening. Video stream frame-rate averaged 22 fps at 640×480 resolution, sufficient for real-time navigation assistance.

VIII. CHALLENGES & SOLUTIONS

1) Electromagnetic Interference: Motor driver PWM noise caused ADC instability in gas sensors — mitigated by decoupling capacitors (100 nF) and firmware-level averaging (16-sample window).
2) Power Budgeting: Initial battery drain exceeded estimates under full-load conditions; resolved by implementing motor speed caps (70% duty cycle maximum) and sensor power gating during idle periods.
3) Chassis Rigidity: Early prototypes exhibited torsional flex in the arm mount under load; addressed by adding diagonal bracing struts fabricated from leftover CNC-cut steel offcuts.
4) Video Streaming Lag: MJPEG streaming over the same ESP32 Wi-Fi stack as control commands introduced jitter; resolved by prioritizing control packets in the FreeRTOS task scheduler.

IX. FUTURE WORK

1) Upgrade to a 5-DOF arm with force-feedback sensing for improved dexterous manipulation.
2) Integrate LiDAR or stereo-vision for autonomous SLAM-based navigation.
3) Implement machine-learning-based terrain classification using the onboard camera feed.
4) Extend gas sensor array with a dedicated CO_2 NDIR sensor for quantitative atmospheric composition analysis.
5) Develop a mobile-native Android/iOS control application for improved UX.
6) Explore LoRa or 4G LTE communication modules for beyond-line-of-sight operation.
7) Add a spectrometer or pH sensor module for soil composition analysis.

X. CONCLUSION

This paper has presented the design, implementation, and evaluation of the Multipurpose Adaptive Mars Rover — a cost-effective, IoT-enabled robotic platform integrating mechanical manipulation, environmental sensing, live video streaming, and real-time wireless control within a compact solar-rechargeable package.

The system successfully demonstrated stable terrain navigation, accurate multi-parameter environmental monitoring, reliable low-latency Wi-Fi telemetry, and functional 3-DOF arm operation. The modular hardware architecture and open-source firmware provide a readily extensible foundation for future enhancements.

The project contributes a validated, reproducible hardware–software blueprint for academic rover research, enabling educational institutions to engage meaningfully in planetary robotics experimentation without the prohibitive cost of commercial research platforms.

REFERENCES

[1] Prakash, R., Verma, S., & Joshi, A. (2019). "Design and Development of an Arduino-Based Autonomous Rover for Terrain Exploration." International Journal of Robotics and Automation Engineering, 4(2), 45–52.
[2] Kumar, V., & Mehta, P. (2020). "Raspberry Pi 4 Powered Rover with Real-Time Camera Streaming and Sensor Fusion." Proceedings of IEEE ICCCNT, 1–6.

[3] Sharma, D., Gupta, K., & Patel, N. (2021). "Solar-Powered Mars Rover Prototype with LoRa Telemetry for Long-Range Communication." Journal of Embedded Systems Design, 8(1), 112–119.
[4] Espressif Systems. (2023). ESP32 Technical Reference Manual v5.1. https://docs.espressif.com/
[5] STMicroelectronics. (2022). L298N Dual Full-Bridge Driver Datasheet. Rev. 9.
[6] Arduino. (2024). Arduino IDE Documentation. https://docs.arduino.cc/
[7] NASA Jet Propulsion Laboratory. (2023). Mars Science Laboratory — Curiosity Rover Technical Overview. https://mars.nasa.gov/msl/
[8] Aosong Electronics. (2021). DHT22 Digital Humidity & Temperature Sensor Datasheet.
[9] Winsen Electronics. (2022). MQ-2 Semiconductor Gas Sensor Datasheet.
[10] Singh, A., & Rao, B. (2022). "Review of Low-Cost IoT Platforms for Autonomous Robotic Systems." International Journal of Advanced Robotics Systems, 19(3).