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

IoTerrain: An IOT-Based Six-Wheeled Rover for Environmental Data Collection

Dr. Soni Chaturvedi,Daisy Dahegonkar,Dipali Kuthe,Smital Burde,Sanskruti Borkar

Department of Electronics and Communication Engineering

***Abstract*—**Environmental observation in outdoor, agricultural, industrial, and difficult-to-reach locations can be limited by the need for people to enter the area and take measurements manually. This paper presents the design of IoTerrain, a six-wheeled mobile rover intended to collect environmental readings and make them available through an Internet of Things (IoT) interface. The proposed platform combines a six-wheel chassis, an embedded controller, a motor-driving stage, environmental sensors, a wireless connection, and a user-facing monitoring dashboard. The rover is intended to move to selected locations, acquire sensor measurements, associate them with time and location information when available, and transmit the readings for remote observation. A modular architecture is described so that sensors can be changed to match the application, such as temperature, humidity, soil moisture, or gas-level monitoring. The paper details the system architecture, hardware and software roles, operating sequence, implementation plan, and evaluation criteria. Since project-specific test logs and measured results were not supplied for this draft, performance values are not claimed; the proposed tests are provided to guide validation on the completed prototype.

***Keywords*—**Internet of Things, six-wheeled rover, environmental monitoring, ESP32, mobile robot, sensor network, remote dashboard.

I. INTRODUCTION

Environmental conditions vary across locations and over time. Measurements collected at only one fixed point may not represent an entire field, worksite, or outdoor area. Manual sampling can also require repeated visits and may be inconvenient where access is uneven or exposure should be minimized. A mobile sensing platform can address part of this problem by carrying sensors to different points and reporting readings to an operator.

The proposed IoTerrain rover uses a six-wheeled mechanical platform to support movement over paths that may be less uniform than indoor floors. The IoT layer provides a way to view collected readings remotely, while the embedded controller coordinates sensing and motion commands. The purpose is not to replace calibrated environmental instruments; rather, the rover provides a flexible platform for gathering preliminary measurements and observing spatial variation.

The main contributions of this paper are: (1) a modular architecture for a six-wheel environmental sensing rover; (2) a description of the sensing, movement, and IoT data path; (3) a practical implementation sequence; and (4) a validation plan that distinguishes measured results from design expectations.

II. RELATED WORK

IoT systems connect embedded devices, sensors, and network services so that measurements can be collected and reviewed beyond the physical device [3]. Mobile robots extend this approach by moving sensing equipment through an area instead of relying only on stationary nodes. Recent rover designs have combined ESP32-class controllers, environmental sensors, wireless control, and web-based monitoring. For example, an IEEE conference paper describes an IoT-enabled rover using an ESP32 platform for environmental monitoring and web-based control [1]. A separate rover telemetry study describes collecting environmental and platform data and presenting it through an IoT service [2].

Six-wheel arrangements are commonly selected when a mobile platform needs multiple ground-contact points and a chassis suitable for outdoor movement. Actual mobility depends on wheel size, ground clearance, chassis geometry, motor torque, load distribution, and terrain. Therefore, a six-wheel design should be evaluated on the intended surface rather than assumed to be suitable for every terrain. The proposed work combines these general design ideas in a project-oriented, modular environmental monitoring platform.

III. PROPOSED SYSTEM

A. System Overview

The rover is organized into five functional blocks: mobility, sensing, embedded control, wireless communication, and remote presentation. The mobility block contains the chassis, six wheels, drive motors, and motor driver. The sensing block contains the environmental sensors selected for the final build. The controller reads sensor outputs, interprets operator commands, and issues motor-control signals. A wireless connection transfers readings to a monitoring interface when network coverage is available.

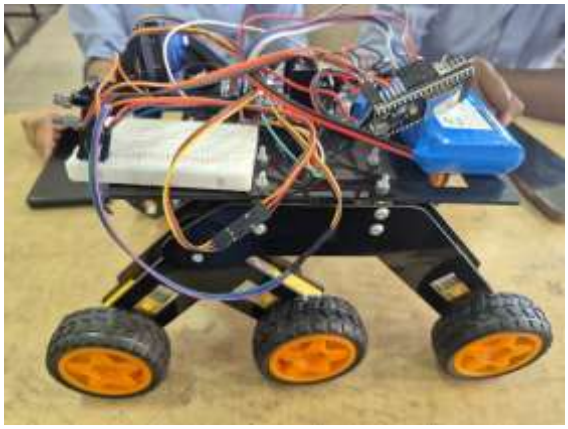


Fig. 1. IoTerrain six-wheeled rover prototype for environmental data collection.

B. Functional Architecture

Environmental sensors → ESP32/controller → Wi-Fi or supported IoT link → data service/dashboard

Operator command → controller → motor driver → six-wheel drive

The two paths should be handled with suitable safeguards. A lost network connection should not cause unexpected motor movement; the software should stop or hold the rover in a defined safe state when remote control is unavailable. Sensor readings should include timestamps where possible, and the dashboard should distinguish missing data from a valid zero reading.

C. Six-Wheel Mobility

The chassis supports six wheels arranged along the left and right sides. Depending on the selected drivetrain, motors may be grouped by side or controlled in smaller groups. Differential steering can be implemented by driving the left and right sides at different speeds or directions. A motor driver must be selected according to the motor voltage, stall current, number of independently controlled channels, and heat dissipation requirements. The exact driver should be confirmed against the motors used in the prototype; it should not be chosen from wheel count alone.

D. Environmental Sensing

Sensor selection depends on the intended application. A temperature-and-humidity sensor can describe ambient conditions; a soil-moisture probe can support field observations; and an appropriate gas sensor can indicate changes in the presence of target gases. Each sensor requires correct wiring, calibration, and interpretation of its operating range. Low-cost gas and soil sensors may be affected by humidity, temperature, placement, ageing, and soil composition. Their values should therefore be treated as indicative unless validated against a suitable reference instrument [5].

E. IoT Monitoring

The controller packages each sample into a consistent record containing the sensor name, value, unit, timestamp, and optional rover position. The record is sent to an IoT endpoint or local dashboard over the supported wireless connection. The interface can display the latest reading, recent history, and connection status. If cloud storage is used, network credentials should be kept out of public source code and stored configuration should be protected.

IV. HARDWARE AND SOFTWARE DESIGN

A. Hardware Components

The expected component groups are: (i) six-wheel chassis and wheels; (ii) DC geared motors; (iii) an ESP32 or equivalent controller; (iv) a compatible motor driver; (v) selected environmental sensors; (vi) battery and regulated power distribution; (vii) wiring, connectors, and mounting hardware; and (viii) optional position or obstacle sensors. These are functional requirements rather than a claim that every item has already been installed. The final bill of materials should list exact part numbers, quantities, ratings, and costs for the actual prototype.

B. Controller and Interfaces

An ESP32 is a suitable candidate where integrated Wi-Fi and a compact embedded controller are useful [4]. Digital sensors may use I^2C, SPI, UART, or GPIO depending on the chosen module; analog sensors require compatible input conditioning and attention to the controller's input-voltage limits. Motors should be powered through the motor driver, not directly from controller pins. Motor and logic power should be planned to reduce resets caused by current surges, and all connected modules should share a suitable reference ground where required by the circuit.

C. Software Modules

The firmware can be separated into initialization, motion control, sensor acquisition, data formatting, communication, and fault handling. Initialization checks that required modules respond. Sensor acquisition samples each channel at a defined interval. Data formatting attaches units and timestamps. Communication publishes the record and reports transmission status. Fault handling checks for sensor timeouts, low battery if measurable, and loss of remote commands. The dashboard can be implemented as a web page or an IoT platform interface; the final paper should name the software and service actually used.

V. WORKING PRINCIPLE

After power-up, the controller initializes the sensors, motor driver, and communication interface. The operator selects a movement command from the control interface. The controller converts the command into direction and speed signals for the motor driver, which supplies the required motor current. While the rover moves or pauses at a sampling point, the controller reads the environmental sensors at configured intervals. Each valid sample is packaged and transmitted to the monitoring interface. The operator can review the readings and decide where the rover should move next.

A typical operating cycle is: power on → initialize modules → connect to network → receive movement command → move to sampling point → read sensors → transmit readings → update dashboard → repeat. If communication fails, the control logic should enter the predefined safe state and attempt reconnection without issuing uncontrolled movement commands.

VI. IMPLEMENTATION PROCEDURE

The prototype should be assembled and verified in stages. First, inspect the chassis, wheel alignment, motor mounts, and battery arrangement. Next, test each motor independently with the selected driver while the wheels are safely lifted. Confirm the controller's logic supply and verify that motor current does not exceed driver or battery ratings. Then connect one sensor at a time and check that readings are plausible and expressed in the correct units. Calibrate sensors according to their datasheets and intended use.

After the local functions work, implement the network connection and send a small test record to the dashboard. Confirm that timestamps, units, and sensor labels appear correctly. Integrate remote movement commands only after communication and stop behavior have been checked. Finally, test the complete system on a controlled surface before conducting field trials. Record firmware version, sensor

models, sampling interval, payload format, battery configuration, and test conditions so that the work can be reproduced.

VII. PERFORMANCE EVALUATION PLAN

Because no measured project logs were supplied with the request, this draft does not state numerical accuracy, range, latency, battery life, or success-rate results. The completed prototype should be evaluated using repeatable tests, and the results table should be populated only with actual observations.

Recommended tests include: (1) sensor comparison against a reference instrument; (2) repeated readings at a fixed location to assess variation; (3) packet-delivery and dashboard-update checks at known distances; (4) command-response delay and safe stopping during connection loss; (5) runtime under a stated battery and payload condition; and (6) mobility trials on the actual intended surfaces. For each test, report the number of trials, method, environmental conditions, and limitations. Suggested metrics are sensor error relative to a reference, successful transmissions divided by attempted transmissions, median and maximum update delay, operating time per charge, and completion rate for a defined route. Any reported result should identify the test setup and avoid generalizing beyond the tested conditions.

VIII. APPLICATIONS AND LIMITATIONS

Potential applications include preliminary environmental surveys, agricultural field observation, inspection of industrial areas, educational robotics, and monitoring locations where a mobile sensor platform is useful. The modular design allows different sensors to be fitted for different tasks, provided their electrical and measurement requirements are addressed.

The system also has limitations. Wi-Fi-based control depends on network availability and may not provide long-range coverage. Low-cost sensors may require calibration and may not be suitable for regulatory measurements. Wheel traction and stability depend on terrain and payload. Battery runtime depends on motor load, sensor load, communication activity, and battery condition. These limitations should be discussed using observations from the actual build.

IX. CONCLUSION AND FUTURE WORK

This paper has presented the architecture and planned implementation of IoTerrain, an IoT-based six-wheeled rover for environmental data collection. The design combines a mobile chassis, embedded control, configurable sensors, wireless data transfer, and a remote monitoring interface. Its modular structure supports adaptation to different sensing tasks, while staged testing provides a practical path toward safe integration. The paper is a design-and-implementation draft rather than a report of verified experimental results. Future work can include a calibrated sensor suite, improved communication for low-coverage locations, location tagging, local data buffering during network outages, power optimization, and systematic field evaluation.

ACKNOWLEDGMENT

The authors may acknowledge the project guide, department, institution, laboratory staff, and any individuals who provided equipment or technical support. Replace this paragraph with acknowledgments that accurately reflect the support received.

REFERENCES

[1] M. Vimal, K. Sathyabama, and A. Abishek, “Design and implementation of an IoT-enabled rover with ESP32 CAM for object detection, environmental monitoring, and web-based control,” in *Proc. 2025 Int. Conf. Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, 2025, doi: 10.1109/ICMSCI62561.2025.10894020.

[2] A. S. Vinokurova, T. Jaimes, and A. Abdrasulov, “Development and implementation of a wireless telemetry system for a mechanical rover prototype,” in *Proc. LACCEI Int. Multi-Conf. Eng., Educ., Technol.*, 2023, doi: 10.18687/LACCEI2023.1.1.1514.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] Espressif Systems, “ESP32 series datasheet.” Consult the datasheet for the exact ESP32 variant used in the prototype.

[5] Manufacturer datasheets for the selected environmental sensors, motor driver, and power supply used in the final prototype (add exact manufacturer, document title, revision, and access date).

Author checklist before submission: Replace all bracketed institution details; confirm that each listed component matches the built rover; add a real block diagram and circuit diagram if required; insert actual test data and photographs; remove any feature not implemented; and verify reference details against the original publications.