PROGRESS IN ENGINEERING APPLICATION AND TECHNOLOGY
e-ISSN: 2773-5303

PEAT

Vol. 7 No. 1 (2026) 328-338
https://publisher.uthm.edu.my/periodicals/index.php/peat

Geofencing-Based Real-Time Monitoring System for Endangered Animal Tracking and Protection

Tasha Nadira Abdullah[1], Aimi Syamimi Ab Ghafar[1]*

[1] *Department of Engineering Technology, Faculty of Engineering Technology*
Universiti Tun Hussein Onn Malaysia, Pagoh, 84600, Malaysia

*Corresponding Author: aimi@uthm.edu.my
DOI: https://doi.org/10.30880/peat.2026.07.01.038

Article Info

Received: 12 January 2026
Accepted: 20 February 2026
Available online: 25 May 2026

Keywords

Real-time, MPU6050, animal tracking, UPS HAT E, Wildlife Conservation, GPS Module NEO-6M, IoT-Based Monitoring

Abstract

This project is to develop a real-time animal tracking and monitoring system to enhance the conservation of endangered species. By integrating Internet of Things (IoT) technology with geofencing and data analytics, the project aims to address the limitations of traditional wildlife monitoring methods that often lack real-time data and structured analysis. The system utilizes a Raspberry Pi 4 microcontroller equipped with a UPS HAT module to ensure uninterrupted operation, a GPS module for location tracking, and an MPU6050 accelerometer and gyroscope sensor for motion detection and behavioural pattern analysis. The systme use Telegram for delivering real-time notifications and monitoring updates, providing a cost-effective and accessible solution that enables conservationists to receive immediate alerts on standard mobile devices without requiring specialized infrastructure. With the implementation of geofencing, automated alerts are triggered via Telegram when animals move beyond designated safe zones or into high-risk areas, enabling rapid response from conservation teams to prevent poaching incidents, human-wildlife conflicts, or other threats. Ultimately, the project seeks to support data-driven decision-making and improve the effectiveness of protection strategies for endangered wildlife through continuous real-time monitoring.

1. Introduction

The conservation of endangered species faces critical challenges as traditional monitoring methods prove lacking for modern wildlife protection needs. Current approaches such as camera traps and manual tracking cannot provide continuous real-time location data, lack immediate alert mechanisms when animals enter dangerous zones, and fail to detect behavioral anomalies that may indicate distress or threats [1] These limitations result in delayed responses to poaching activities, human-wildlife conflicts, and other critical situations, ultimately compromising the effectiveness of conservation efforts. Furthermore, these traditional methods are resource-intensive, requiring substantial manpower and time for data collection and analysis, making them impractical for large-scale or long-term monitoring operations[2]

Existing wildlife tracking systems face more challenges that make the systems less effective and harder to access. Commercial tracking solutions are very expensive, making wide deployment difficult for many conservation organizations, especially in developing countries where most endangered wildlife lives[3]. Reliable power supply in remote wilderness areas remains a critical challenge, as power interruptions may result in data loss during crucial monitoring periods, potentially losing key information about animal movements during dangerous time[4]. Furthermore, current systems usually do not have motion sensors for behaviour monitoring

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 328-338

and easy communication platforms for sending real-time alerts to field workers, creating problems for quick threat response and making it harder to detect early warning signs of danger to tracker animals. [5][6]

This project proposes an enhanced real-time monitoring system through geofencing to track and protect endangered animals. The system will combine an animal tracking device based on Raspberry Pi 4 equipped with UPS HAT module, a GPS module for location tracking, and an MPU6050 sensor for motion detection and behavioural pattern analysis. The collected data is transmitted in real-time through Telegram, providing conservationists with immediate notifications and monitoring updates on standard mobile devices without requiring specialized infrastructure. The system implements geofencing technology to trigger automated alerts via Telegram when animals move beyond designated safe zones or enter high-risk areas, enabling rapid response from conservation efforts by facilitating continuous real-time monitoring, efficient data transmission, and data-driven decision-making for wildlife protection strategies.

2. Methodology

The section outlined the methodology employed in developing the Geofencing-Based Real-Time Monitoring System for Endangered Animal Tracking and Protection. The system integrated GPS-based location tracking and motion sensing using IoT devices, combined with geofencing logic to enable real-time monitoring of animal movement. Key hardware components included a Raspberry Pi, NEO-6M GPS module, MPU6050 sensor, and UPS HAT E module, which collectively supported location tracking, animal activeness monitoring, and battery management. A telegram-based alert system was implemented to provide instant notifications when geofence boundaries were breached, abnormal movement was detected, or battery levels fell below a defined threshold. The methodology aimed to enable conservationists to monitor animal behavior remotely, receive timely alerts, and ensure reliable system operation during field deployment.

2.1 System Design

Figure 1 illustrates the block diagram of the geofencing-based real-time animal tracking and monitoring system. The GPS module provides real-time location data, while the MPU6050 sensor monitors animal activeness. The UPS HAT E module supplies power and enables battery status monitoring. All sensor data are processed by Raspberry Pi using Python-based algorithms to perform geofencing analysis and system monitoring. When a geofence breach, abnormal movement, or low battery condition di detected, alert notification is sent to the user via the Telegram messaging platform.

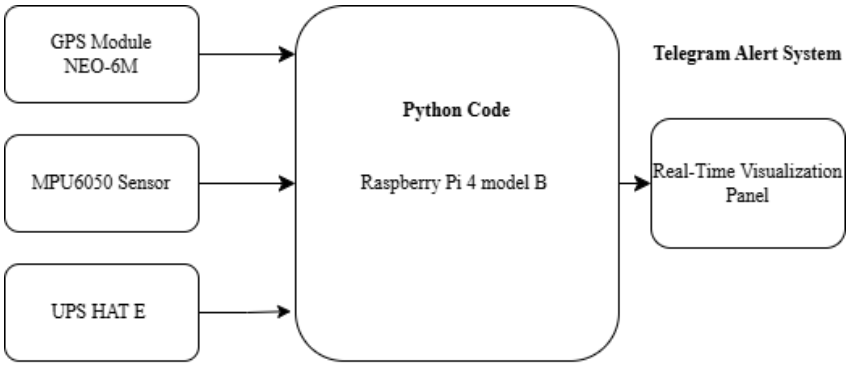


Fig. 1 *Block Diagram of the system*

2.2 Hardware Development

The circuit connection, shown in Figure 2, Raspberry Pi acts as the central processing unit, receiving real-time location data from the GPS module to determine the animal's position relative to the predefined geofence boundary, while motion data from the MPU6050 sensor is obtained via the I^2C interface to analyze animal activeness. All hardware components are powered and coordinated through the Raspberry Pi, ensuring continuous data acquisition and processing. The assembled hardware system was tested to verify stable communication and accurate sensor readings before being fully integrated with the software logic, enabling real-time alert notifications to be sent directly to users through a Telegram bot.

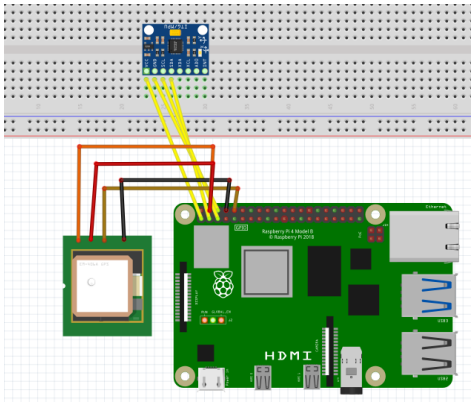


Fig.2 *Circuit connection of the system*

2.3 Software Development

Figure 3 describes a GPS-based geofencing alert system that continuously monitors a device's location and sends notifications based on proximity to a predefined center point. The system starts by fetching current GPS coordinates, then uses the Haversine Formula to calculate the distance between the current location and the geofence center. It classifies the location into three concentric zones: "Inner Zone" (within the base radius – safe area), a "Warning Zone" (radius + 10m – approaching boundary), and a "Alert Zone" (radius + 100m – outside safe perimeter). After determining which zone the device is in, the system sends an alert via Telegram and displays the status including distance, zone classification, speed, and time.

Figure 4 describes an MPU6050-based motion detection system that monitors accelerometer and gyroscope data to detect movement and classify activity states. The system begins by initializing the MPU6050 sensor and checking if initialization is successful (if not, it loops back to retry). Once initialized, it performs accelerometer baseline calibration to establish reference values for detecting motion. The system then continuously reads accelerometer and gyroscope data and checks if the sensors read was successful (retrying if failed). After removing baseline offsets from the readings, it evaluates three sequential conditions: first, checking for shock/impact detection; second, checking for tilt detection (change in orientation); and third, if either is detected, calculating the movement magnitude. If the magnitude exceeds a predefined threshold, the system classifies the state as "ACTIVE" (motion detected), otherwise it remains in "RESTING" (no significant motion). Finally, the system displays the current sensor data and status, then loops back to continuously monitor for motion.

Figure 5 describes a battery monitoring and alert system that continuously checks the battery voltage and sends a notification when it drops below a critical threshold. The system starts by initializing the I2C bus and setting up peripherals for communication with battery monitoring hardware. It then enters a continuous monitoring loop that reads the current battery state data from the connected battery management system or voltage sensor. The system compares the current time with the last charging time to calculate how long the battery has been discharging, then displays both the charging/discharging status and the current battery percentage or voltage level. A decision point checks whether the battery level has fallen below a predefined threshold (typically around 20-30% or a specific voltage like 3.3V). If the battery is critically low, the system sends an alert via Telegram to notify the user or administrator that the battery needs charging. After sending the alert, the system waits for 2 seconds (to avoid sending repeated rapid alerts and to reduce processing load), the loops back to read the battery state again.

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 328-338

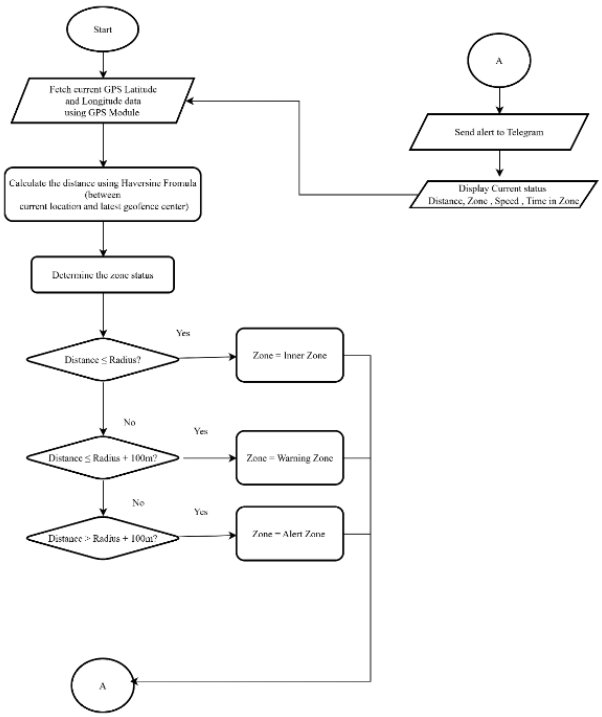


Fig.3 *Flowchart of the system for location detection*

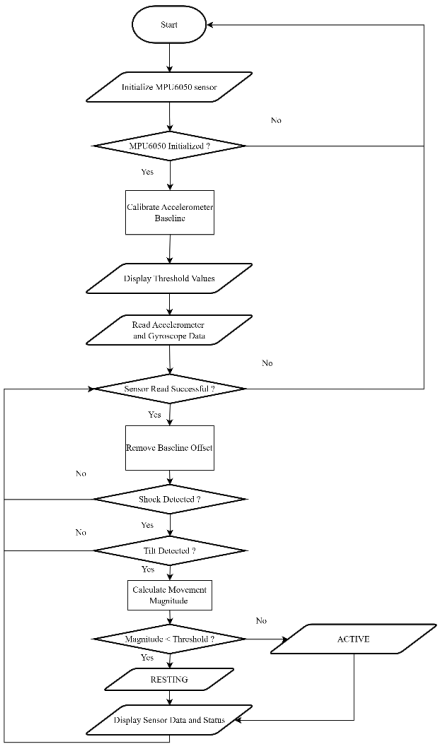


Fig.4 *Flowchart of the system for motion detection*

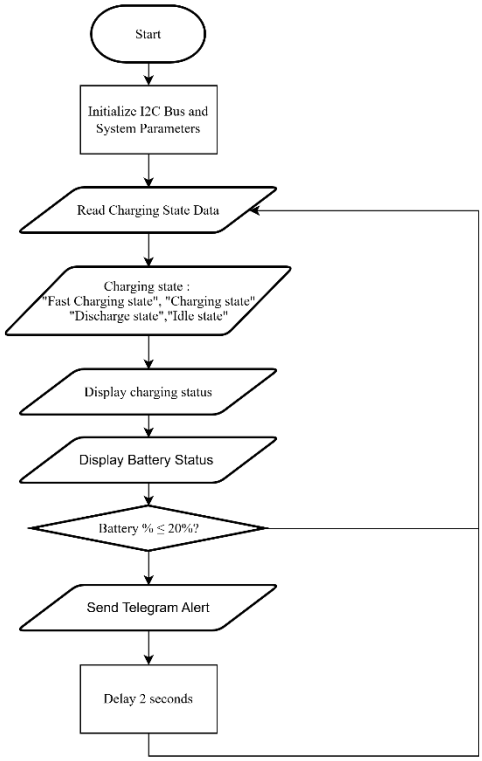


Fig.4 *Flowchart of the system for battery alert*

3. Results and Discussion

This section presents the results obtained from the development and testing of the animal tracking and geofencing system. The primary focus is on evaluating the system's functionality, reliability, and real-time performance in tracking animal movements and ensuring boundary compliance using GPS technology. Key findings, data analysis, and insights gained from the implementation process discussed, highlighting the system's effectiveness in meeting the project objectives.

3.1 Accuracy of GPS location detection capability.

Table 1 presents the results of the GPS location detection capability of the system. It compares the latitude and longitude values obtained from the GPS module (Detected GPS Location) with the reference coordinates of the actual location to evaluate the accuracy of the system.

The results obtained from the GPS module highlight its capability to provide accurate location detection when compared with reference GPS data recorded via Google Maps. The comparison shows that the average percentage error for latitude is 0.004150, while the longitude records an even smaller average error of 0.000024. These extremely low percentage errors demonstrate the GPS module's precision in determining geographical coordinates, ensuring that it provides reliable data for tracking and geofencing applications.

In summary, the results validate the system's capability to detect and report GPS locations with minimal errors. The combination of low percentage errors and a relatively small average distance deviation make this GPS module a reliable choice for real-time tracking applications. Its consistent performance ensures that it meets the accuracy requirements necessary for the geofencing functionalities described in this system.

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 328-338

Table 1 *Results of GPS location detection capability*

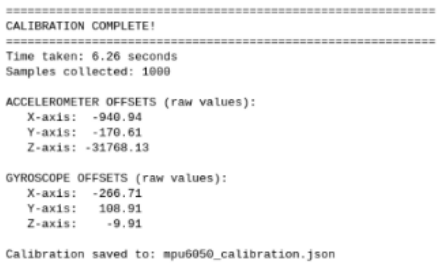
Detected GPS location		Real GPS location		Percentage of Error (%)		Distance between both locations (m)
Latitude	Longitude	Latitude	Longitude	Latitude	Longitude	
2.741793	102.132980	2.741906	102.133004	0.004121	0.000023	12.78
2.741793	102.132980	2.741906	102.133003	0.004121	0.000023	12.75
2.741793	102.132980	2.741906	102.133004	0.004121	0.000023	12.78
2.741793	102.132980	2.741906	102.133005	0.004121	0.000024	12.80
2.741793	102.132980	2.741906	102.133006	0.004121	0.000025	12.83
2.741793	102.132980	2.741907	102.133005	0.004158	0.000024	12.91
2.741793	102.132980	2.741907	102.133006	0.004158	0.000025	12.93
2.741793	102.132980	2.741908	102.133005	0.004194	0.000024	13.02
2.741793	102.132980	2.741908	102.133005	0.004194	0.000024	13.02
2.741793	102.132980	2.741908	102.133006	0.004194	0.000025	13.04
Average Value:				0.004150	0.000024	12.88

3.2 Animal Activeness Analysis

Animal activeness analysis using gyroscope and accelerometer sensors involves attaching an MPU6050 or similar inertial measurement unit (IMU) to an animal to continuously monitor its movement patterns and activity levels. The accelerometer measures linear acceleration along three axes (x,y,z) to detect changes in velocity and orientation, capturing movements like walking, running, jumping, or resting, while the gyroscope measures rotational motion and angular velocity to detect turning, spinning, or changes in body orientation. The raw sensor data is processed using algorithms that calculate movement magnitude, detect sudden shocks or impacts, and classify activity states based on predefined thresholds calibrated for the specific animal species and size.

3.2.1 Accelerometer Analysis

Figure 5 shows the MPU6050 sensor was first calibrated using 1000 stationary samples to eliminate inherent bias and offset in the accelerometer and gyroscope readings. The calibration results indicate non-zero offsets on all accelerometer axes, with a significant Z-axis offset caused by the constant effect of gravitational acceleration. Experimental observations during the resting condition show that these fluctuations result in magnitude variations of up to approximately ±0.2 m/s^2. In contrast, intentional movements produce magnitude deviations that exceed 0.6 m/s^2 from the resting reference value. Based on these observations, a delta magnitude threshold of 0.5 m/s^2 was selected. This threshold value is sufficiently higher than the noise level observed in the calibrated resting state, while remaining sensitive enough to detect genuine movement.



Fiq.5 *Calibration of MPU6050 sensor*

Table 2 shows an analysis of the MPU6050 sensor's accelerometer data that was done to determine animal activeness, focusing on linear motion along the X-, Y-, and Z-axes. After calibration, the accelerometer readings remained near 0 g when the sensor was stationed, so showing effective control of offset errors and noise. During periods of animal movement, large changes in acceleration values were observed across all axes, which

corresponded to variations in activity intensity and posture. This variation served as the basis for recognizing between low activity (resting) and high activity (active movement). The stable baseline and the responsive acceleration changes all show the accuracy of the accelerometer data for real-time monitoring of animal activeness.

Table 2 *Accelerometer Data Analysis Results*

Axis	Calibrated Offset (raw)	Expected Value (Stationary)	Observed Behavior	Interpretation
X	-940.94	0.05	Small fluctuations around 0g	Stable horizontal motion
Y	-170.61	0.04	Minimal variation when still	Low noise level
Z	-31768.13	0.06	Slight variation due to gravity	Normal vertical response

3.2.2 Gyroscope Analysis

Table 3 shows how the device rotates along the X, Y, and Z axes. The X and Y axes have small values, which means the device does not tilt much from side to side. The Z axis shows higher peaks, which means the device can detect turning or spinning movements. This is important for the project because it proves that the device can measure rotation accurately. Together with the accelerometer, the gyroscope helps track both movement and orientation of the device, which is useful for monitoring or alerting in real-time.

Table 3 *Gyroscope Data Analysis Results*

Axis	Calibrated Offset (raw)	Expected Value (Stationary) (deg/s)	Observed Behavior	Interpretation
X	-266.71	≈ 0	Small variations during motion	Detects roll movement
Y	108.91	≈ 0	Clear changes during rotation	Detects pitch movement
Z	-9.91	≈ 0	Responsive to turning motion	Detect yaw movements

3.2.3 Movement Magnitude Analysis

Table 4 shows the movement magnitude analysis of the system. The results indicate that the use of absolute magnitude values alone is insufficient for accurate motion classification due to the continuous influence of gravitational acceleration. The magnitude readings remain within the range of approximately 9–10 m/s^2 even when the system is in a resting state, which leads to false positive detections when a zero or low threshold is applied. A noticeable transition in the classification occurs at a magnitude value of around 10 m/s^2, which closely corresponds to the gravitational acceleration of approximately 9.8 m/s^2. To overcome this limitation, a delta-based magnitude approach is implemented, whereby the difference between the measured magnitude and the resting reference value is evaluated. Based on experimental observations, a delta magnitude threshold of 0.5 m/s^2 is selected, as it effectively differentiates between resting conditions affected by minor noise and genuine movement events. When the delta magnitude exceeds this threshold, the system is classified as active; otherwise, it is in a resting state. This threshold selection significantly reduces false detections and enhances the overall reliability of the movement monitoring system.

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 328-338

Table 4 *Movement Magnitude Analysis*

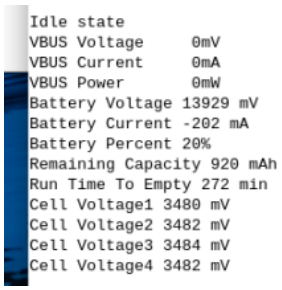
Magnitude	Active	Resting
0	Active	Active
1	Active	Active
2	Active	Active
3	Active	Active
4	Active	Active
5	Active	Active
6	Active	Active
7	Active	Active
8	Active	Active
9	Active	Active
10	Active	Resting
11	Resting	Resting
12	Resting	Resting
13	Resting	Resting
14	Resting	Resting
15	Resting	Resting

3.3 Battery Monitoring Mechanism

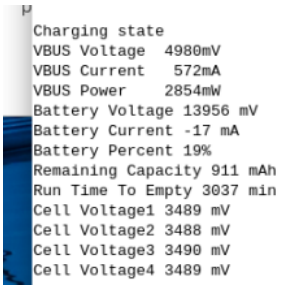
Figure 7(a) shows the battery condition during the idle state when the device is not connected to a power source. The VBUS voltage and current are 0, which means no external power is supplied. The battery is discharging, as shown by the negative battery current. The battery level is about 20%, with a remaining capacity of 920 mAh. All four cell voltages are balanced at around 3.48 V, indicating the battery cells are in stable condition.

Figure 7(b) shows the battery condition during the charging state; the device is connected to a power source. The VBUS voltage is about 5 V, and current is flowing into the device, showing that charging is active. The battery current becomes very small, indicating the battery is slowly charging. The battery level remains around 19-20%, and the cell voltages slightly increase to about 3.49 V, showing that energy is being added to the battery.

Overall, the results show that the battery operates normally, with balanced cell voltages and clear differences between idle and charging conditions.



(a)



(b)

Fig. 7 *(a) Result battery in idle state (b) Result battery in charging state*

3.4 Implementation of System

3.4.1 Zone Monitoring Alert

Table 5 shows the alert behavior of the zone monitoring system implemented in this project. Three zone conditions are defined: Safe, Alert, and Danger. When the tracked animal remains within the Safe Zone, no alert is triggered, indicating that the animal is still inside the designated protected boundary.

When the animal enters the Alert Zone, the system activates the alert mechanism and sends a notification via Telegram to inform the user that the animal is approaching the geofence boundary. This early warning allows conservation personnel to take preventive action.

If the animal moves into the Danger Zone, which is outside the predefined geofence, the system also triggers an alert. This condition represents a high-risk situation, and immediate notification is sent to ensure a rapid response.

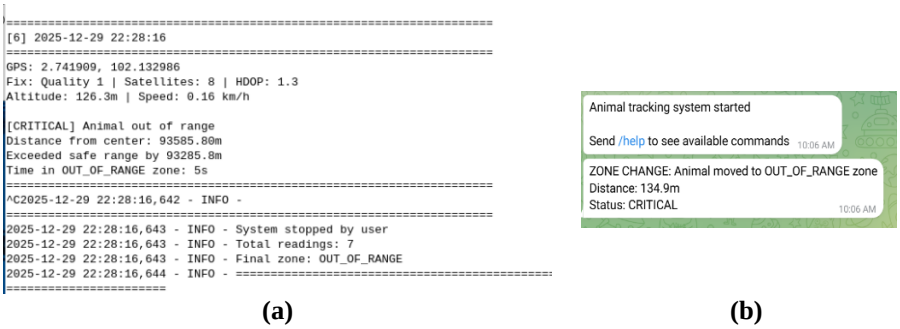
Table 5 *Zone-Based Alert Trigger Conditions*

Zone	Alert Trigger
Safe	No
Alert	Yes
Danger	Yes

Figure 8 shows the zone monitoring alert displayed in the Raspberry Pi terminal. The system records the date and time along with GPS location of the animal. The GPS signal quality is good, with 8 satellites connected and a low Horizontal Dilution of Precision (HDOP) value, which indicates accurate positioning. The speed is very low, showing that the animal is almost stationary.

A critical alert is triggered because the animal has moved outside the safe zone. The distance from the center is 93,585.8 meters, which exceeds the allowed safe range. The system also shows how much the safe distance has been exceeded and how long the animal has remained in the out-of-range zone.

Finally, the system logs confirm that monitoring has ended, the total number of readings taken, and that the system can successfully detect and report dangerous zone violations in real time.



(a) **(b)**

Fig. 8 *(a) Zone Monitoring alert output in the Raspberry Pi terminal (b) Zone Monitoring alert notification displayed via Telegram*

3.4.2 Battery Status Alert

Figure 9(a) shows the battery alert message displayed in the Raspberry Pi terminal when the battery level drops to a low condition. In this situation, the battery percentage is recorded at 15%, which is below the predefined threshold of 25%. Once this condition is detected, the system automatically triggers a low battery warning. The terminal output displays important battery information such as the battery percentage, voltage value, and current operating status. A warning message is clearly shown to inform the user that the battery level is low and that recharging is required. At the same time, the MPU6050 sensor status is also displayed, indicating that the device is in resting condition and that the alert is generated specifically due to the battery level.

Figure 9(b) shows the battery alert notification received through Telegram. After the low battery condition is detected by the system, a notification is sent to the user via Telegram in real time. The message informs the user that the battery level has fallen below 25% and requests the user to connect the charger. This notification allows the user to receive the alert remotely without needing to continuously monitor the Raspberry Pi terminal. By using Telegram as the notification platform, the system ensures that battery-related issues can be addressed promptly, helping to prevent unexpected system shutdown

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 328-338

(a) **(b)**

Fig. 9 *(a) Battery alert output in the Raspberry Pi terminal (b) Battery alert notification displayed via Telegram*

3.5 Response time for alert and notification

Table 6 presents the response time analysis for alert and notification delivery based on ten experimental observations. The event trigger time represents the moment at which an event was detected by the system, while the notification received indicates when the corresponding alert was received by the user via Telegram. The response time is calculated as the time difference between these two timestamps. The recorded response times range from 1.23 s to 2.03 s, indicating that all notifications were delivered within a short time interval. This narrow range of values demonstrates consistent system performance with minimal variation across repeated tests. Overall, the results confirm that the implemented notification mechanism can provide timely and reliable alerts, making it suitable for real-time monitoring applications.

Table 6 *Notification Timing Received via Telegram*

Test No.	Event Trigger Time (hh:mm:ss)	Notification Received Time (hh:mm:ss)	Response Time (s)
1	19:28:34	19:28:35.51	1.51
2	19:28:36	19:28:38.03	2.03
3	19:28:38	19:28:39.45	1.45
4	19:28:40	19:28:41.36	1.36
5	19:28:42	19:28:43.23	1.23
6	19:28:44	19:28:46.01	2.01
7	19:28:46	19:28:47.50	1.50
8	19:28:48	19:28:49.52	1.52
9	19:28:50	19:28:51.57	1.57
10	19:28:52	19:28:53.65	1.65

3.6 Conclusion

The primary aim of this project was to develop a real-time animal tracking system using Raspberry Pi, GPS module, MPU6050 sensor, and UPS HAT E to enhance wildlife monitoring and protection. This aim was successfully achieved through the integration of the Raspberry PI with the GPS module, which accurately captured real-time location data, including latitude and longitude. In addition, the MPU6050 sensor enabled motion monitoring, while the UPS HAT E ensured continuous system operation through reliable power management. The system effectively performed zone monitoring based on predefined geofencing rules and generated alerts when zone boundaries were crossed, demonstrating its suitability for practical wildlife tracking applications.

Acknowledgement

This project paper is supported by the Universiti Tun Hussein Onn Malaysia. The author would like to thank the Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia for providing the necessary research facility for this study.

References

[1] F. Rovero, F. Zimmermann, D. Berzi, and P. Meek, "'Which camera trap type and how many do I need?' A review of camera features and study designs for a range of wildlife research applications," *Hystrix*, vol. 24, no. 2, 2020, doi: 10.4404/hystrix-24.2-6316.

[2] R. Tarapure, "Beast Sentinel: Smart IoT-Based Wildlife Tracking and Intrusion System," *Int J Res Appl Sci Eng Technol*, vol. 13, no. 8, pp. 982–986, Aug. 2025, doi: 10.22214/ijraset.2025.73691.

[3] S. A. Birari, N. R. Wankhade, and S. B. Bagal, "IoT Based Anti-Poaching System for Trees and Wildlife Monitoring System in Remote Area," 2024. [Online]. Available: https://www.jisem-journal.com/

[4] H. J. Williams *et al.*, "Optimizing the use of biologgers for movement ecology research," Jan. 01, 2020, *Blackwell Publishing Ltd*. doi: 10.1111/1365-2656.13094.

[5] S. Das, S. Nandy, S. Chakraborty, J. Singh, and T. Halder, "Implementation of Complimentary filter on MPU 6050".

[6] M. N. Osman, M. H. F. Ismail, K. A. Sedek, N. A. Othman, and M. Maghribi, "Low-Cost Home Security Notification System Using IoT and Telegram Bot: A Design and Implementation," *Journal of Computing Research and Innovation*, vol. 7, no. 2, pp. 327–337, Sep. 2022, doi: 10.24191/jcrinn.v7i2.325.