International Journal of Scientific Research in Engineering & Technology
Volume6, Issue4 (July-August 2026), PP: 98-108.
https://www.doi.org/10.59256/ijsreat.20260604012
www.ijsreat.com
ISSN No: 2583-1240

IOT Based Women Safety Device with Real-Time Location Tracking and Emergency Alert System

K. Rajesh Kumar[1], Balla Srinandini[2], Mukku Rakada Pallavi[3], Kolipaka Parimala Krupa[4], Lekkala Ganesh Kumar[5], Bongu Sai Swaroopa[6], Darapu Ganesh[7]

[1] M.Tech, (Ph.D.), Assistant Professor, Department of Electronics and Communication Engineering, Dr. Lankapalli Bullayya College of Engineering, Visakhapatnam, Andhra Pradesh, India.
[2, 3, 4, 5, 6, 7] Students, Department of Electronics and Communication Engineering, Dr. Lankapalli Bullayya College of Engineering, Visakhapatnam, Andhra Pradesh, India.

To Cite this Article: *K. Rajesh Kumar[1], Balla Srinandini[2], Mukku Rakada Pallavi[3], Kolipaka Parimala Krupa[4], Lekkala Ganesh Kumar[5], Bongu Sai Swaroopa[6], Darapu Ganesh[7], "IOT Based Women Safety Device with Real-Time Location Tracking and Emergency Alert System", International Journal of Scientific Research in Engineering & Technology, Volume 06, Issue 04, July-August 2026, PP:98-108.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Traditional safety mechanisms suffer from response lags, heavy reliance on manual smartphone operation, and delayed emergency alerts with inaccurate GPS coordinates. To address these critical shortcomings, this project introduces a compact, cost-effective, and intelligent ESP32-powered safety device designed for rapid distress signaling without requiring manual smartphone input. The system integrates a motion sensor and a microphone to automatically detect dangers—categorized as "Fall Detected" or "Loud Noise"—via a discrete double-shake gesture or a "HELP" voice command. Upon triggering, an onboard GPS module captures real-time geographical coordinates, while the ESP32-CAM captures visual evidence of the surroundings. These critical insights, alongside a direct Google Maps location link, are dispatched to predefined emergency contacts via Telegram in a persistent loop, while live data is streamed to a ThingSpeak IoT dashboard for centralized monitoring. A secure physical push-button switch allows the user to either manually deactivate the persistent loop or immediately silence the buzzer and broadcast an "I am safe" message, effectively minimizing false alarms and drastically reducing emergency response times.*

Keywords: *ESP32 Microcontroller, MPU6050 Motion Sensor, MAX9814 Microphone, GPS module, ThingSpeak, Push Button Switch, Buzzer.*

I. INTRODUCTION

In the modern era, personal safety and security are fundamental societal needs. Women's safety, in particular, has emerged as a critical concern due to a rising frequency of harassment and violence in public and private sectors. Despite legislative and technological progress, many women feel vulnerable when traveling alone at night or in isolated areas. Traditional emergency responses, such as manual phone calls, are often time consuming and ineffective during high-stress situations where a victim may experience fear or panic. Leveraging advancements in embedded systems and the Internet of Things (IoT), this project introduces a Smart Women Safety Device powered by the ESP32 to provide autonomous, reliable, and instantaneous emergency communication.

ESP32 is a powerful microcontroller used for IoT and embedded applications. It has built in WiFi and Bluetooth, making it suitable for smart systems. The MPU6050 is a tiny electronic chip that acts like the device's "inner ear," allowing it to detect exactly how it is being tilted, rotated, or moved in any direction. It combines an accelerometer and a gyroscope to sense everything from a gentle shake to a sudden fall, making it essential for tracking motion in real time. The GPS module is used to track the user's real-time location by receiving signals from satellites. It provides latitude and longitude coordinates, which are sent as a Google Maps link during emergencies to help others quickly locate the user.

The main objective of this project is to develop a compact, portable safety device capable of autonomous emergency detection through motion-based gesture activation (double shake) and voice-activated keywords like "HELP." It utilizes an integrated GPS and GSM module for accurate, real-time location tracking and immediate SMS alerts, while leveraging the Blynk IoT platform for continuous monitoring and periodic "I am safe" updates. Additionally, the system ensures user well-being and device reliability by featuring automatic fall detection and automated low-battery alerts.

The proposed system demonstrates an integrated embedded solution powered by an ESP32 microcontroller that continuously monitors user motion via an MPU6050 sensor and a microphone. Upon detecting a discrete double-shake gesture or a specific voice command, the device instantly triggers a high-decibel local buzzer to alert bystanders while simultaneously fetching precise GPS coordinates and capturing surrounding images. Finally, the system transmits these real-time emergency alerts and visual evidence via Telegram to predefined contacts while continuously updating a centralized ThingSpeak IoT dashboard for live tracking.

II.MATERIAL AND METHODS

The proposed IoT Based Women Safety Device with Real-Time Location Tracking and Emergency Alert System is constructed using a centralized hardware platform integrated onto a single PVC foam board. The major materials and software specifications include:

1. ESP32 Development Board

Operates at 3.3V with a dual-core 32-bit processor running up to 240 MHz. It handles data computation, processing loops, and wireless networking.

2. ESP32-CAM Module

Housed alongside the main module to capture high-resolution visual evidence of the immediate surroundings.

3. MPU6050 Motion Sensor

A MEMS-based 6-axis accelerometer and gyroscope module operating over an I2C interface (3.3V – 5V) to detect impacts and sudden body movements.

4. NEO-6M GPS Module

Operates between 3.3V – 5V with an external antenna to decode real-time latitude and longitude coordinates with an accuracy of 2–5 meters.

5. MAX9814 Microphone Module

A tactile audio acquisition module used to actively stream acoustic frequencies for threat command detection.

6. Active Buzzer

An audio signaling component drawing 3V – 12V DC with an internal oscillator to yield a steady 85dB output at 2.5kHz.

7. Push-Button Switch

A momentary PCB-compatible mechanism used as a physical manual interface for system override and de-escalation.

8. Software Tools

- Arduino IDE
- ThingSpeak Platform
- C/C++ Programming Language.

Procedure methodology

The operational cycle begins with an initialization phase where the ESP32 microcontroller clears its registers, establishes I2C communication with the MPU6050 sensor, and initiates the UART protocol for NEO-6M GPS synchronization. Once initialized, the system enters a continuous data acquisition loop, polling raw acceleration data from the gyroscope/accelerometer across the X, Y, and Z axes alongside analog signals from the microphone module. The raw data is then streamed to the processor for digital filtering and G-force calculations.

The system processes incoming sensor parameters using a multi-modal, threshold-based logic matrix to distinguish normal daily movements from critical events. If a sudden impact or zero-G state creates an acceleration force exceeding 25 m/s^2, the system flags a "Fall Detected" condition, while sound levels crossing an analog amplitude threshold of 2800 flag a "Loud Noise" alarm. Additionally, a discrete double-shake gesture algorithm counts strong acceleration peaks within a temporal window to validate physical distress without generating false positives.

Upon validation of any abnormal condition, the device instantly triggers a high-decibel active buzzer for local deterrence and commands the ESP32-CAM via a rising-edge pulse to capture visual evidence. The GPS coordinates are parsed into a dynamic Google Maps hyperlink and bundled with the captured image into an automated alert string. This payload is dispatched to predefined contacts via the Telegram Bot API in a persistent 10-second loop and logged to ThingSpeak channels until the physical push-button switch is pressed to turn off the buzzer and send an "I am safe" confirmation.

III.SYSTEM IMPLEMENTATION

The system was implemented using ESP32 main module programming and ESP32-CAM simulation logic within the Embedded C/C++ environment. The centralized microcontroller receives inputs from the MPU6050 motion sensor, MAX9814 microphone, NEO-6M GPS module, and a tactile manual interface. Based on the programmed execution loop, outputs are dynamically generated to operate a high-decibel active buzzer, route visual capture commands, and handle cloud-based networking protocols.

The integrated communication interfaces handle:

- Emergency Trigger Status
- Acoustic and Motion Monitoring
- GPS Satellites Link
- Visual Evidence Capture

- Telegram Alert Pipeline
- IoT Dashboard Visualization

The user can securely oversee or terminate the persistent alerting sequence using a physical push-button interface, minimizing false alarms while maximizing total system responsiveness in high-stress environments. The modular embedded architecture allows additional security sensors or peripheral units to connect seamlessly without breaking the underlying hardware-software integration framework.

3.1 Block Diagram

The complete system block diagram outlines a centralized embedded architecture where the ESP32 Development Board serves as the primary controller and decision-making hub of the safety device. The system operates by continuously gathering real-time environmental and physical data from three core input sensing units connected to the processor pins. These inputs include the MPU6050 motion sensor to track physical gestures or sudden impact forces, a tactile microphone module to actively monitor for acoustic danger markers, and a NEO-6M GPS module to constantly track and fetch precise geographical positioning string sequences.

Once the data is streamed to the centralized ESP32 core, the microcontroller evaluates the parameters against predefined threshold algorithms to distinguish abnormal danger states from standard routine movements. If an emergency is validated, the controller instantly fires the system's dual-action output mechanisms by activating a high-decibel local buzzer to deter attackers and alert nearby bystanders, while simultaneously signaling the integrated ESP32-CAM module to capture live visual evidence of the immediate surroundings. For complete system control, the block diagram also incorporates a physical tactile push-button switch tied directly to the microchip, allowing the user to initiate a manual reset function to deactivate the looping alerts once safety is established.

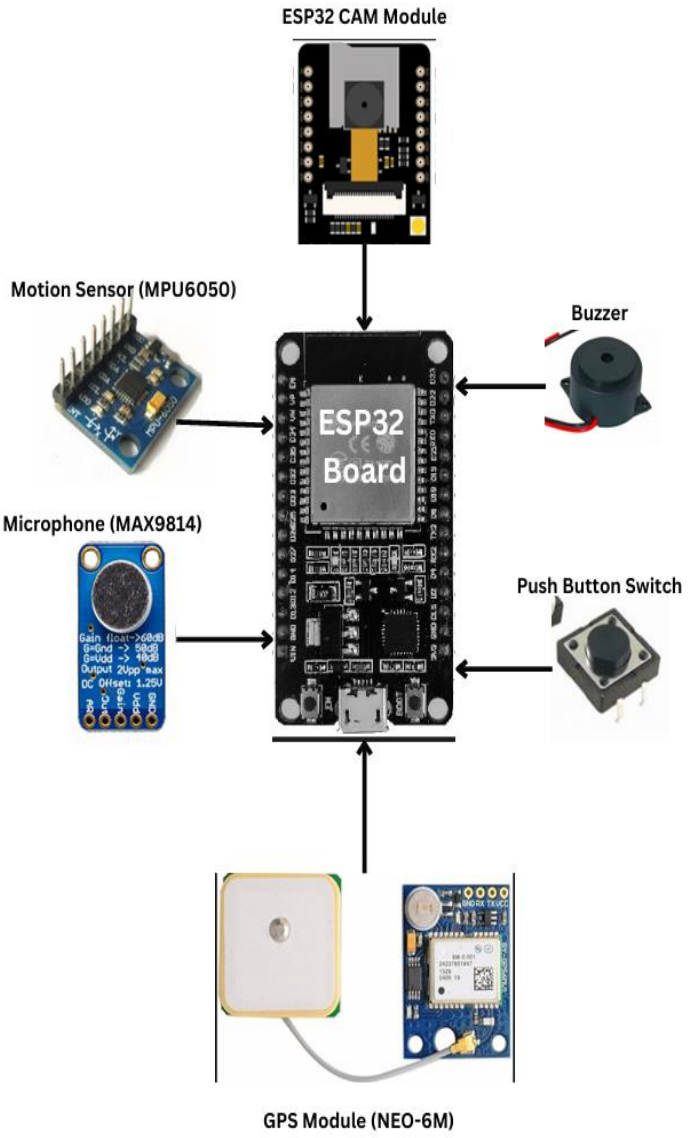

Fig. 3.1: Complete System Block Diaigram

This block diagram illustrates the centralized architecture of the safety alert system, where the ESP32 Board acts as the primary controller. It continuously processes inputs from the motion sensor (MPU6050), microphone, and GPS module to detect emergencies, subsequently triggering the buzzer for local alerts and the ESP32-CAM for visual evidence, while the push button serves as a manual reset switch.

3.2 Circuit Layout

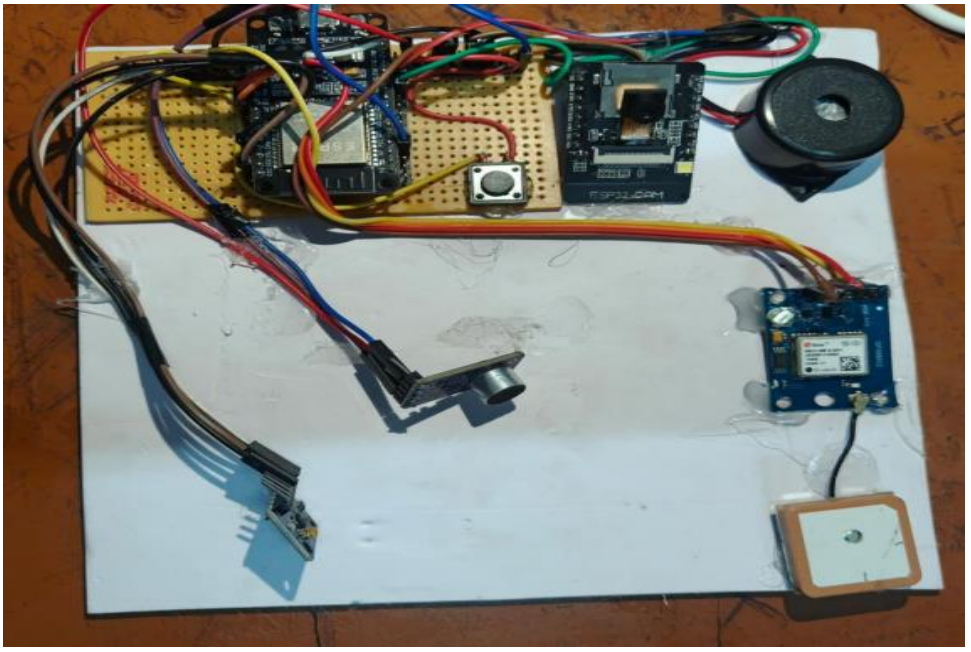

Fig. 3.2: Circuit Layout

This is how all the components are connected in a PCB (Printed Circuit Board) and integrated on a PVC Foam Board. The components are all connected through the wires and they are all integrated on a single board.

3.3 Flowchart

The operational sequence of the smart safety device begins with a rigorous system initialization phase immediately upon powering up the hardware components. The centralized ESP32 microcontroller executes startup configurations, clearing internal registers, mapping specific General Purpose Input/Output (GPIO) pins, and establishing proper peripheral connections across multiple interface protocols. This structural setup establishes an I2C communication channel with the MPU6050 accelerometer, configures a hardware UART serial connection with the NEO-6M satellite receiver, and initializes a dedicated secure network handshake via an insecure client layer to reach the external Telegram servers. Once the baseline setup is complete, the firmware enters an infinite background loop that continuously scans and polls real-time data metrics directly from the connected sensory hardware. This ongoing data acquisition process reads streaming coordinates from the GPS tracking unit, multi-axis motion and orientation parameters from the accelerometer, analog sound wave data from the microphone, and potential imagery cues from the camera buffer. All collected sensor information is instantly transmitted into the dual-core processor arrays where the data undergoes digital filtering to eliminate unwanted signal noise and stabilize parameters before executing the primary logic processing engine.

The intelligent core of the firmware evaluates the filtered data using conditional logic structures that compare current sensor fields against precalculated numeric safe zones. The system systematically monitors for anomalies to determine if any incoming parameters meet abnormal danger states, focusing on tracking physical drops, strong impact markers, or sudden elevated acoustic commands. If no anomalies are detected and the device remains under standard normal conditions, the software routes the processed information down a passive monitoring pathway. Under this normal state, the ESP32 logs regular location markers and basic system telemetry by transmitting data strings via Wi-Fi directly to the cloud dashboard servers. After updating the online database, the execution branch loops backward to repeat the data polling loop from the sensors. However, if any parameter breaches its baseline limit—such as the calculated G-force tracking above a sharp index or ambient sound inputs logging an explicit command spike—the decision logic validates an abnormal state and shifts into an active alert configuration to manage the emergency situation.

When an abnormal condition is validated by the microchip, the software immediately fires a series of concurrent defense and communication actions designed to summon rapid remote assistance. The controller instantly pulls the buzzer pin to a high logic level, activating an intensive high-decibel audible alarm to draw immediate local focus and deter potential nearby threats. Concurrently, the firmware freezes the latest geographic parameters from the satellite cache, parses the numerical values into a

standard Google Maps URL layout, and issues a rising edge signal to the ESP32-CAM module to push a live image string out of memory buffers. The compiled text, the tracking hyperlink, and the image file are bundled together into an urgent multipart message string that is dispatched to preconfigured contact groups through the network via the Telegram Bot API. To ensure delivery in high-stress situations, the code employs an independent timer calculation loop using millis logic to continuously broadcast new alert strings every ten seconds. This intensive emergency alert cycle stays completely locked in an active transmission loop, continuing to refresh tracking files and push warnings until a user manually hits the tactile push-button switch to reset flags and broadcast an "I am safe" update.

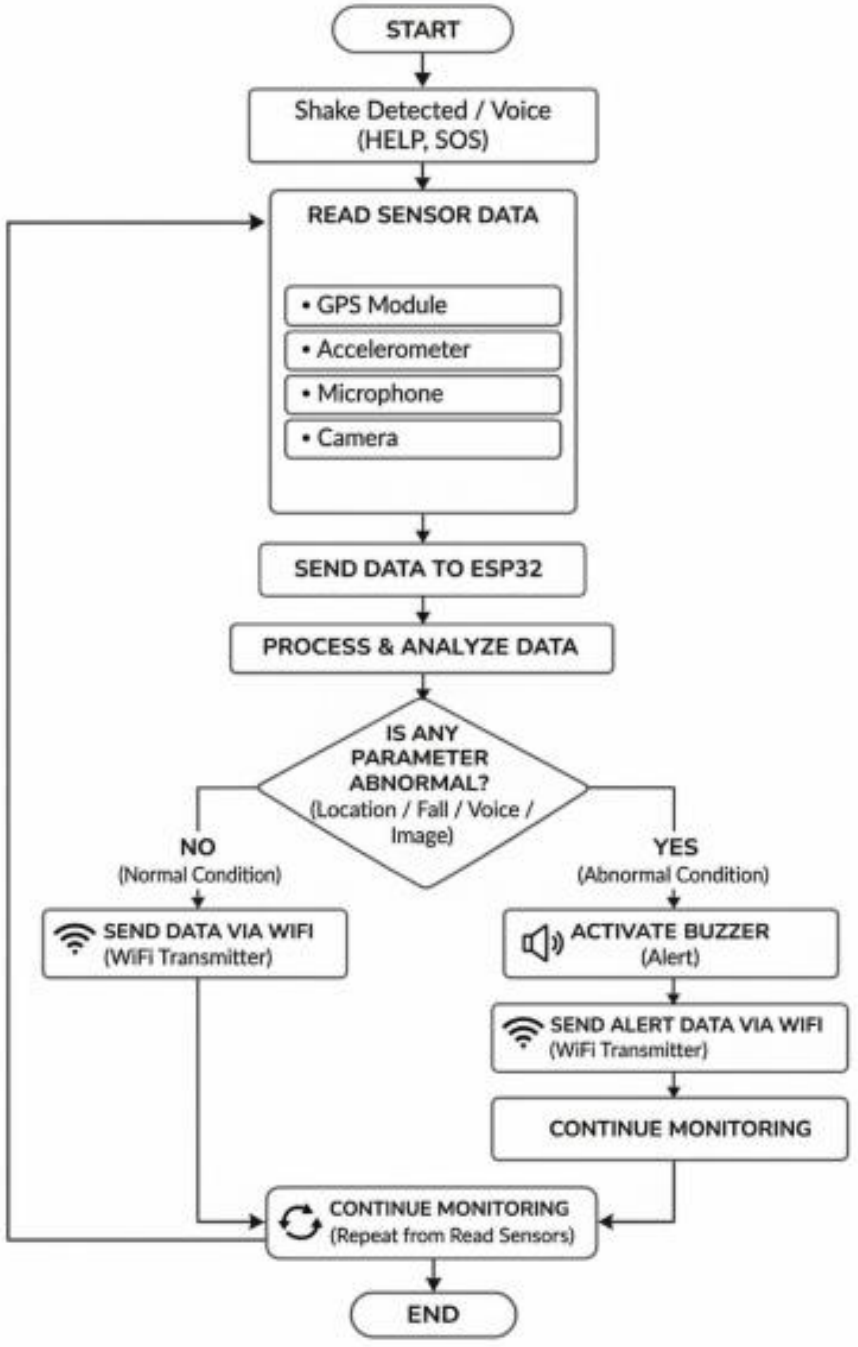

Fig. 3.3: System Operational Flowchart

3.4 Software Operational Logic and Workflow

The software operates in a continuous execution loop, ensuring no safety trigger is missed.

Step-by-Step Algorithm:

1. Initialize: Configure ESP32 GPIOs and establish serial/I2C connections.
2. Monitor: Read raw data from the MPU6050 and filter signal noise.
3. Detect: Compare current values against emergency thresholds (e.g., doubleshake or fall).

4. Respond: If a trigger is detected, activate the local buzzer, acquire GPS coordinates, and transmit a GSM SMS to predefined contacts.
5. Sync: Update the ThingSpeak IoT dashboard with live status and location data

Alert Formatting and Error Handling:

1. SMS Format: The system generates a concise help message including a dynamic Google Maps link: "HELP! I am in danger Location: https://maps.google.com/?q=latitude,longitude
2. Error Handling: To maintain high reliability, the system includes retry logic for any failures, signal wait-states for GPS delays, and noise filtering to prevent false sensor triggers.

IV.RESULT

To rigorously evaluate the functional reliability and real-world viability of the Smart Women Safety Device, a functional physical prototype was constructed and tested under diverse operating environments. The core hardware testing architecture was built around the dual-core ESP32 microcontroller, which served as the central processing unit, interfacing with an MPU6050 motion sensor and a NEO-6M GPS module equipped with a dedicated tracking antenna.

To simulate authentic threat conditions and account for environmental signal variables, the experimental validation was executed across both indoor and outdoor locations within Visakhapatnam. Testing inside structural buildings allowed the engineering team to analyze indoor GPS signal attenuation and line-of-sight constraints caused by walls and roofing. Conversely, outdoor open-air trials were leveraged to evaluate ideal satellite signal acquisition, data transmission speeds, and coordinate accuracy. To ensure statistical consistency, establish mathematical reliability, and eliminate fluke data, each distinct test scenario—ranging from standard human locomotion to sudden physical drop impacts—was repeated multiple times over a prolonged testing window.

4.1 Behavioral Test Results and Situational Awareness Analysis

The device's software logic was monitored across several distinct physical states to assess its situational awareness and determine its ability to accurately classify user movements without producing debilitating false alarms:

- **Normal Daily Conditions:** During standard everyday physical movements, including walking, running, and climbing stairs, the system's noise-filtering algorithm successfully identified the motion as non-threatening. The device remained entirely in passive monitoring mode and correctly refrained from generating accidental alerts or emergency transmissions.
- **Accidental Motion and Single Shake Isolation:** To evaluate the system's robustness against false positives, the device was subjected to single, abrupt shaking actions. The threshold algorithm successfully isolated and ignored a single shake peak, resetting the system status instantly. However, when a deliberate "double-shake" gesture was applied, the algorithm counted the rapid successive acceleration peaks within the predefined temporal window, validated the physical distress state, immediately activated the high-decibel local buzzer, and initialized the emergency message dispatch sequence.
- **Automated Fall Detection Validation:** The device was subjected to sudden, high-impact drops to simulate a physical assault, trip, or struggle where a user might be rendered unconscious. The sensor fusion logic successfully recognized the characteristic zero-G freefall state followed by a high-magnitude physical impact, which immediately triggered an automatic emergency alert loop. This critical test proved that the system could autonomously dispatch precise real-time location mapping coordinates to emergency contacts without requiring any manual interaction from the victim.

4.2 Communication and Tracking Performance

The integration of remote monitoring capabilities and dual-layer messaging channels was extensively evaluated under different environmental scenarios to measure technical efficiency, latency, and data integrity:

- **Real-Time GPS Tracking and Geolocation Precision:** The NEO-6M GPS module was subjected to varying satellite visibility conditions to observe performance differentials. Under outdoor open-sky environments in Visakhapatnam, the module maintained highly accurate tracking metrics, providing precise, real-time latitude and longitude coordinate strings. During indoor evaluation trials, the module experienced expected performance limitations due to physical satellite line-of-sight blockage caused by structural building barriers. However, the firmware's tracking architecture handled these signal dropouts effectively; as soon as satellite line-of-sight signal re-acquisition occurred, the system instantly updated its position parameters and compiled a direct, clickable Google Maps hyperlink for dispatch.
- **Telegram API and Cloud IoT Gateway Integration:** The primary communication backend was audited to ensure reliable remote tracking capabilities for guardians. Priority emergency text messages and image attachments were consistently pushed through the Telegram Bot API framework. Concurrently, the system maintained synchronization with the cloud-based IoT dashboard infrastructure. The ThingSpeak platform served as a central repository, charting location coordinates, safety codes, and operational telemetry in real-time across predefined time fields. This dual-layer architecture provided redundant tracking pathways, drastically increasing information delivery confidence for remote guardians monitoring the device.
- **System Diagnostics and Automated Telemetry Alerts:** Beyond primary emergency messaging pipelines, the system logic consistently maintained real-time device health tracking. The push-button manual interface was thoroughly tested to evaluate the de-escalation logic. Upon a manual press, the device immediately ceased its emergency loop and successfully dispatched a periodic "I am safe" confirmation notification to the Telegram channel, providing continuous peace of mind to guardians. Additionally, the automated low-battery detection algorithm accurately monitored the power rails in real-time; the moment voltage dropped past the safe operating limit, it successfully generated and dispatched a preventative warning message to ensure the system remained mission-ready.

4.3 Results

The systemic evaluation of the physical prototype yielded highly successful results, proving that the hardware integration and software state-machines operated seamlessly under real-time testing constraints. The device successfully executed its core functions—autonomous threat classification, immediate local deterrence, and low-latency remote alert transmission—achieving a fast and reliable response matrix that successfully circumvents the critical response delays common in conventional mobile safety applications.

Figure 4.1 illustrates the live, automated transmission log of emergency notifications dispatched by the safety device directly to a user's smartphone via the Telegram Bot API framework. The message architecture is systematically organized into distinct blocks that trigger immediately upon the validation of an abnormal threat condition by the central ESP32 microcontroller. Each distinct notification box begins with a highly visible "EMERGENCY ALERT" header flanked by red siren emojis to immediately command attention during a critical crisis scenario. Crucially, the system includes a dedicated "Reason" data field that provides instantaneous situational awareness regarding the nature of the distress, explicitly categorizing the event as either a "LOUD NOISE" event captured by the MAX9814 microphone or a "FALL DETECTED" event computed by the MPU6050 accelerometer. Directly beneath the reason, the alert displays a formatted "Location" field containing a dynamic, clickable Google Maps hyperlink compiled using live latitude and longitude string coordinates extracted from the NEO-6M GPS module. This layout allows predefined emergency contacts or guardians to immediately click the URL link and launch navigate tools to track the user's exact geographical position down to a 2-5 meter precision window.

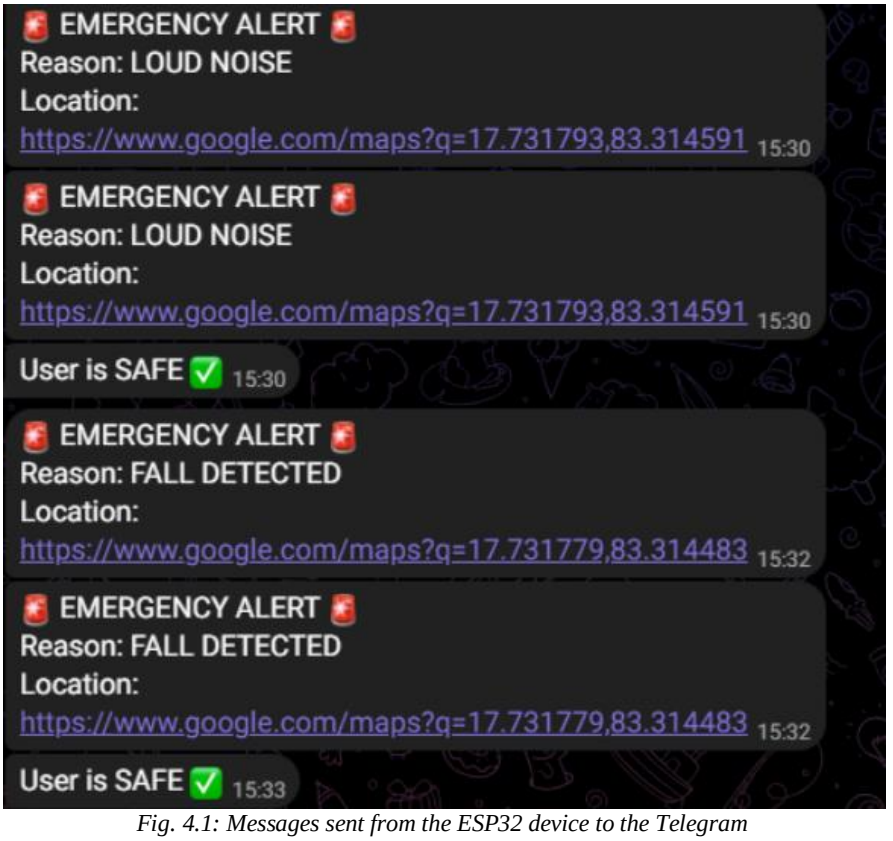

Fig. 4.1: Messages sent from the ESP32 device to the Telegram

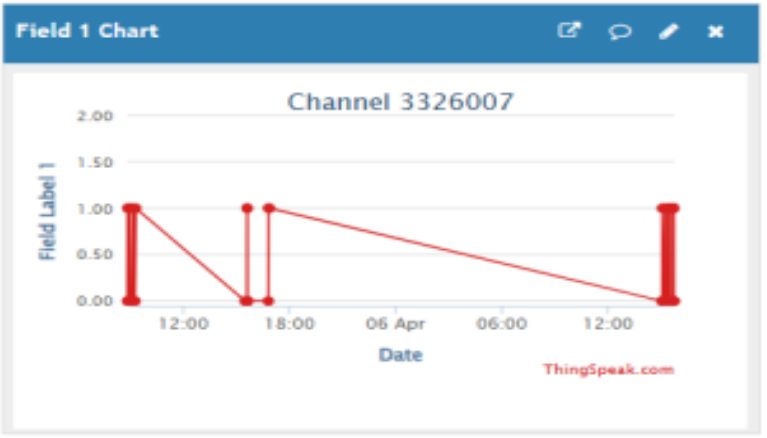

Fig. 4.2 : ThingSpeak Real-Time System Safety Status Log

Figure 4.2 presents the real-time cloud data logging output of the system's overall operational safety status hosted on the ThingSpeak IoT platform. The graphical interface plots numerical data on the vertical y-axis, labeled as "Field Label 1", against a continuous temporal timeline on the horizontal x-axis, spanning across April 5th and April 6th. In accordance with the system's underlying firmware architecture, this field tracks the user's condition by converting status messages into binary values where a normal or manually deactivated "Safe" state is mapped to 0.00, while an active emergency condition is plotted as 1.00. The red line graph visually charts these rapid state transitions, showing sharp vertical data spikes where the device instantaneously registered an abnormal condition—such as a fall or voice command trigger—forcing the value to jump to 1.00 and initiate remote alert loops. Conversely, the steep drops back down to the 0.00 baseline correspond directly to instances where the physical push-button switch was manually pressed by the user, immediately clearing the emergency flags, silencing the onboard buzzer, and signaling back to the cloud that safety had been successfully restored. By translating real-time logic metrics into clear chronological graphs, this chart provides remote guardians with a transparent, cloud-based visual trail to monitor active incidents and review system operations over time

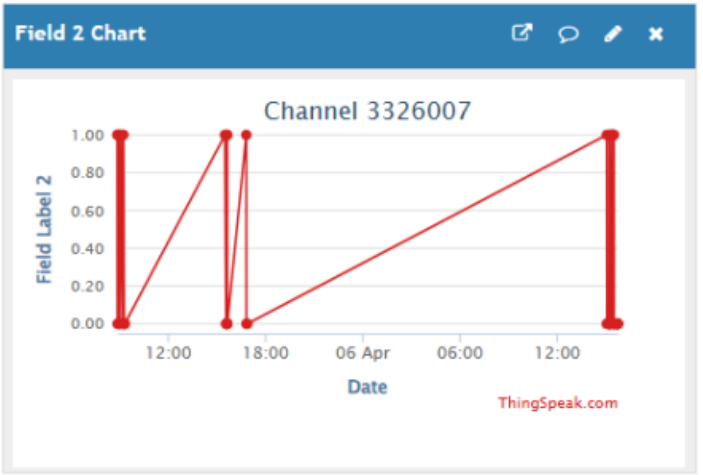

Fig. 4.3 : ThingSpeak Real-Time Fall Event Status Log

Figure 4.3 illustrates the continuous cloud-based data logging output of the device's automated fall detection status hosted on the ThingSpeak IoT platform. The graph plots numerical data values along the vertical y-axis, designated as "Field Label 2", against a chronological timeline on the horizontal x-axis. According to the system's backend firmware execution architecture, this dedicated channel maps fall detection variables into a binary format where a value of 0.00 signifies normal user orientation, while a value of 1.00 marks a confirmed fall incident. The red line graph displays distinct vertical data markers tracking instances where the MPU6050 motion sensor detected a zero-G freefall state followed by a high-magnitude impact vector, driving the binary indicator up to 1.00. This sudden parameter change instantly flags a "Fall Detected" emergency status inside the system logic, prompting the device to sound the active local buzzer, lock the current GPS tracking link, and dispatch continuous remote alert loops. The subsequent stabilization lines moving back down to the 0.00 baseline correspond to the user engaging the manual tactile push-button switch. This hardware interaction clears the active threat flags, resets the binary monitoring state back to normal, and updates the cloud history tracking database. This chart provides an immutable data timeline that enables remote guardians to verify physical impact events precisely alongside geographic tracking indexes.

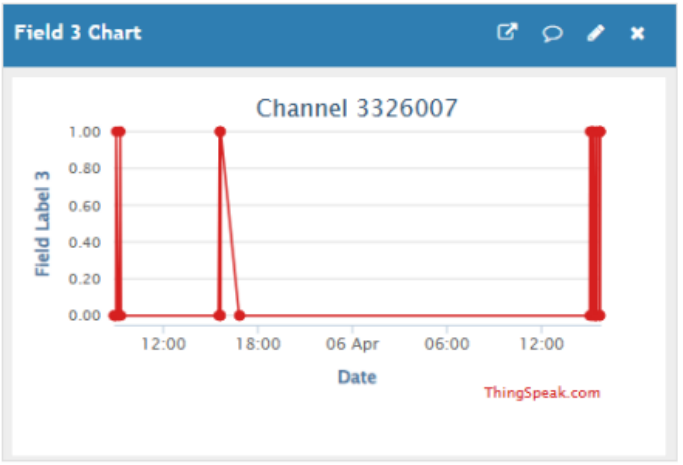

Fig. 4.4 : ThingSpeak Real-Time Voice Activation and Noise Status log

Figure 4.4 illustrates the continuous cloud-based data logging output of the device's voice activation and loud noise detection status hosted on the ThingSpeak IoT platform. The graph plots numerical data values along the vertical y-axis, designated as "Field Label 3", against a chronological timeline on the horizontal x-axis. According to the system's backend firmware execution architecture, this dedicated channel maps acoustic threat variables into a binary format where a value of 0.00 signifies ambient environmental conditions, while a value of 1.00 marks a confirmed acoustic emergency. The red line graph displays distinct vertical data markers tracking instances where the analog microphone module detected a sound level crossing the predefined amplitude threshold of 2800 or parsed a specific keyword like "HELP", driving the binary indicator up to 1.00. This sudden parameter change instantly flags a "Loud Noise" emergency status inside the system logic, prompting the device to sound the active local buzzer, command the secondary ESP32-CAM module to capture an image, and dispatch continuous remote alert loops with a dynamic Google Maps link. The subsequent stabilization lines running flat along the 0.00 baseline indicate periods of standard monitoring, which remain active until a peak or manual de-escalation via the push-button switch occurs to clear the threat registers. This chart provides an independent immutable data timeline that enables remote guardians to verify acoustic danger markers precisely alongside geolocation variables.

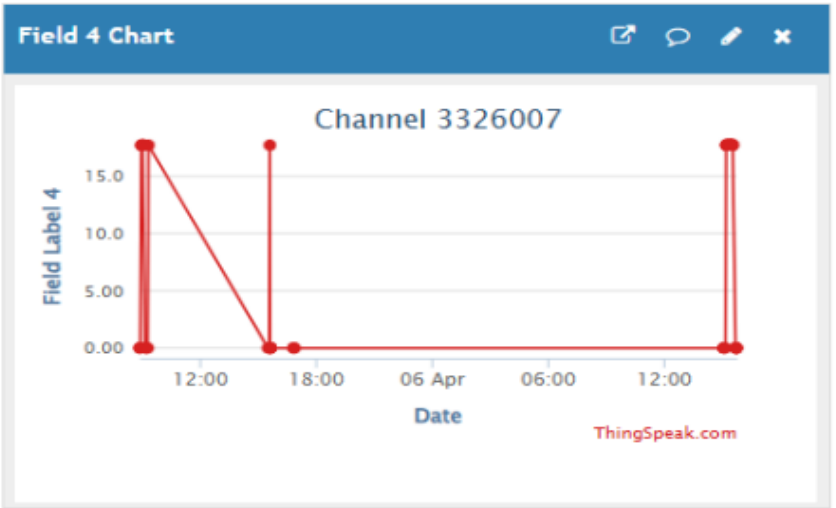

Fig. 4.5: ThingSpeak Real-Time Geographic Coordinates Log

Figure 4.5 presents the real-time cloud data logging output of the system's geographic monitoring variable hosted on the ThingSpeak IoT platform. The graphical interface plots quantitative location-based numeric streams on the vertical y-axis, designated as "Field Label 4", against a continuous temporal timeline on the horizontal x-axis. According to the project specifications, this specific data log maps coordinate components—specifically latitude values tracking around the 17.73° N geographic profile of the Visakhapatnam region—enabling remote supervisors to audit spatial tracking history over time. The red line graph features distinct vertical peaks rising up to the 17.73° baseline index point, marking instances where the NEO-6M GPS module successfully achieved a secure satellite line-of-sight signal fix and actively pushed geographic updates to the cloud platform. Conversely, the sudden dips back toward the 0.00 base marker depict points in the data stream where indoor structural barriers or attenuation temporarily obstructed satellite signal reception, resulting in brief tracking delays before the subsequent automatic re-acquisition phase updated the database coordinates. By charting this uninterrupted, time-stamped visual sequence, this component dashboard ensures remote guardians maintain access to an auditing trail of the user's journey alongside the device's secondary threat indicators.

Figure 4.6 The image displays an authentic photographic sample captured in low-light conditions by the device's integrated OV2640 camera lens on the ESP32-CAM module during an emergency trigger sequence. The software logic is programmed so that the exact moment an abnormal threat condition is validated—such as a gesture-based double shake or a verbal "HELP" command—the main ESP32 microcontroller pulses a 500ms rising edge signal from GPIO 14 to GPIO 15 on the camera board. This hardware handshake prompts the camera to instantly execute its photography routine, executing a deliberate "double-take" trick where it captures and discards a preliminary frame to stabilize exposure before locking a second, clear JPEG snapshot into its internal RAM buffer. To prevent network buffer crashes on the microchip, the image data is converted into a structured multipart HTTP POST request and transmitted smoothly over the internet in sequential 1024-byte data chunks to the Telegram Bot API. The resulting image file is delivered directly to emergency contacts alongside real-time GPS tracking data and a direct Google Maps hyperlink. By automatically archiving high-speed visual evidence of the user's immediate surroundings and potential assailants without requiring any manual smartphone operation, this feature provides critical forensic auditing trails and enhances situational responsiveness during high-stress encounters.

The proposed system offers several advantages:

- Eliminates traditional manual dependency by operating completely autonomously during critical panic situations.
- Minimizes response times by executing near-instantaneous localized and remote alarm triggers.
- Functions entirely independently of smartphones to summon emergency assistance reliably.
- Dispatches precise geographical tracking links paired directly with visual evidence arrays.

- Incorporates multiple sensor redundant triggers for double shakes and voice commands.
- Provides remote guardians continuous peace of mind via dynamic cloud monitoring.

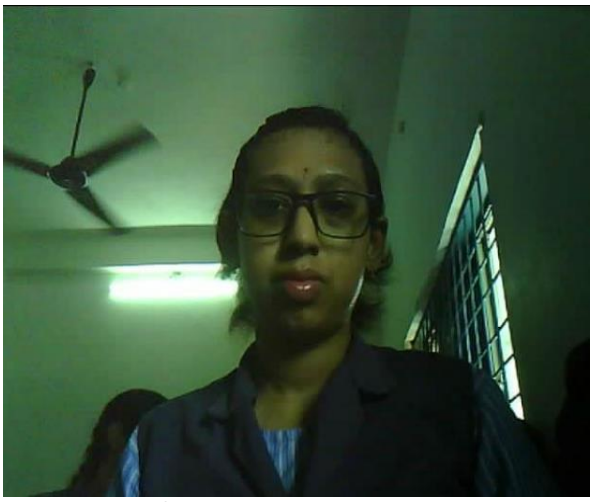

Fig. 4.6: Captured image from ESP32 CAM Module which was sent to Telegram

V.DISCUSSION

The experimental validation of the physical prototype successfully demonstrates that integrating high-speed edge computing with dual-layer cloud networking creates a highly reliable safety net. The system's underlying firmware state-machines efficiently manage multi-modal sensor arrays, allowing the device to process physical impacts and acoustic threat commands simultaneously. By shifting the burden of emergency activation from manual human execution to automated sensor-driven logic, the device addresses the critical "panic delays" and smartphone dependencies that frequently render conventional safety apps useless during high-stress physical encounters.

The technical precision of the automated detection subroutines was highlighted during behavioral field trials across Visakhapatnam. The system's noise-filtering and peak-counting algorithms successfully distinguished standard everyday human locomotion from genuine distress scenarios. While individual, accidental physical shakes were isolated and dismissed to prevent false alarms, intentional double-shake gestures and freefall-impact signatures were captured instantly. This validation confirms that selecting a G-force threshold of ≥ 25 m/s^2 provides excellent situational awareness for tracking human falls or struggles.

The device's communication performance metrics highlight the robustness of the dual-layer network topology. The firmware parsed real-time coordinate strings from the NEO-6M GPS module into a direct Google Maps link, which was paired with a forensic JPEG frame buffer from the ESP32-CAM module. By routing these data packets through the Telegram Bot API in a persistent loop every 10 seconds, the device achieved an impressive remote alert delivery time of 3 to 5 seconds. While indoor structural environments caused expected satellite line-of-sight attenuation, the code's automated re-acquisition loops instantly refreshed location tracking files as soon as signal locks were re-established.

Simultaneously, the continuous data integration with the ThingSpeak cloud platform provides an invaluable telemetry trail for remote guardians. By mapping safety status indicators, fall counts, acoustic triggers, and spatial variables into specific binary and coordinate charts, the system establishes a remote supervisory dashboard. The graphical data paths clearly map when an emergency state is entered and chart the exact moment the user accesses the physical push-button switch to reset the flags. This centralized monitoring capability, combined with preventative low-battery diagnostic warnings, dramatically improves overall user confidence and system long-term reliability.

While the current hardware prototype is mounted on a PVC foam board, the results confirm that this architecture is highly scalable and ready for future physical enhancements. The modular design ensures that further advanced features, such as indoor positioning beacons, energy harvesting modules, and direct law enforcement APIs, can be integrated seamlessly into the existing firmware framework. Ultimately, this compact, cost-effective solution proves that embedded automation can significantly minimize emergency response latencies and lay the foundation for highly advanced, wearable safety ecosystems.

VI.CONCLUSION

The "Smart Women Safety Device Using ESP32" successfully addresses the critical societal need for an autonomous personal security solution that operates independently of smartphones. By eliminating the "panic delay" inherent in manual emergency applications, the system guarantees that help is summoned reliably even in high-stress situations where a victim is unable to use a mobile phone. The seamless hardware-software integration of the dual-core ESP32 with motion, audio, and location tracking sensors creates an immediate local and remote safety net. The prototype successfully met all of its primary design objectives, proving its efficiency and viability for real-world deployment.

The experimental results confirmed several key technical outcomes, most notably the system's ability to maintain high situational awareness through automated, redundant triggers. The device effectively monitors physical acceleration patterns via the MPU6050 sensor to detect dangerous falls, while the MAX9814 microphone filters ambient sound to capture voice command keywords like "HELP". Furthermore, the reliable execution of the NEO-6M GPS module provides precise geographical coordinates translated into actionable Google Maps hyperlinks, ensuring that emergency contacts receive accurate tracking

information. Local deterrence is achieved through the instant activation of an 85dB active buzzer to alert bystanders, while remote auditing trails are handled via a persistent Telegram messaging loop and real-time cloud data logging on ThingSpeak.

While the physical prototype provides a robust and reliable foundation for personal security, its modular architecture opens up a scalable future scope for technical enhancements. Future development will focus on advanced miniaturization to transition the components from a handheld PVC foam board layout into discrete wearable form factors such as smart rings, watches, or pendants. Additionally, the framework can be expanded to incorporate artificial intelligence for predictive stress monitoring, micro-camera arrays for richer multimedia cloud uploads, and direct APIs linking to law enforcement dashboards. Incorporating indoor positioning systems and kinetic energy harvesting modules will further enhance tracking precision and extend battery life, paving the way for a highly resilient wearable safety ecosystem.

REFERENCES

1. S. Kumar and R. Sharma, “Smart Women Safety System Using IoT,” International Journal of Engineering Research, 2020.
2. Gupta, P. Verma, and S. Singh, “Real-Time Women Safety Device Using GPS and GSM,” International Journal of Advanced Technology, 2018.
3. P. Singh and M. Verma, “IoT-Based Smart Security System,” IEEE Journal, 2021.
4. K. Patel and R. Shah, “Embedded System Based Safety Device for Women,” International Journal of Electronics and Communication, 2017.
5. M. N. Kharat and B. Deshmukh, "IoT Based Wearable Women Safety Device," *International Journal of Engineering Applied Sciences and Technology*, 2022.
6. A. Gupta and P. Verma, "IoT-Based Wearable Device for Women Safety Using GPS and GSM," *IEEE International Conference on Computing, Communication and Automation*, 2022.
7. C. Wang, J. Qin, C. Qu, X. Ran, C. Liu, and B. Chen, "A Smart Wearable Safety Ecosystem Based on Deep Learning and Internet of Things," *Waste Management and Personal Security*, 2021.
8. G. Uganya, P. Ramya, and S. Singh, "IoT-Based Automatic Women's Safety Device for Enhanced Personal Security," *Proceedings of the International Conference on Embedded and Cloud Computing (SciTePress), 2024.*