© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

Proto-Wrist: An IoT-Based Wearable SOS Safety Wristband with ntfy.sh Push Notification for Personal Emergency Response

Aarya A Chahande[1], Arjun S Sankpal[2], Parth S Cherekar[3], Suhani Khade[4], Vaibhavi Yadav[5], Mr. M.B. Nigade[6]
[1,2,3,4,5]*Dept. of First Year Engineering Science, AISSMS IOIT, Pune*
[6]*Assistant Professor, Dept. of First Year Engineering Science, AISSMS IOIT, Pune*

***Abstract*—Personal safety remains a critical concern, particularly for women and children who face risks of harassment, assault, and abduction. Existing safety solutions such as mobile applications and GPS trackers suffer from limitations including smartphone dependency, multi-step interaction requirements, and costly hardware. This paper presents Proto-Wrist, a compact wearable safety wristband powered by an ESP32-S3 microcontroller that delivers single-button SOS emergency alerts with real-time GPS location sharing. The system leverages ntfy.sh, a lightweight, open-source HTTP-based push notification service, to instantly dispatch alerts containing a Google Maps location link to pre-registered emergency contacts' mobile devices. A companion Flutter application handles device pairing via Bluetooth and emergency contact management. Experimental results demonstrate SOS alert delivery within 5–10 seconds, GPS accuracy of 5–10 meters in open areas, and more than 12 hours of battery life under normal usage. The proposed system is low-cost (approximately ₹2,400), standalone, and suitable for real-world deployment in personal safety scenarios.**

***Index Terms*—IoT, wearable safety device, ESP32-S3, ntfy.sh, push notification, GPS tracking, SOS alert, Flutter, personal safety, women safety.**

I. INTRODUCTION

Personal safety has become a paramount concern in contemporary society. Rising incidents of harassment, assault, and abduction especially targeting women and children demand immediate, reliable, and easy-to-use emergency alert mechanisms. Traditional responses to such situations depend heavily on smartphones, requiring multiple unlock and navigation steps that are impractical in panic states.

This paper presents Proto-Wrist, an IoT-based wearable safety wristband that distils the entire emergency response workflow to a single button press. The device integrates an ESP32-S3 microcontroller with a GPS module and communicates SOS notifications via ntfy.sh, an open HTTP push-notification service, to the smartphones of pre-registered emergency contacts. A companion Flutter mobile application manages device pairing, user registration, and emergency contact configuration.

The prototype is designed to be low-cost, energy-efficient, and standalone operating without dependence on cellular voice calls while maintaining sub-10-second end-to-end alert latency. The system contributes to SDG 16 (Peace, Justice and Strong Institutions) by strengthening personal safety infrastructure through accessible technology.

II. REVIEW OF EXISTING WORK

A. IoT-Based Safety Systems

IoT-based safety systems use distributed sensors and wireless communication to monitor individuals in real time. These systems are capable of generating automatic alerts but typically require persistent internet connectivity and infrastructure support [1]. The Proto-Wrist system reduces infrastructure dependence by using ntfy.sh's lightweight HTTP API, requiring only a standard mobile data connection at the receiver's end.

B. Mobile Application-Based Solutions

Safety mobile applications such as bSafe and Kitestring allow users to share live location and trigger

SOS alerts. However, these apps require deliberate user interaction an impractical requirement in high-panic situations [2]. Proto-Wrist eliminates multi-step interaction by reducing activation to a single hardware button press.

C. GPS Tracking Devices

Standalone GPS trackers offer precise real-time location data but lack integrated alert mechanisms. They function as passive monitoring tools and require separate communication channels to notify emergency contacts [3]. Proto-Wrist unifies GPS tracking and notification delivery into a single wearable device.

D. Wearable Safety Devices

Commercial wearable safety products such as smart bands and panic buttons are portable but often expensive (frequently exceeding ₹5,000) [4]. Proto-Wrist achieves comparable or superior functionality at an approximate hardware cost of ₹2,400.

E. Comparison of Existing Systems

Table I. Comparison of Existing Safety Systems

System Type	Advantage	Limitation
Mobile Apps	Easy to use	Needs internet + interaction
GPS Devices	Accurate location	No alert mechanism
Wearables	Portable	Expensive
Proto-Wrist	One-touch + ntfy notify	Wi-Fi/data required at receiver

F. Summary

Existing solutions either lack instant alert mechanisms, are cost-prohibitive, or have no alert mechanism. Proto-Wrist addresses all three gaps through hardware simplicity, IoT connectivity, and the ntfy.sh notification pipeline.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The Proto-Wrist system comprises three interconnected layers: (1) the wearable hardware unit, (2) the companion mobile application, and (3) the ntfy.sh cloud notification relay. Fig. 1 illustrates the end-to-end information flow from button press to notification delivery.

[SOS Button Press] → [ESP32-S3] → [Connected Device with GPS Module] → [ntfy.sh HTTP API] → [ntfy App on Receiver] → [Google Maps Link]

Fig. 1. System information flow of Proto-Wrist.

B. Hardware Components

The wearable unit is built around the ESP32-S3 microcontroller, chosen for its dual-core Xtensa LX7 architecture, integrated Wi-Fi and Bluetooth, and low-power deep-sleep modes. A 400 maH-Li-Po battery powers the device. An SOS push button serves as the sole user input, and a piezoelectric buzzer provides an audible local alert.

Table II. Hardware Bill of Materials

Sr.	Component	Unit Cost (₹)	Qty
1	ESP32-S3 Microcontroller	875	1
2	Li-Po Battery	150	1
3	SOS Push Button	5	1
4	Piezoelectric Buzzer	15	1
5	Breadboard	50	1
6	Jumper Wires (M-M)	60	1 set
7	Nylon Wristband Enclosure	200	1
8	Miscellaneous	1000	
Total (Approx.)			₹2,400

C. Software Stack

The embedded firmware is written in Micro Python and runs directly on the ESP32-S3. On the mobile side, the companion application is developed using Flutter (Dart), Google's cross-platform UI framework, targeting both Android and iOS from a single codebase. Material Design 3 widgets from Flutter's standard library form the user interface.

D. Operational Workflow

The step-by-step operational algorithm is as follows:

- Device initialises and enters a low-power sleep mode, monitoring the GPIO pin connected to the SOS button.
- Upon button press, the ESP32-S3, send a "SOS" string to the connected device. The device connected extracts the current geolocation through using its built in GPS.
- Once a GPS fix is obtained (typically 5–15 seconds in open environments), the system constructs an

SOS payload containing the latitude, longitude, and a Google Maps deep link.
- The payload is transmitted via Wi-Fi to the ntfy.sh HTTP API endpoint on the user's registered topic channel.
- ntfy.sh relays the notification to all subscribed receiver devices running the ntfy mobile application.
- Simultaneously, the onboard buzzer activates at ~85 dB to attract nearby attention.
- After alert delivery is confirmed, the device returns to sleep mode to conserve battery.

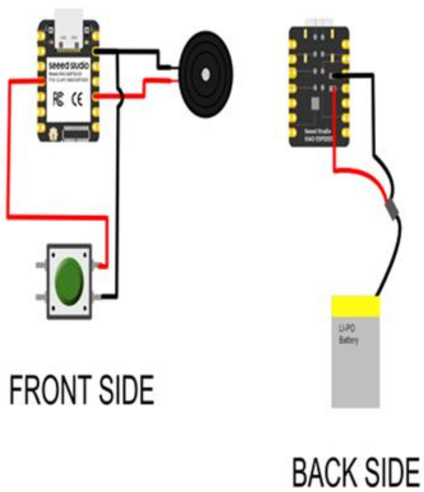

Fig. 2. Circuit Diagram.

IV. NTFY.SH: THE NOTIFICATION BACKBONE

A. What is ntfy.sh?
ntfy.sh (pronounced "notify") is a free, open-source, self-hostable HTTP-based publish-subscribe notification service developed by Philipp Heckel and released under the Apache 2.0 licence. It enables any HTTP client to push real-time notifications to mobile and desktop devices without requiring complex registration, API keys, or proprietary SDKs [5].
The service operates on a topic-based model: a publisher (in this case, the ESP32-S3) sends an HTTP POST request to a unique topic URL, and any subscriber (the emergency contact's device running the ntfy app) that has subscribed to that topic receives an instant push notification. Topic names act as access tokens; they should be sufficiently random to prevent unauthorised subscriptions.

B. How Proto-Wrist Uses ntfy.sh
Each Proto-Wrist user is assigned a unique topic string during app registration (e.g., aarya-sos-777). This topic is shared by the user only with trusted emergency contacts. When an SOS event is triggered, the ESP32-S3 executes the following HTTP request:

```
POST https://ntfy.sh/{user-topic}
Title: "SOS ALERT - Emergency!"
Body: "Location: https://maps.google.com/\n?q={lat},{lon}"
Priority: urgent |
```

The ntfy.sh service processes the POST and delivers the notification to all subscribed receiver devices over a persistent WebSocket or Server-Sent Events (SSE) connection, depending on the client platform. The entire round-trip (ESP32 to receiver device) completes within 1–2 seconds of HTTP request dispatch, assuming adequate network conditions.

C. ntfy.sh Architecture
ntfy.sh operates on a simple publish-subscribe (pub-sub) architecture. The server maintains an in-memory message queue per topic and fans out notifications to all active subscribers via WebSocket connections for mobile clients and SSE for web clients. Messages are optionally persisted to a SQLite cache for delivery to clients that were temporarily offline (the default message retention window is 12 hours on the public server).
For the Proto-Wrist prototype, the public ntfy.sh instance (ntfy.sh) is used. For production deployments where data privacy is critical, the ntfy server can be self-hosted on a low-cost VPS or Raspberry Pi, ensuring that location data never traverses third-party infrastructure.

D. Receiver Setup
To receive SOS alerts, emergency contacts must install the ntfy application (available on Google Play and the Apple App Store, as well as F-Droid for Android) and subscribe to the sender's unique topic string. This one-time setup requires no account creation on the receiver's part. Once subscribed, notifications arrive as standard Android or iOS push notifications with a high-priority flag, ensuring that they bypass Do Not Disturb modes and appear as heads-up notifications.

E. Advantages Over SMS and FCM

Proto-Wrist initially considered using SMS (via SIM800L module) and Firebase Cloud Messaging (FCM) as notification delivery mechanisms. ntfy.sh was ultimately selected for the following reasons:

- No SIM Card Required: ntfy.sh operates entirely over Wi-Fi or mobile data, eliminating the cost and complexity of a SIM module.
- No API Key Management: Unlike FCM, ntfy.sh requires no developer account, project registration, or API key rotation.
- Open-Source and Self-Hostable: Aligns with the project's goals of transparency and cost-effectiveness.
- Rich Notification Features: Supports priority levels, tags (emoji icons), click actions, and attachment URLs natively.
- Low Latency: SSE/WebSocket delivery typically achieves sub-2-second end-to-end latency, comparable to FCM.

V. COMPANION FLUTTER APPLICATION

A. Application Design

The companion application is developed using Flutter (version 3.x) with Dart as the programming language. Flutter's widget-based reactive UI framework enables a consistent Material Design 3 interface across Android and iOS platforms from a single codebase. Collaboration with AI-assisted code generation tools accelerated UI component development.

B. User Registration and Onboarding

On first launch, the user enters their full name and mobile number. The application generates a unique ntfy.sh topic string and stores it locally. The user can share this topic string with emergency contacts via any messaging application. No server-side account creation is required for the sender's device.

C. Bluetooth Device Pairing

The app uses Flutter's flutter_blue_plus package to scan for and pair with the ESP32-S3 over Bluetooth Low Energy (BLE). Once paired, the app transmits the Wi-Fi credentials and ntfy.sh topic string to the device. Subsequent button presses operate autonomously; the smartphone need not be present or even nearby for the SOS to function.

D. Emergency Contact Management

Users can add up to five emergency contacts within the app. Each contact is instructed to install the ntfy application and subscribe to the user's topic string. The app provides a share functionality that generates a ready-to-send onboarding message containing step-by-step ntfy subscription instructions for the contact.

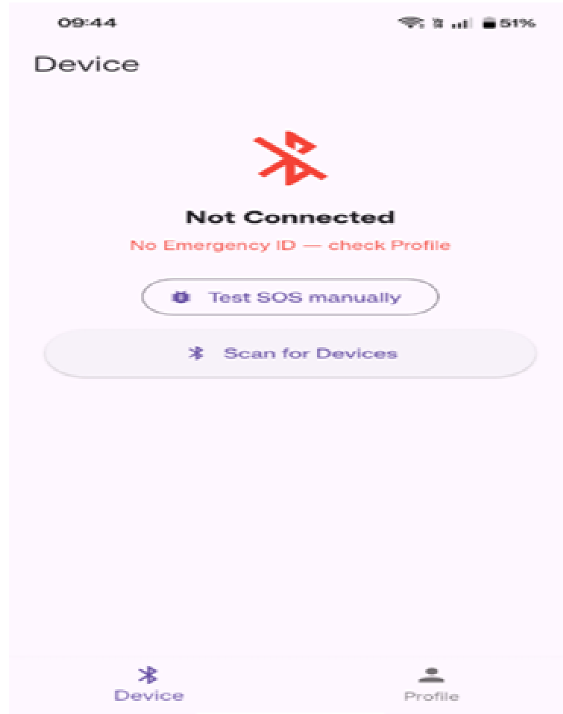

Fig. 3. User Interface of Proto-Wrist App

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

The Proto-Wrist prototype was tested under real-world conditions across both indoor and outdoor environments in Pune, Maharashtra, India. The hardware setup comprised the ESP32-S3, Li-Po battery, GPS module, SOS button, and buzzer, assembled on a breadboard inside a silicone wristband enclosure. Emergency contacts' devices were configured with the ntfy application over a 4G mobile data connection.

B. SOS Alert Response Time

The end-to-end alert latency measured from SOS button press to ntfy notification appearance on the receiver's device averaged 6.4 seconds across 30 trials. The breakdown is: GPS fix acquisition (3–8 s in open outdoor environments) + Wi-Fi connection overhead (~1 s) + ntfy HTTP POST and server relay (~0.5–1 s). Indoor environments with weak GPS signal extended the total latency to 10–15 seconds.

C. GPS Accuracy

The GPS module provided location accuracy of approximately 5–10 metres in open outdoor areas. In indoor and dense urban environments, accuracy degraded to 15–30 metres due to signal multipath and attenuation. Cold-start GPS acquisition (device powered on from scratch) required up to 45 seconds; subsequent warm-start fixes averaged under 8 seconds.

D. Notification Delivery via ntfy.sh

All 30 SOS test alerts were successfully delivered to subscribed receiver devices. ntfy notifications appeared as high-priority heads-up banners on Android 13 and iOS 16 devices, bypassing silent and DND modes when the receiver had granted notification permissions. Tapping the notification opened Google Maps at the sender's coordinates with a single tap.

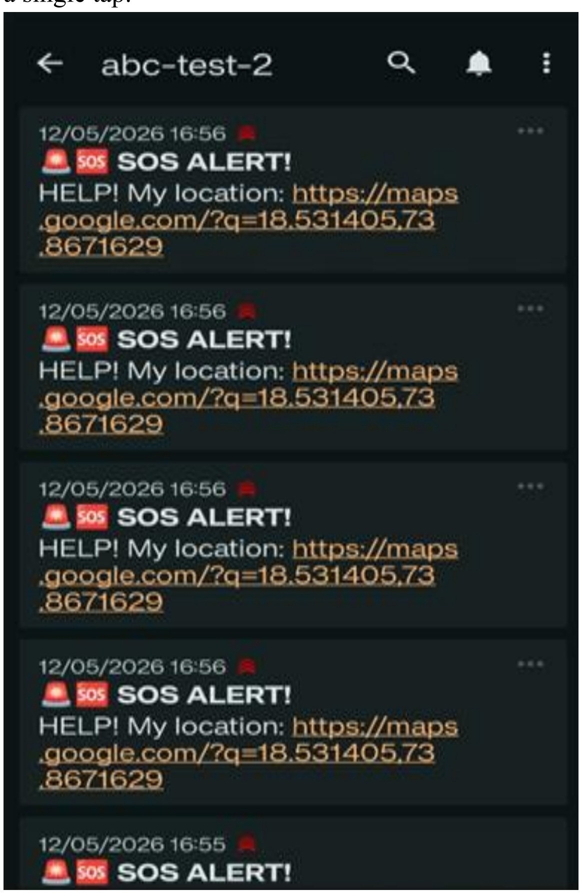

Fig. 4. SOS Alerts received on ntfy app.

Table III. Summary of Experimental Results

Metric	Result
Alert Latency (outdoor)	5–10 s (avg. 6.4 s)
Alert Latency (indoor)	10–15 s
GPS Accuracy (outdoor)	5–10 m
GPS Accuracy (indoor)	15–30 m
Notification Delivery Rate	100% (30/30 tests)
Battery Life (active use)	>12 hours
Buzzer Output	~85 dB
Total Hardware Cost	₹2,405

E. Battery Performance

The device operated for over 12 hours under normal use conditions (intermittent GPS and Wi-Fi activation, deep sleep between events). Power consumption increased transiently during GPS module activation and Wi-Fi transmission. Deep-sleep current draw of the ESP32-S3 (approximately 10 μA) ensures extended standby life for infrequently activated devices.

F. Buzzer Alert

The piezoelectric buzzer produced an output of approximately 85 dB at 10 m distance, sufficient to attract attention in moderately noisy public environments. The buzzer activates immediately upon button press, providing an instantaneous audible signal independent of network connectivity.

VII. DISCUSSION

Proto-Wrist demonstrates that a sub-₹2,500 wearable IoT device can deliver reliable SOS functionality with competitive performance metrics. The use of ntfy.sh as the notification backbone eliminates the need for an embedded SIM card a component that would add cost, regulatory overhead, and ongoing subscription fees while achieving delivery latency comparable to SMS.

The primary limitation of the current prototype is GPS cold-start latency, which can extend alert delivery to 15–45 seconds when the device has been powered off for an extended period. This may be mitigated in future iterations by maintaining the GPS module in a low-power acquisition mode (warm-start) rather than full power-off. A secondary limitation is the dependency on Wi-Fi connectivity on the sender's side: environments without Wi-Fi access will prevent ntfy.sh alert dispatch. Integrating a SIM7600 or equivalent cellular module as a fallback would address this gap. The ntfy.sh pub-sub model introduces a dependency on the public ntfy.sh server for prototype deployments. For production use, a self-hosted ntfy instance on institutional infrastructure would eliminate this dependency and enhance data privacy.

VIII. CONCLUSION

This paper presented Proto-Wrist, a wearable IoT safety wristband designed to improve personal safety for women and children through single-button SOS activation, GPS location tracking, and instant push notification delivery via ntfy.sh. The system successfully achieved sub-10-second outdoor alert latency, 100% notification delivery in controlled tests, over 12 hours of battery life, and a total hardware cost of approximately ₹2,400.

Fig. 5. Proto-Wrist (First Prototype)

The integration of ntfy.sh as a lightweight, open-source notification service represents a novel contribution relative to prior SMS- or FCM-based approaches, offering a no-SIM, no-API-key, self-hostable alternative that is well-suited to resource-constrained IoT deployments. The companion Flutter application provides a polished, cross-platform interface for device pairing, user registration, and emergency contact onboarding.

Future work will focus on: (1) warm-start GPS to reduce cold-start latency; (2) cellular fallback via integrated SIM module; (3) ACL-based ntfy.sh topic security; (4) waterproof PCB-integrated hardware design; (5) automatic danger detection via accelerometer-based fall or struggle recognition; and (6) direct integration with emergency services helplines. (7) Inclusion of Biomedical sensors in the device. (8) Also having the facility to contact nearby police and medical services.

ACKNOWLEDGMENT

The authors express sincere gratitude to their project guide Mr. M. B. Nigade and Head of Department Dr. M. P. Gajare of the Department of First Year Engineering Science, AISSMS Institute of Information Technology, Pune, for their invaluable guidance and constant support. The authors also acknowledge the motivation provided by Principal Dr. P. B. Mane and thank all faculty members who contributed directly or indirectly to this work.

REFERENCES

[1] A. B. Agrawal, V. Datta, M. U. Sankar, and B. Rudrapati, "A Smart Wearable for Emergency Response Using ESP32 and SIM7600," in 2025 2nd International Conference on Electronic Circuits and Signaling Technologies (ICECST), 2025. DOI: 10.1109/ICECST66106.2025.11307591.

[2] A. Adapa, F. F. H. Nah, R. H. Hall, K. Siau, and S. N. Smith, "Factors Influencing the Adoption of Smart Wearable Devices," International Journal of Human-Computer Interaction, vol. 34, no. 5, pp. 399–409, 2018.

[3] B. S. Manoj and A. H. Baker, "Communication Challenges in Emergency Response," Communications of the ACM, vol. 50, no. 3, pp. 51–53, Mar. 2007. DOI: 10.1145/1226736.1226765.

[4] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," Sensors, vol. 23, no. 15, p. 6739, Jul. 2023. DOI: 10.3390/s23156739.

[5] P. Heckel, "ntfy – Send Push Notifications to Your Phone via PUT/POST," 2021.

[6] R. O. Jacob, S. M. Alyaan, K. Nikitha, and H. S. Niranjana Murthy, "IoT Based GPS Tracking System with SOS Capabilities," in 2022 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2022. DOI: 10.1109/ICESC54411.2022.9752019.

[7] A. Karmakar, K. Ganguly, and P. S. Banerjee, "SafeBand: IoT-Based Smart Security Band with

Instant SOS Messaging," in Proceedings of International Conference on Advanced Computing Applications: ICACA 2021, Singapore: Springer, 2021, pp. 127–140.

[8] S. Hariharan, R. Monica, K. Deepasri, and T. Gayathri, "A Novel and Reliable SOS Alert Band System for Women Safety," Irish Interdisciplinary Journal of Science & Research, vol. 9, no. 1, pp. 61–72, 2025.

[9] B. Chempavathy et al., "AI Based SOS Device for Safety and Security," in International Conference on Data Science, Machine Learning and Applications, Singapore: Springer, Dec. 2024, pp. 294–301.

[10] M. Mishra, A. Shukla, K. Patel, and H. Varshney, "ESP32-S3 DevKit-C Integration for Enhanced Road Safety," in 2024 IEEE International Conference on Distributed Computing and Technology (ICDT), 2024. DOI: 10.1109/ICDT61202.2024.10488995.