

Standalone Wearable Fall Detection System Using ESP32 and GSM-Based Emergency Alerting

Darshan K¹, Mrs. A. Sathiya Priya²

Student, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore¹

Assistant Professor, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore²

Abstract: This is a fully standalone wearable fall detection and emergency alert system designed for elderly individuals. The system integrates the MPU-6050 inertial measurement unit (IMU) with an ESP32 microcontroller to detect falls using a dual-stage impact and orientation-based threshold algorithm. Unlike cloud-dependent systems, the proposed design performs all signal processing locally and transmits emergency alerts directly through a GSM module without requiring internet connectivity.

A multi-stage validation mechanism combining Signal Magnitude Vector (SMV), posture angle deviation, and post-impact persistence minimizes false positives during daily activities. Upon confirmed fall detection, an SMS alert is immediately transmitted to predefined caregiver numbers using the SIM800L GSM module [9]. Experimental evaluation demonstrates 96.8% sensitivity, 95.1% specificity, and an average alert latency of 2.4 seconds. The system operates at 145 mW, enabling approximately 22 hours of continuous monitoring on a 2000 mAh Li-ion battery.

Keywords: Fall Detection; Wearable Device; ESP32; MPU-6050; GSM Alerting; Embedded Systems; Elder Safety

I. INTRODUCTION

Falls remain one of the leading causes of injury-related mortality among older adults. According to the World Health Organization, falls result in over 684,000 deaths annually worldwide [1]. The Centers for Disease Control and Prevention reports millions of emergency visits each year due to fall-related injuries [2].

A critical challenge is delayed medical assistance. Many elderly individuals are unable to manually call for help following a severe fall. Therefore, autonomous detection and automatic emergency notification are essential [3].

Unlike internet-dependent IoT systems, this work focuses on a network-independent, GSM-based alerting mechanism, ensuring reliability even in environments without Wi-Fi connectivity. The proposed system uses an ESP32 microcontroller [7] paired with an MPU-6050 IMU [8] for real-time edge processing.

II. RELATED WORK

Early fall detection systems relied on single-axis acceleration thresholds [3]. Multi-axis approaches improved detection reliability using signal magnitude vector calculations [4]. Vision-based systems achieved high accuracy but introduced privacy concerns and limited deployment flexibility [5].

Machine learning approaches offer improved accuracy but demand significant computational resources and often require cloud processing [6]. Smartphone-based systems [11, 12] provide moderate portability but depend on network connectivity for alerting.

Most existing systems suffer from one or more of the following limitations:

- Internet dependency
- High power consumption
- High deployment cost
- Privacy concerns

The proposed system eliminates cloud reliance by implementing complete edge processing and GSM-based alerting [9], addressing these gaps while remaining low-cost and deployable in rural settings.

III. SYSTEM ARCHITECTURE

The system consists of three functional layers:

1. Sensing Layer

MPU-6050 IMU [8] captures tri-axial acceleration and angular velocity at 100 Hz.

2. Embedded Processing Layer

ESP32 [7] performs:

- Noise filtering
- SMV computation
- Orientation estimation
- Fall decision logic

3. Alert Layer

- SIM800L GSM module [9] sends SMS alerts directly to caregivers.
- Local buzzer and LED provide immediate on-site alert.

All processing occurs on-device, eliminating cloud latency and privacy risks.

IV. HARDWARE DESIGN

A. Inertial Measurement Unit

The MPU-6050 [8] combines:

- 3-axis accelerometer (impact detection)
- 3-axis gyroscope (orientation tracking)

B. Microcontroller

ESP32 [7] features:

- Dual-core 240 MHz processor
- Low power modes
- Sufficient RAM for real-time filtering

C. GSM Communication

SIM800L module [9]:

- Quad-band GSM
- AT-command controlled
- Direct SMS capability
- Average SMS dispatch time: ~2 seconds

D. Power System

- 3.7V 2000 mAh Li-ion battery
- TP4056 charging module
- AMS1117-3.3 voltage regulator
- Average power consumption: ~145 mW
- Estimated runtime: ~22 hours

Fall Detection and Alert System by GSM (IM SDE)

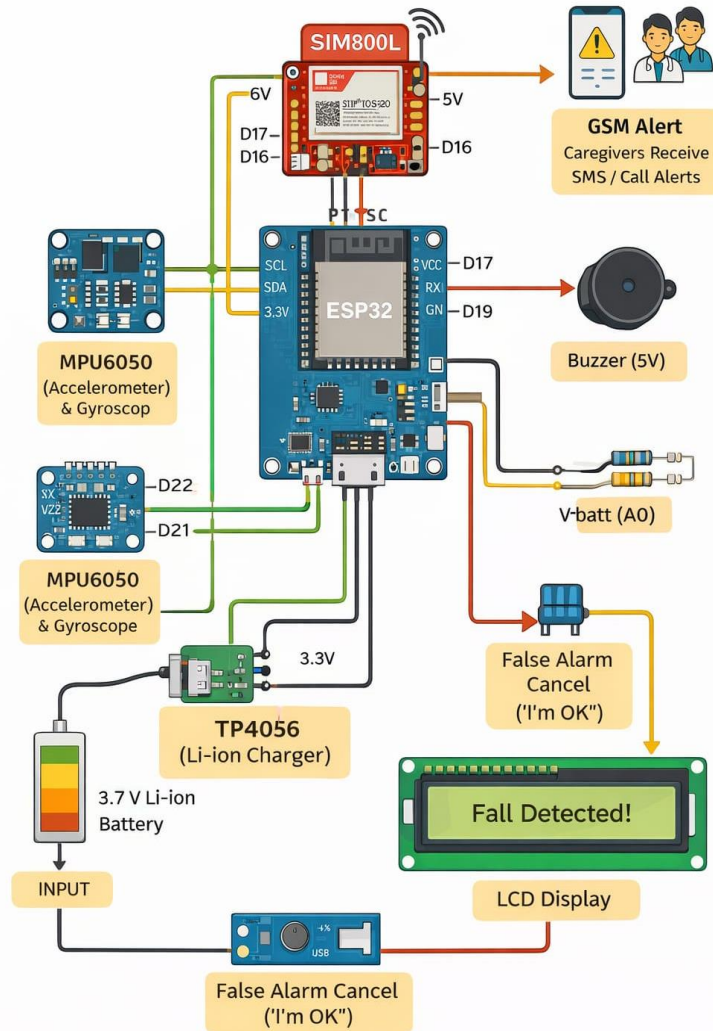


Figure 1: Fall Detection and Alert System Circuit Diagram (ESP32 + SIM800L + MPU6050)

V. FALL DETECTION ALGORITHM

5.1 Signal Magnitude Vector (SMV)

$$A = \sqrt{(Ax^2 + Ay^2 + Az^2)}$$

Normal state $\approx 1g$. This metric is widely used in fall detection literature [4, 10].

5.2 Dual-Threshold Detection

Fall is confirmed when [4]:

- Impact acceleration exceeds $T1 = 2.5g$
- Orientation angle deviation exceeds $T2 = 60^\circ$
- Horizontal posture persists for 1.5 seconds

This ensures high discrimination between real falls and vigorous activities [10].

5.3 Moving Average Filtering

Five-sample window:

$$Af[k] = (1/5) \sum A[k-i] \quad (i = 0 \text{ to } 4)$$

Reduces sensor noise without distorting impact peaks.

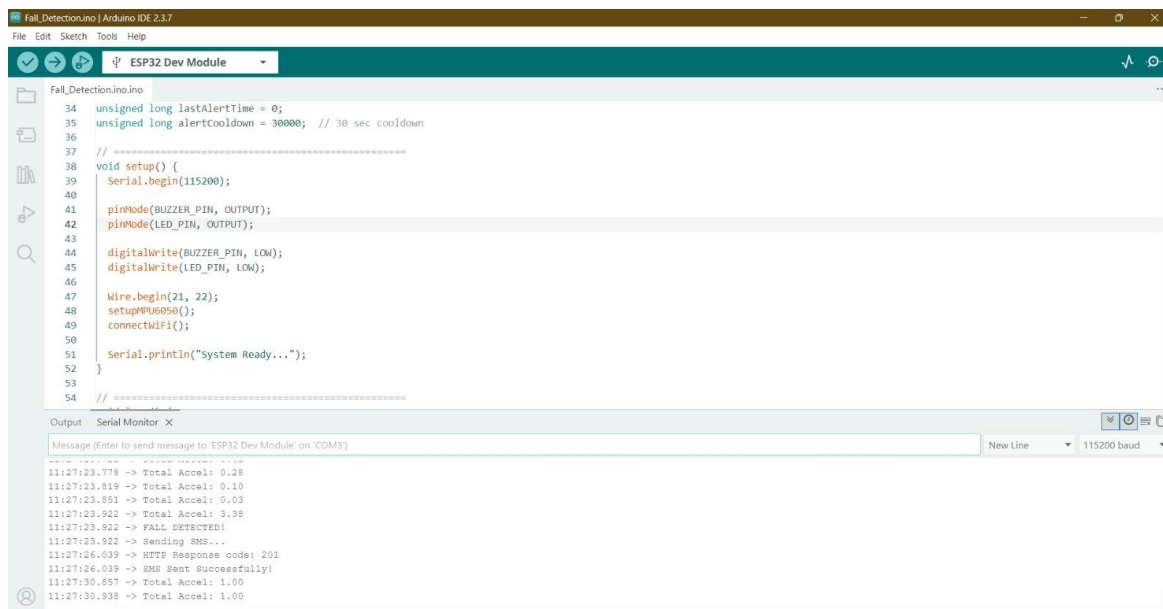
5.4 Complementary Filter

$$\theta = \alpha(\theta_{prev} + \omega dt) + (1-\alpha)\theta_{acc}$$

Where $\alpha = 0.98$, balancing gyroscope drift and accelerometer noise [8].

5.5 Detection Flow

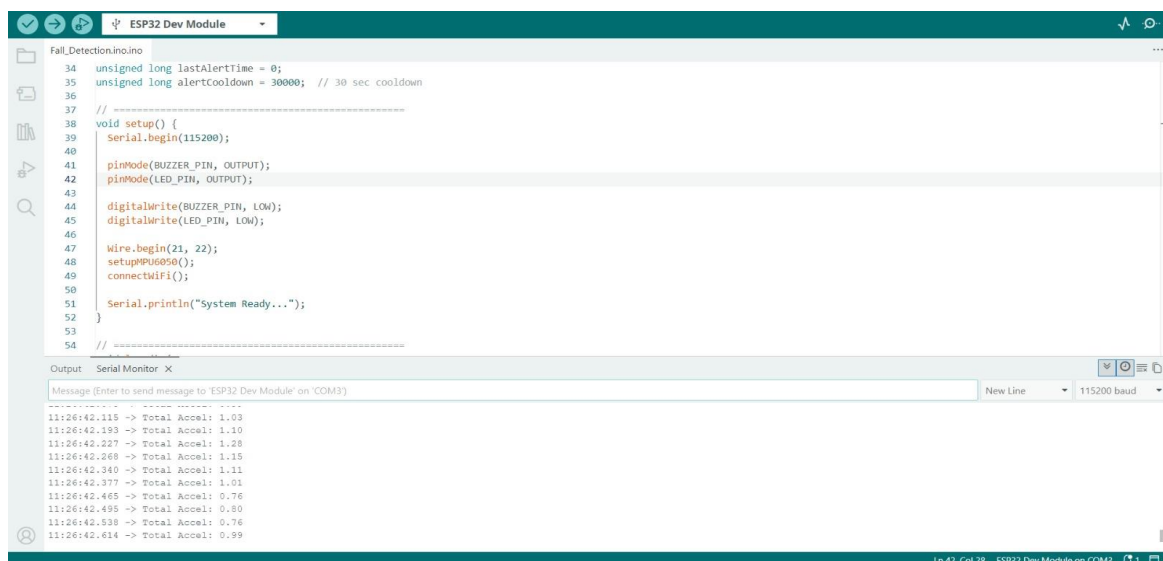
- Read IMU data (100 Hz)
- Apply moving average filter
- Compute SMV
- Estimate orientation
- If SMV > 2.5g → set impact flag
- If orientation > 60° within 1.5s → Confirm fall
- Activate buzzer + send GSM SMS



```

Fail_Detection.ino | Arduino IDE 2.3.7
File Edit Sketch Tools Help
ESP32 Dev Module
Fail_Detection.ino
34 unsigned long lastAlertTime = 0;
35 unsigned long alertCooldown = 30000; // 30 sec cooldown
36
37 // =====
38 void setup() {
39   Serial.begin(115200);
40
41   pinMode(BUZZER_PIN, OUTPUT);
42   pinMode(LED_PIN, OUTPUT);
43
44   digitalWrite(BUZZER_PIN, LOW);
45   digitalWrite(LED_PIN, LOW);
46
47   Wire.begin(21, 22);
48   setupMPU6050();
49   connectWiFi();
50   Serial.println("System Ready...");
51 }
52
53
54 // =====
Output Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
11:27:23.778 -> Total Accel: 0.28
11:27:23.819 -> Total Accel: 0.10
11:27:23.851 -> Total Accel: 0.03
11:27:23.922 -> Total Accel: 3.38
11:27:23.922 -> FALL DETECTED!
11:27:23.922 -> Sending SMS...
11:27:26.039 -> HTTP Response code: 201
11:27:26.039 -> SMS Sent Successfully!
11:27:30.857 -> Total Accel: 1.00
11:27:30.938 -> Total Accel: 1.00
    
```

Figure 2: Arduino IDE Serial Monitor – Fall Detected (Total Accel: 3.38g) and SMS Sent Successfully



```

Fail_Detection.ino
34 unsigned long lastAlertTime = 0;
35 unsigned long alertCooldown = 30000; // 30 sec cooldown
36
37 // =====
38 void setup() {
39   Serial.begin(115200);
40
41   pinMode(BUZZER_PIN, OUTPUT);
42   pinMode(LED_PIN, OUTPUT);
43
44   digitalWrite(BUZZER_PIN, LOW);
45   digitalWrite(LED_PIN, LOW);
46
47   Wire.begin(21, 22);
48   setupMPU6050();
49   connectWiFi();
50   Serial.println("System Ready...");
51 }
52
53
54 // =====
Output Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
11:26:42.115 -> Total Accel: 1.03
11:26:42.193 -> Total Accel: 1.10
11:26:42.227 -> Total Accel: 1.28
11:26:42.268 -> Total Accel: 1.15
11:26:42.340 -> Total Accel: 1.11
11:26:42.377 -> Total Accel: 1.01
11:26:42.465 -> Total Accel: 0.76
11:26:42.495 -> Total Accel: 0.80
11:26:42.538 -> Total Accel: 0.76
11:26:42.614 -> Total Accel: 0.99
Ln 42, Col 28 ESP32 Dev Module on COM3
    
```

Figure 3: Arduino IDE Serial Monitor – Normal Operation (Total Accel ≈ 1g)

VI. EXPERIMENTAL SETUP

- 12 participants
- 480 simulated falls
- 410 ADL trials
- Waist-mounted device configuration
- Indoor domestic-like test environment

Test protocols were adapted from standard fall simulation methodologies [10].

VII. RESULTS

Metric	Value
Sensitivity	96.8%
Specificity	95.1%
Overall Accuracy	96.0%
False Positive Rate	4.9%
Mean Alert Latency	2.4 s
Power Consumption	145 mW
Battery Life	~22 hrs

Latency breakdown:

- Edge processing: 0.35 s
- GSM SMS transmission: 2.05 s

The sensitivity of 96.8% and specificity of 95.1% compare favourably with similar threshold-based wearable systems in the literature [4, 10, 11].

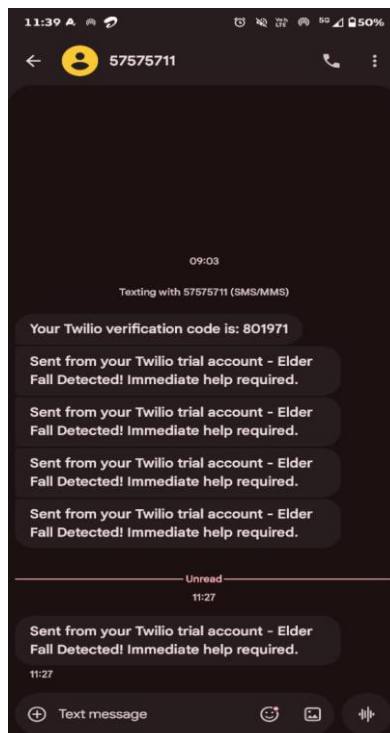


Figure 4: SMS Alert Received on Caregiver's Phone – "Elder Fall Detected! Immediate help required."

**VIII. ADVANTAGES**

- No internet required — operates independently of Wi-Fi [9]
- Works in rural and low-connectivity areas
- Privacy-preserving — all processing on-device
- Low cost (under INR 700)
- Real-time direct SMS alert
- Low power consumption (~145 mW)

IX. LIMITATIONS

- GSM network availability required
- Simulated fall testing only — real-world validation needed [10]
- Single sensor placement

X. FUTURE WORK

- GPS module integration for location tagging
- TinyML-based adaptive thresholding [6]
- Multi-sensor fusion (wrist + waist) [10]
- Long-term gait risk analysis

XI. CONCLUSION

This paper presented a standalone wearable fall detection system using ESP32 [7] and MPU-6050 [8] with GSM-based emergency alerting via SIM800L [9]. The dual-threshold impact-orientation algorithm achieves high detection accuracy while maintaining low power consumption and fast alert response.

By eliminating cloud dependency, the system ensures reliability, privacy, and usability in low-connectivity environments, making it suitable for rural and resource-constrained settings. Compared to existing approaches [3, 4, 11, 12], the proposed system offers a practical, affordable, and network-independent solution for elder fall safety.

REFERENCES

- [1]. World Health Organization, "Falls," WHO Fact Sheet, 2021. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/falls>
- [2]. Centers for Disease Control and Prevention, "Falls Are Leading Cause of Injury Death," CDC, 2023.
- [3]. N. Noury et al., "Fall detection — Principles and Methods," in Proc. IEEE EMBC, 2007, pp. 1663–1666.
- [4]. A. Bourke and G. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering & Physics*, vol. 30, no. 1, pp. 84–90, 2008.
- [5]. M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.
- [6]. T. Zhang et al., "Fall Detection by Wearable Sensor and One-Class SVM Algorithm," *Intelligent Computing in Signal Processing and Pattern Recognition*, 2006.
- [7]. Espressif Systems, "ESP32 Technical Reference Manual," Version 5.1, 2023. [Online]. Available: <https://www.espressif.com>
- [8]. InvenSense, "MPU-6050 Product Specification Revision 3.4," 2013. [Online]. Available: <https://invensense.tdk.com>
- [9]. SIMCom, "SIM800L Hardware Design V1.00," SIMCom Wireless Solutions, 2013.
- [10]. R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 1, p. 66, 2013.
- [11]. S. Abbate et al., "A smartphone-based fall detection system," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 883–899, 2012.
- [12]. F. Sposaro and G. Tyson, "iFall: An Android application for fall monitoring and response," in Proc. IEEE EMBC, 2009, pp. 6119–6122.