International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-599

Vol. 22 No. 4 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

SafeWay AI: An IoT-Based Intelligent Road Accident Detection and Emergency Alert System with Real-Time Location Tracking

G. Mahammad Afshan[1]
Student, B.Tech ECE
Department of Electronics and Communication Engineering
RGUKT RK Valley, Andhra Pradesh, India

S. Bhaskar Rao[2]
Assistant Professor
Department of Electronics and Communication Engineering
RGUKT RK Valley, Andhra Pradesh, India

Tunga Karuna Pradeep[3]
Student, B.Tech ECE
Department of Electronics and Communication Engineering
RGUKT RK Valley, Andhra Pradesh, India

A. Vishnu Teja[4]
Student, B.Tech ECE
Department of Electronics and Communication Engineering
RGUKT RK Valley, Andhra Pradesh, India

***Abstract*—Vehicle collisions remain a critical public-health challenge, and survival odds depend heavily on how quickly medical help arrives. India recorded over 480,000 road accidents in 2023, underscoring the fatal consequences of delayed reporting. This paper presents SafeWay AI, an autonomous IoT and mobile ecosystem that detects accidents and dispatches emergency notifications without human input. The embedded unit combines an ESP32 microcontroller with an MPU6050 inertial measurement unit, a NEO-6M GPS receiver and a SIM800L GSM modem to monitor vehicle kinematics continuously. When abnormal impact forces or rollover-level angular rates are detected, a 15-second grace-period countdown starts. If the driver does not cancel it, the unit acquires the current coordinates and sends SMS alerts, containing a map link, to predefined medical contacts and next of kin. We describe the hardware topology, the threshold-based detection algorithm, the dual-core firmware design and the companion mobile application. The detection logic and GSM command sequence were validated in a microcontroller simulation environment; physical road-level evaluation is planned.**

***Index Terms*—IoT, accident detection, ESP32, MPU6050, GPS, GSM, emergency alert, threshold-based detection, real-time monitoring.**

I. INTRODUCTION

Rapid deployment of emergency medical services is paramount for reducing road-accident fatalities. In practice, response times are limited by the inability of incapacitated victims to summon help themselves. According to the Ministry of Road Transport and Highways (MoRTH), India recorded 172,890 road-traffic deaths in 2023 [1], a toll worsened by undiscovered crash sites and delayed notification of emergency services.

Traditional emergency response relies heavily on bystander intervention. Reported IoT systems add accident detection and location reporting to the vehicle, either on smartphones and cloud platforms [2] or with GPS and GSM modules that send a map link by SMS [3]; ESP32-based designs that use the MPU6050 have also been demonstrated [4]. As noted in [2], many hardware-based systems depend on a single sensor, so one sensor failure disables the whole system. General-purpose location-sharing applications provide ambient tracking, but they are not built around a dedicated, hardware-assisted SOS pipeline that triggers automatically after a collision. SafeWay AI addresses this gap by combining embedded sensing hardware with a mobile application to automate the entire alerting chain, from impact detection to responder tracking.

The main contributions of this work are:

- A threshold-based impact-detection algorithm that uses multi-axis accelerometer and gyroscope data.
- A mobile dashboard that tracks responder ETAs and categorises emergencies (Medical, Police, Fire, Accident).
- A GSM-based hardware design that delivers SMS alerts even when mobile data is unavailable.
- A hands-free SOS protocol that completes without any driver input once the countdown expires.

II. SYSTEM ARCHITECTURE AND DESIGN

SafeWay AI divides its duties between a user-facing mobile application and an embedded hardware unit mounted on the vehicle.

A. Overall System Architecture

The system runs in a continuous loop of sensing, processing and communication. While the vehicle moves, the sensors collect kinematic data. When the ESP32 (or the smartphone processor) detects a dangerous spike in these readings, it requests the latest GPS fix, pushes the data over the cellular network to a cloud server and sends SMS alerts with a map link [3]. Fig. 1 summarises the data flow.

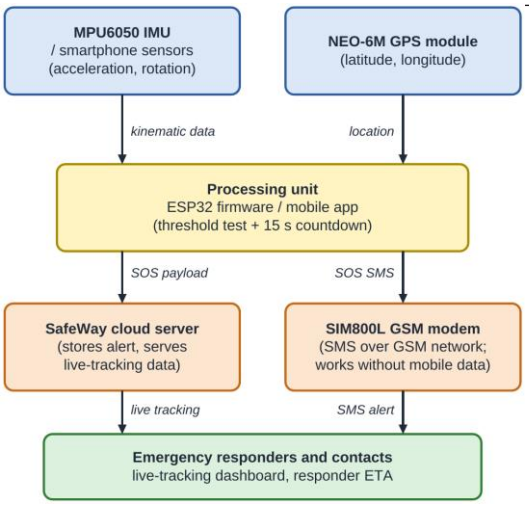


Fig. 1. System architecture and communication flow of SafeWay AI.

B. Hardware Architecture

The ESP32 [5] was selected for the prototype because its dual-core processor can run sensing and communication tasks concurrently and natively operates at 3.3 V logic levels. The hardware comprises three main modules (Fig. 2, Tables I and II):

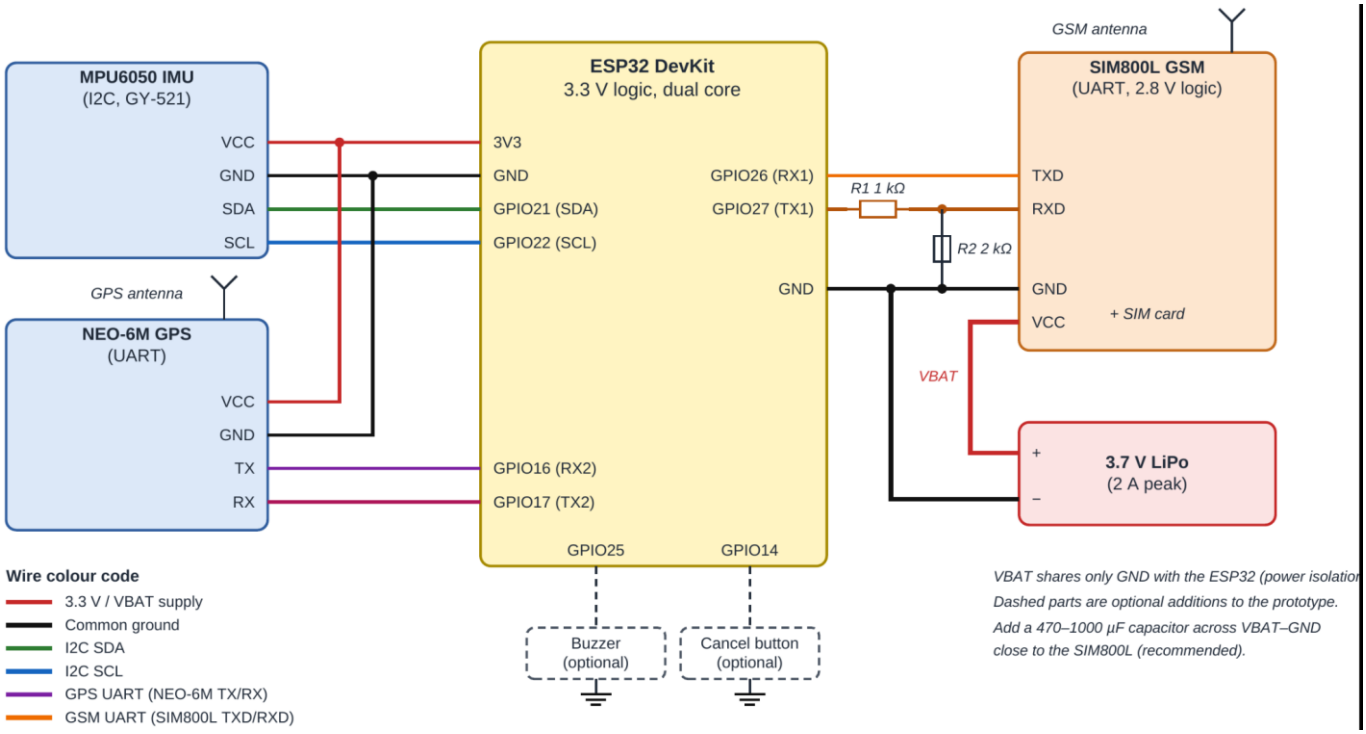


Fig. 2. Circuit diagram of the SafeWay AI hardware unit with module interfacing and power isolation. Pin numbers are a typical ESP32 assignment (Table II) and must be matched to the actual build.

- **Motion sensing (MPU6050 [6]):** connected over I2C, it measures acceleration and angular rate to identify high g-forces indicative of a collision.
- **Location tracking (NEO-6M):** connected to a UART port, it provides the vehicle's latitude and longitude from satellite signals.
- **Communication (SIM800L):** connected to a second UART port. Because it draws up to about 2 A in transmission bursts, it is powered from a separate 3.7 V LiPo battery that shares only the ground line with the ESP32, which keeps supply noise away from the logic signals. The SIM800L uses 2.8 V serial logic, so the ESP32 TX line should pass through a resistor divider (Fig. 2).

TABLE I
PROPOSED HARDWARE COMPONENTS

Component	Function in the proposed system
ESP32	Central logic, threshold evaluation and peripheral coordination.
MPU6050	Measures acceleration and angular rate to detect collisions.
NEO-6M GPS	Provides latitude and longitude of the incident.
SIM800L GSM	Transmits the automated SOS SMS to emergency contacts.
Smartphone app	Sensor-based detection, manual SOS and real-time responder ETA tracking.

TABLE II
TYPICAL PIN CONNECTIONS (MATCH TO THE ACTUAL BUILD)

Module pin	ESP32 / supply	Note
MPU6050 VCC, GND	3V3, GND	Power
MPU6050 SDA, SCL	GPIO21, GPIO22	I2C bus
NEO-6M VCC, GND	3V3, GND	Power
NEO-6M TX	GPIO16 (RX2)	UART2 receive
NEO-6M RX	GPIO17 (TX2)	UART2 transmit

Module pin	ESP32 / supply	Note
SIM800L TXD	GPIO26 (RX1)	UART1 receive
SIM800L RXD	GPIO27 (TX1)	Via R1/R2 divider
SIM800L VCC	3.7 V LiPo (+)	Not from ESP32
SIM800L GND	Common GND	Shared ground only

C. Mobile Application and Software

The mobile application is the control centre for users and responders. The current prototype has eight screens covering tasks such as selecting the emergency type (Medical, Police, Fire or Accident) and sharing live location. Family members and medical teams use the dashboard to see where the ambulance is and when it will reach the crash site.

III. ACCIDENT DETECTION AND ALERT METHODOLOGY

Distinguishing routine driving vibrations (for example, potholes) from severe impacts is essential to avoid false alarms.

A. Kinematic Data Acquisition

The sensors sample the vehicle's dynamic forces continuously. The instantaneous total acceleration magnitude $A(t)$ is

$$A(t) = \sqrt{(a_x^2 + a_y^2 + a_z^2)} \quad (1)$$

where a_x, a_y and a_z are the linear acceleration components along the X, Y and Z axes. A stationary sensor already reads about 1 g (9.81 m/s²) because of gravity, so the threshold must lie well above this baseline. To identify potential rollovers, the system also monitors the angular-velocity magnitude $\Omega(t)$:

$$\Omega(t) = \sqrt{(g_x^2 + g_y^2 + g_z^2)} \quad (2)$$

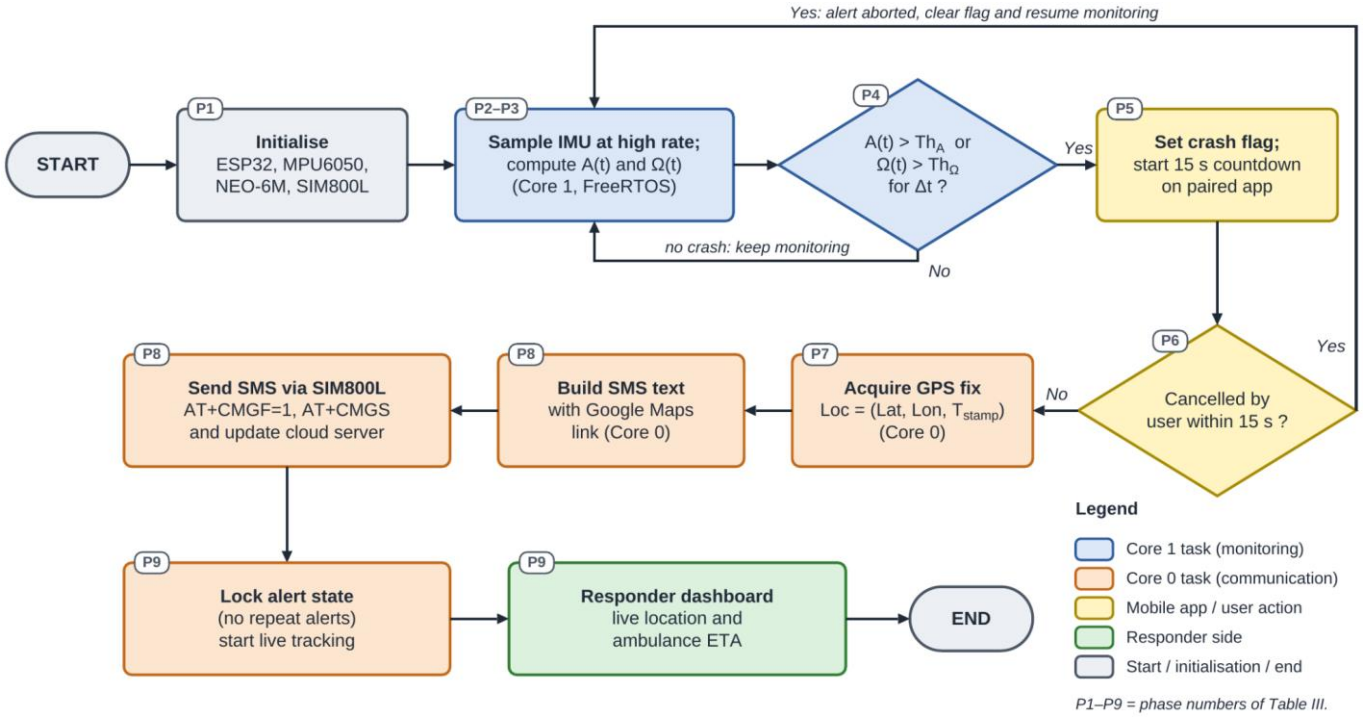


Fig. 3. Flowchart of the accident detection and alert algorithm (the P-tags refer to the phases of Table III).

where g_x, g_y and g_z are the rotational rates measured by the gyroscope [6]. The gyroscope range is also widened (for example to ±2000 °/s) so that fast rollover rotation does not saturate the reading.

B. Threshold-Based Identification

Upper limits Th_A and Th_Ω are set from crash-physics considerations. A candidate event exists when either limit is exceeded:

$$C(t) = [A(t) > Th_A] \vee [\Omega(t) > Th_\Omega] \quad (3)$$

The system registers a critical accident event (*Accident* = 1) only if $C(t)$ holds continuously for a duration Δt; otherwise *Accident* = 0. This duration check filters out transient shocks, and Δt is to be tuned on trial data. The accelerometer full-scale range must exceed the chosen threshold, otherwise the reading saturates. The MPU6050 default range of ±2 g saturates at about 19.6 m/s², so a wider range (for example ±16 g) is configured for crash detection.

C. False-Alarm Reduction and Countdown

Accidental drops of the device or harsh braking should not dispatch medical teams. A validated impact therefore starts a 15-second local countdown on the paired mobile interface. If the occupant is unharmed, a manual cancellation aborts the alert. If the countdown expires, the occupant is assumed to be incapacitated and the system escalates to the communication phase. An optional buzzer and cancel push-button on the unit (dashed in Fig. 2) can provide the same cancellation without the phone.

D. GPS Acquisition and Alert Transmission

After an un-cancelled crash is confirmed, the controller queries the NEO-6M for a satellite fix, which is cached as

$$\text{Loc} = (\text{Lat}, \text{Lon}, T_{\text{stamp}}) \quad (4)$$

The GSM modem then formats these coordinates into an SMS containing a Google Maps hyperlink and sends it to the designated responders and emergency cloud server. Fig. 3 gives the complete detection flow.

IV. SYSTEM IMPLEMENTATION AND WORKING PRINCIPLE

A. Dual-Core Software Architecture

To keep the system responsive while it monitors the road, the ESP32 firmware uses the FreeRTOS environment and divides the workload across the two processor cores:

- **Core 1 (real-time monitoring):** dedicated to reading the MPU6050 at a high rate. When the computed values indicate a crash, it sets a digital flag.
- **Core 0 (communication and alerting):** runs in the background, keeps the GPS location updated and, when it sees the crash flag, sends the AT commands to the GSM modem and builds the Google Maps URL.

This split reduces the chance that the system misses secondary impacts during the multi-second latency of cellular transmission.

B. Operational Workflow

The operating sequence is organised for fail-safe execution (Table III and Fig. 3).

TABLE III
SYSTEM OPERATING SEQUENCE

Phase	Action / process
1	**Initialisation:** the microcontroller boots and connects to GPS satellites and the cellular network.
2	**Monitoring:** continuous data acquisition from the MPU6050 or smartphone sensors.
3	**Impact analysis:** $A(t)$ and $\Omega(t)$ are computed and compared with Th_A and Th_Ω.
4	**Identification:** a potential accident state is triggered.
5	**Countdown:** 15-second grace period for manual cancellation.
6	**Confirmation:** alert verified (no user override).
7	**Location retrieval:** latitude and longitude are extracted.
8	**Transmission:** SMS is generated and the server is updated.
9	**Response:** live tracking starts on the responder dashboard.

V. SIMULATION RESULTS

The SafeWay AI mobile prototype has completed initial validation of the user interface, location sharing and responder-tracking concepts. The embedded logic was verified in a microcontroller simulation environment before physical deployment.

A. Simulation of Accident Detection

To validate the detection model and the UART command sequence without risking hardware, the MPU6050 kinematic algorithm and the SIM800L AT-command sequence were simulated. A crash threshold of 20.0 m/s² was used for the test. For a simulated sudden impact, the system registered a peak acceleration of 20.19 m/s² and immediately entered the emergency state.

The controller then halted normal sensor monitoring to prevent duplicate alerts and executed the AT-command sequence below (the recipient number is masked):

```
CRASH DETECTED! Impact Force: 20.19 m/s^2
>>> INITIATING GSM EMERGENCY SEQUENCE <<<
TX -> AT+CMGF=1
TX -> AT+CMGS="+91XXXXXXXXXX"
TX -> ALERT: Accident Detected!
    Location: https://maps.google.com/
    ?q=14.3644,78.4756
>>> SMS SUCCESSFULLY DISPATCHED <<<
```

The simulation confirms that the logic evaluates the multi-axis thresholds, formats the GPS coordinates into a standard URL and dispatches the SMS payload without serial blocking or string fragmentation (Tables IV and V). The 20.0 m/s² (≈2.04 g) threshold served only as a demonstration value; it lies just above the ±2 g default range, which is why the wider range of Section III-B is needed. The deployed threshold must be tuned on real driving data so that potholes and hard braking do not trigger alerts.

TABLE IV
SIMULATION TEST CASES AND RESPONSES

Test parameter / condition	Observed system response
Applied impact force 20.19 m/s², threshold > 20.0 m/s²	Threshold crossed (Pass)
Crash detection status	CRASH DETECTED (Pass)
GPS location generated	Lat 14.3644, Lon 78.4756 (Pass)
GSM SMS dispatch status	Successfully dispatched (Pass)
Repeat-alert prevention	System locked (Pass)

TABLE V
SIM800L AT-COMMAND SIMULATION RESULTS

AT command	Purpose	Response
AT	Check communication	OK
AT+CMGF=1	Set SMS text mode	OK
AT+CMGS="…"	Set target mobile number	>
Payload + Ctrl+Z	Send URL and payload	+CMGS: 14

VI. PLANNED PHYSICAL EVALUATION

After physical testing, the system will be evaluated with standard classification metrics. Detection accuracy is

$$\text{Acc} = (TP + TN) / (TP + TN + FP + FN) \times 100\% \quad (5)$$

where TP, TN, FP and FN are the true positives, true negatives, false positives and false negatives. The false-positive rate, which is critical for user acceptance, is

$$\text{FPR} = FP / (FP + TN) \quad (6)$$

Because a missed crash is the most serious failure, the recall (sensitivity) and the alert transmission delay will also be reported:

$$\text{Recall} = TP / (TP + FN) \quad (7)$$

$$T_d = t_{\text{SMS received}} - t_{\text{countdown expiry}} \quad (8)$$

Table VI lists the parameters that will be measured in physical prototype testing.

TABLE VI
PERFORMANCE PARAMETERS FOR PHYSICAL EVALUATION

Parameter	Method	Status
Detection accuracy	Eq. (5) over staged impact and normal-drive trials	To be measured
False-positive rate	Eq. (6) over normal-drive trials	To be measured
Recall (missed crashes)	Eq. (7) over staged impact trials	To be measured
GPS location error	Comparison with surveyed reference points	To be measured
Alert transmission delay	Eq. (8): countdown expiry to SMS receipt	To be measured
Sensor response time	Time from impact to flag being set	To be measured
System reliability	Successful alerts over total trials	To be measured

VII. APPLICATIONS, ADVANTAGES AND LIMITATIONS

A. Applications

- **Two- and four-wheeler safety:** can be integrated into vehicle electronics or used as a standalone smartphone application.
- **Fleet monitoring:** lets logistics companies monitor the safety and real-time location of transport assets.
- **Personal emergency assistance:** the multi-category SOS interface also supports non-traffic emergencies such as medical and fire incidents.

B. Advantages

The main advantage of SafeWay AI is that it removes the dependence on manual reporting in an emergency, so help can be summoned even when the victim is unconscious. Real-time tracking also improves coordination by letting family members follow the ambulance ETA.

C. Limitations

- **Network coverage:** the system depends on GSM and internet connectivity. In dead zones, cloud updates may fail, although SMS messages may still queue.
- **GPS availability:** signals can be blocked in tunnels or dense urban areas, temporarily degrading location accuracy.
- **Sensor calibration:** improper placement or calibration of the module in the vehicle can increase false alarms.
- **Validation stage:** results so far come from simulation; real-world accuracy and false-alarm rates remain to be established.

VIII. FUTURE SCOPE

- **AI/ML-based accident classification:** edge-deployed models to classify severity and type (for example, head-on collision versus rollover) from sensor patterns instead of fixed thresholds.
- **Threshold tuning:** collecting real driving data, including potholes, speed-breakers and hard braking, to set Th_A, Th_Ω and Δt.
- **In-vehicle hardening:** enclosure, backup supply, and an on-unit buzzer and cancel button.
- **Wearable integration:** connecting the app to smartwatches to send the victim's vital signs (heart rate, SpO_2) to paramedics in real time.
- **Smart traffic management:** integrating alerts with city infrastructure to control traffic signals and create a "green corridor" for ambulances.

IX. CONCLUSION

SafeWay AI addresses a critical weakness in current road-safety practice: the reliance on a conscious person to report an incident. By combining low-cost IoT components with a dedicated smartphone application, the work establishes an autonomous crash-detection and notification pipeline. Simulation results validate the ESP32 and MPU6050 logic and the GSM alert sequence, and the dual-core design is intended to prevent serial blocking during emergency transmission. Initial software prototyping also shows the viability of the mobile tracking dashboard. Future work will harden the device for in-vehicle deployment, complete physical testing and add edge-based machine learning to refine impact classification. Ultimately, autonomous alerting can shorten emergency response times and help reduce traffic-related mortality.

REFERENCES

[1] Ministry of Road Transport and Highways (MoRTH), *Road Accidents in India 2023*, Government of India, New Delhi, 2023.

[2] F. Bhatti, M. A. Shah, C. Maple and S. U. Islam, "A novel Internet of Things-enabled accident detection and reporting system for smart city environments," *Sensors*, vol. 19, no. 9, Art. no. 2071, May 2019, doi: 10.3390/s19092071.

[3] S. Sarker, M. S. Rahman and M. N. Sakib, "An approach towards intelligent accident detection, location tracking and notification system," arXiv:2001.00453, Dec. 2019.

[4] H. H. Azlan and S. Salimin, "Vehicle accident detection system using accelerometer," *Evol. Elect. Electron. Eng.*, vol. 4, no. 1, pp. 195–202, 2023.

[5] Espressif Systems, *ESP32 Series Datasheet*, Shanghai, China, 2023.

[6] InvenSense Inc., *MPU-6000 and MPU-6050 Product Specification*, Rev. 3.4, San Jose, CA, USA, 2013.