Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 280 (2026) 406–413

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 17th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 14-16, 2026, Istanbul, Türkiye

A Lightweight IoT Healthcare Wearable for Fall Detection and Ambient Hazard Sensing

Hakilo Sabit

[a]*Department of Electrical and Electronic Engineering, School of Engineering, Computer, and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand*

Abstract

Rapid advancements in digital and embedded technologies have transformed modern healthcare, enabling innovative approaches to continuous patient monitoring. Caring for elderly individuals presents ongoing challenges, particularly when caregivers cannot remain physically present to provide support. This project addresses this need by developing an Internet of Things (IoT)-enabled wearable monitoring system capable of delivering real-time access to key health and environmental indicators. The proposed device integrates multiple sensors to monitor vital signs and safety-related events, including fall detection, and thermal comfort parameters. Detected events—such as abnormal temperature levels or sudden falls—trigger immediate alerts, ensuring timely intervention during emergencies. All sensor readings are transmitted to a web server, where data are processed and presented through an accessible dashboard for remote monitoring. This work demonstrates a proof-of-concept wearable platform designed to enhance caregiver awareness, improve responsiveness, and support safer independent living for elderly individuals. The system provides a foundation for future development in IoT-based healthcare monitoring solutions, offering the potential for scalable and continuous oversight of vulnerable populations.

Keywords: IoT-healthcare; wearable; IoMT, fall detection; thermal comfort; online caregiver awareness

---

1. Introduction

The state of current technological advances are so pervasive that it has changed the way our society operates. This has changed the way humans now shop, work and live daily. One typical advancement in technology is that the influence it has on modern healthcare. Caring for an elderly family member presents significant challenges, particularly in

---

* Corresponding author. Tel.: +64-9921-9999-ext. 7941 ; fax: +0-000-000-0000.
E-mail address: hakilo.sabit@aut.ac.nz

1877-0509

ensuring their continuous safety and well-being. Providing appropriate healthcare support can become difficult when caregivers must attend to responsibilities outside the home, creating concern about potential incidents that may occur in their absence. Given that employment and other obligations often require caregivers to be away for extended periods, the need for reliable monitoring solutions becomes essential. A promising approach to addressing this challenge is the development of a wearable healthcare monitoring platform capable of supporting continuous, real-time oversight of the individual's condition.

This project aims to enable continuous access to patient vital signs and significant health-related events through a wearable, sensor-based system grounded in Internet of Things (IoT) technologies [1]. Advances in digital health and embedded sensing now make it possible to deliver healthcare monitoring in innovative and more effective ways than previously achievable [2]. The proposed work focuses on developing an IoT-enabled wearable device for elderly individuals, allowing caregivers to remotely monitor environmental comfort parameters and detect falls in real time, thereby enhancing safety, responsiveness, and overall quality of care.

The device is designed to provide caregivers with key physiological and environmental information, including heart rate, fall detection, and thermal comfort indicators. It will generate alerts when abnormal temperature readings or sudden falls are detected, ensuring timely intervention even when the caregiver is not physically present. All sensor data and event notifications will be transmitted to a web server, where they will be visualised in an accessible and comprehensible format. The goal is to develop a system capable of detecting, logging, and transmitting these events to an online platform that enables caregivers to remotely monitor and manage the patient's wellbeing with ease and reliability.

2. Literature Review

The most common form of wearable health technology would be Apple watch and all smart watch devices. These devices are advertised for all age groups and are marketed as a healthy lifestyle aid device. They have simple functions for health monitoring which can give heart rate, body temperature, pedometer functions. They have more advanced functions which can determine time spent most active, fall detection, and your food and water intake which is quite remarkable [3]. A health monitoring system offers numerous opportunities to transform traditional approaches to patient management [4]. In addition, such a solution can reduce overall healthcare costs, enhance clinical workflows, and facilitate the adoption of remote patient monitoring capabilities [5, 6, 7].

Fall detection is where this project will be focused on mostly. Fall detection is a major problem amongst elderly people as they are more prone to fall than others. Statistics has proven that each year, 3 million elderly people are treated in emergency departments for fall injuries [8]. Stats also suggest that 1 in 5 elderly fall injuries cause broken bones or head injury [8].

Healthcare monitoring systems are already widely implemented in many countries, including New Zealand, where organisations such as St John, the Ministry of Health, and ADT Security are actively investing in research on wearable health technologies for older adults. ADT's studies indicate that wearable solutions must remain subtle and minimally intrusive, as cultural and personal ethical concerns often influence user acceptance. Research also suggests that New Zealanders aged 65 and above—particularly those of NZ European descent—tend to be more receptive to such technologies [9](Superseniors, 2016). The number of people in these older ages could reach 1.3 million around 2040, and 1.5 million by the 2050s [10]. Because these projections are based on official New Zealand statistics, the findings are considered robust. Consequently, the market value for wearable healthcare technologies is expected to grow, driven by increasing demand and advancements in device functionality, form factor, and affordability.

3. Methodology and System Design

3.1. System Components: Microcontroller, Communication, and Sensors

The wearable healthcare monitoring system is built around the ESP32 FireBeetle, a low-power microcontroller designed for IoT applications. Its integrated Wi-Fi and Bluetooth capabilities enable reliable wireless data transmission to remote servers or mobile devices, while its energy-efficient architecture supports continuous operation in wearable devices. Two sensors provide the essential monitoring functions. The ADXL345 three-axis accelerometer detects motion and sudden impacts, allowing accurate fall detection through its high sensitivity and low noise performance. The DHT22 temperature and humidity sensor monitors ambient environmental conditions, offering stable and accurate readings to assess patient comfort and identify abnormal temperature changes.

Together, the ESP32 FireBeetle, ADXL345, and DHT22 form a compact and efficient hardware platform capable of capturing key activity and environmental data, supporting real-time alerts and remote monitoring within an IoT-based healthcare system.

3.2. Free-Fall Detection Theory

The fall detection system developed detects a fall by analysing raw data which comes from a 3-axis accelerometer. Threshold based monitoring has been developed based on data collected from the accelerometers and gyroscopes which are the two most common sensors for fall detection. The threshold method triggers when a monitored parameter surpasses a predefined limit. During software development for the module, the fall-detection sensitivity was calibrated by testing the accelerometer's response when dropped from various heights to identify an appropriate balance between excessive sensitivity and insufficient responsiveness. The accelerometer includes multiple built-in functions, including a free-fall detection feature that operates using a threshold-based mechanism. Free fall is set when acceleration detected is less than the value stored in the threshold register and is also experienced for longer than the time register threshold [11].

To provide context for the fall-detection process, the graphs in Figure 1 a and b illustrate accelerometer data collected during various daily activities and during simulated free-fall events. When a person sits down, the results show a pronounced spike in the Y-axis, indicating a rapid downward acceleration—an expected pattern when transitioning quickly to a seated or resting position. In contrast, when standing up, the Y-axis exhibits only gradual and minor changes in acceleration, reflecting the slower and more controlled upward movement, which typically requires greater effort than downward motion [11].

Free-fall detection using an accelerometer is typically characterised by three sequential phenomena [11]. First, weightlessness occurs when the vector sum of acceleration approaches 0 g, indicating that the body is momentarily in free fall. Although brief during real-life falls, this signature provides an important initial indicator. Second, impact follows, marked by a sharp acceleration spike as the body collides with the ground or an object, followed by a rapid return toward baseline as illustrated in Figure 2. This sudden change distinguishes actual falls from ordinary movements. Finally, during rest, the acceleration signal stabilises into a near-flat line, reflecting the period in which a person—particularly an elderly individual—remains motionless after a fall. Prolonged stillness may also indicate loss of consciousness. Together, these patterns form a reliable basis for fall detection and underpin the accelerometer's built-in free-fall functionality.

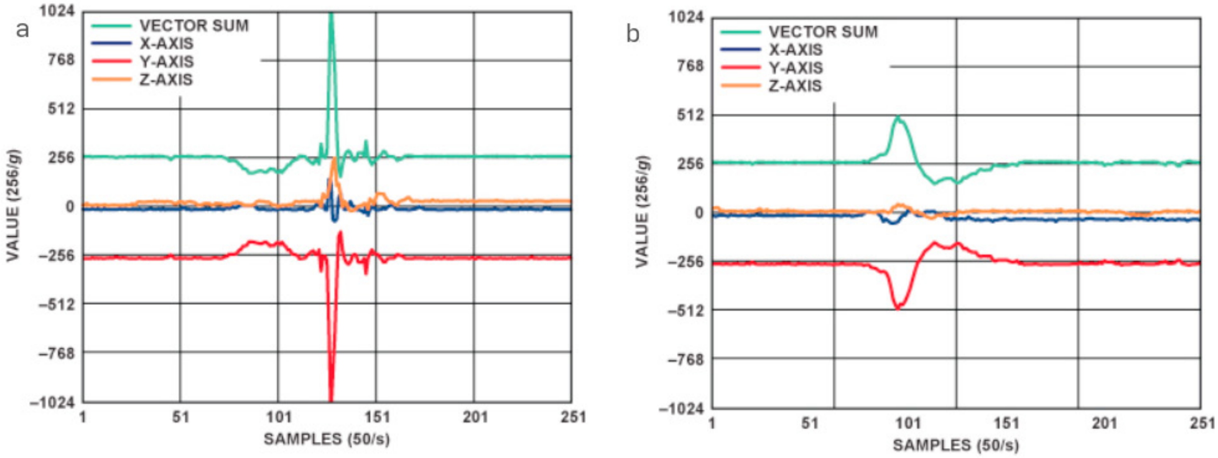


Fig. 1. (a) sitting down; (b) standing up.

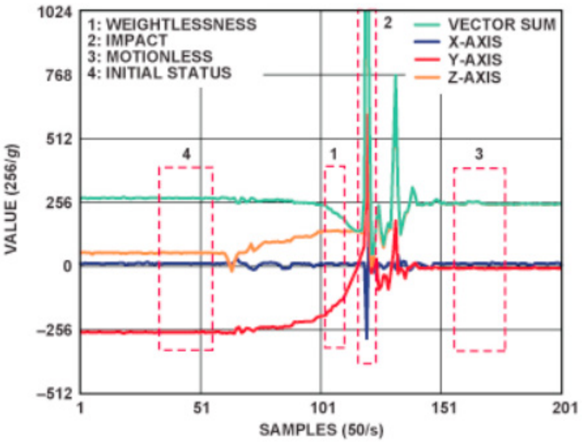


Fig. 2. free fall

3.3. Communication

The implementation of this device was meant to be seen over a network and accessible by a computer on the internet. Connected to the accelerometer is an ESP32 FireBeetle which processes the raw data from the sensor and formats it into human readable form to be sent over the internet to a website dashboard. The ESP32 utilises 2 communication protocols, SPI and I2C protocols. The protocol that is most common and the one used for this project is I2C. I2C protocol allows multiple slave devices to communicate to a master chip i.e., the ESP32 FireBeetle. Each I2C bus terminal communicates using 2 signals, an SDA and SCL signal. SDA is the data output signal. It comes in the form of 8 bits and are sent on the I2C bus. SCL is the clock signal. This is used to synchronise all data being transferred over the I2C bus. The benefit of using I2C communication is it can connect multiple master devices to connect to one or multiple slave devices. This is useful as it allows multiple microcontrollers to log their data to one memory card [12]. I2C is synchronous, this means the output of bits is synchronized to the clock signal shared between the master and the slave. The clock signal is always controlled by the master.

Once serial communication with the device was established, the next step involved enabling network connectivity to transmit sensor data in real time. A WebSocket API was implemented to facilitate this communication. WebSockets provide a full-duplex protocol over a TCP connection, allowing continuous, event-driven data exchange between the device (client) and the server without the need for repetitive polling. This approach was selected over traditional HTTP polling due to its higher efficiency, reduced overhead, and suitability for continuous sensor-based IoT applications.

Figure 3 illustrates the system architecture and overall network topology. Sensor measurements are transmitted from the ESP32 acting as the gateway, which connects to the local wireless access point to reach the internet. The server receives the transmitted data and presents the processed sensor readings on a web-based dashboard for remote monitoring. The ESP32 also hosts the server internally, utilising its onboard flash storage to retain sensor values. Through the Arduino IDE, data are written and retrieved using the SPIFFS file system, ensuring persistence across device restarts.

3.4. Software Development

The software used to program these devices to communicate was JavaScript, html, and Arduino IDE. Arduino IDE is used to communicate with the modules I have on the bread board. The main components to be programed were the modules which include the ESP32 functionality, the sensor readings and WebSocket Server. The ESP32 acts as the server for this project so the client can connect remotely. The configuration would have to establish digital pins for the temp/humidity sensor and the buzzer. The ESP32 is able to connect to an AP by use of the Wi-Fi library included in the Arduino IDE. Like a regular wireless AP, the code in the Arduino IDE needs to include the SSID and password

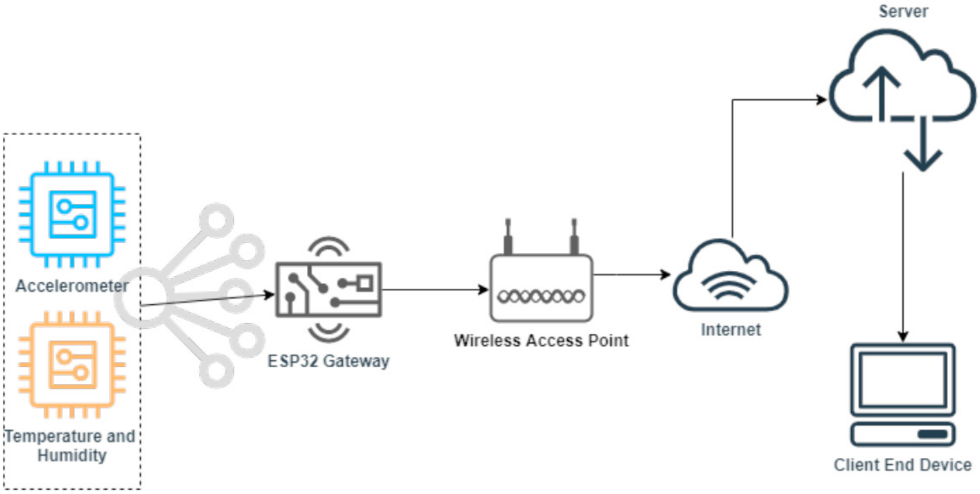


Fig. 3. system block diagram.

in order to access the AP and gain access to the internet, a port to listen on and transfer the data also needs to be set and it can be any number from port 80 upwards.

Because the ESP32 functions as both a gateway and a server, its assigned IP address serves as the access point for establishing a WebSocket connection over the local network to retrieve sensor readings. With each new sensor update, the webpage automatically refreshes the displayed values, as WebSockets operate using an event-driven communication model. To optimise efficiency, conditional checks were implemented in the Arduino code to compare incoming sensor values with previously transmitted data. This prevents redundant transmissions, reducing unnecessary network traffic and avoiding excessive webpage updates that could disrupt interface performance.

The ESP32 transmits sensor data via WebSockets using JSON formatting. JSON is a lightweight, human-readable data structure composed of key–value pairs. For example, a temperature reading may be represented as: *”temp”: ”26.7”*

This structure allows the receiving application to retrieve the corresponding value by referencing the key. JSON also supports arrays and mixed data types, including Booleans, numbers, strings, and nested objects. In this project, a JSON string is generated and transmitted each time new sensor data are captured, with each sensor variable assigned to its associated key.

On the client side, JavaScript processes the incoming WebSocket messages from the ESP32, extracts the temperature, humidity, and accelerometer values, and updates the webpage accordingly. Additional functionality is provided through integration with a weather API, which retrieves monthly average temperature data from the MetService API [13]. The HTML interface then displays both the live sensor readings and the external weather data on a dashboard for real-time monitoring.

3.5. Prototype and Experiemntal Setup

The breadboard wiring integrates two digital pins, the SDA and SCL lines for I²C communication, and the 3.3 V and ground rails for power distribution. A piezo buzzer is connected to a dedicated digital pin, with the positive terminal linked to the pin and the negative terminal grounded; the pin is programmed to switch high upon fall detection and low otherwise. The DHT22 sensor is connected to a second digital pin, transmitting temperature and humidity data directly to the ESP32.

The accelerometer’s SDA and SCL pins are connected to the corresponding ESP32 I²C pins, along with power and ground. The CS (chip select) pin is tied high to ensure proper operation, as the accelerometer does not enter a valid default mode if left floating, preventing communication with the ESP32. The complete wiring configuration is illustrated in Figure 4.

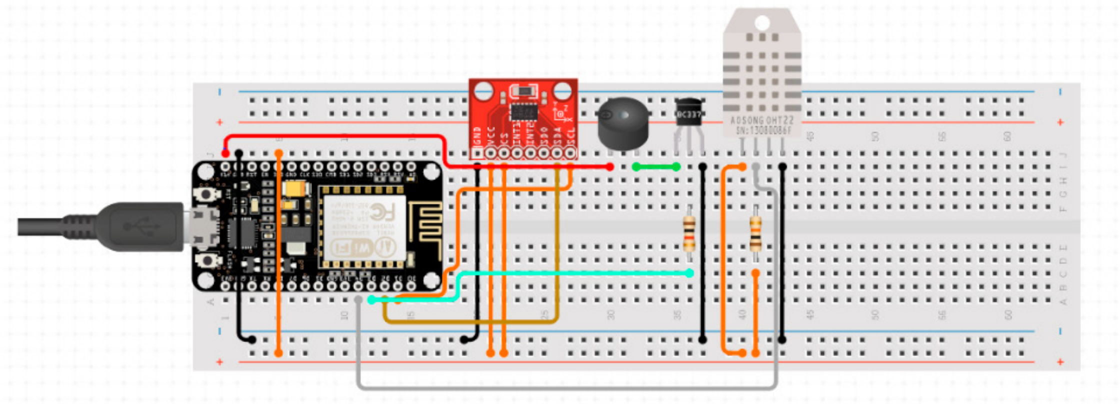


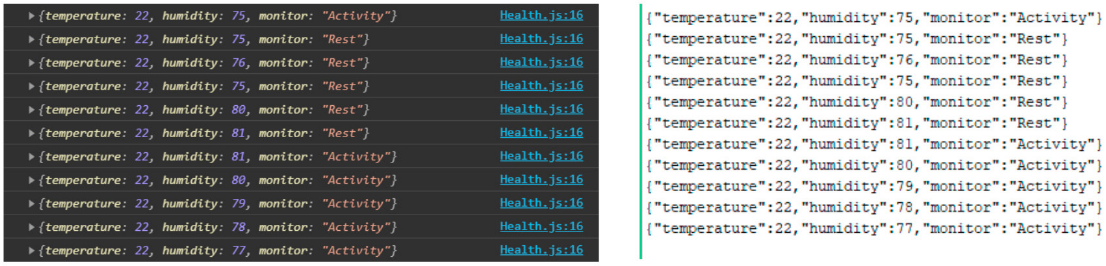
Fig. 4. Prototype wiring diagram.

4. Experimental results

Fall detection was evaluated by dropping the accelerometer module from various heights to determine an appropriate sensitivity threshold. A free-fall threshold value of 7 and a duration of 50 ms provided reliable detection—minimising false triggers from hand movements while accurately identifying actual drops. Temperature and humidity measurements were verified using the DHT22 sensor by directing warm, moist breath toward the device, resulting in expected increases in humidity (≈ 99 %) and a slight temperature rise.

Sensor outputs were first validated via the Arduino serial monitor before being transmitted wirelessly. The ESP32 was connected to a mobile hotspot, and successful network connection was confirmed through IP address assignment. A WebSocket server running on port 82 enabled real-time transmission of sensor data in JSON format. A corresponding JavaScript client, connected to the same network, parsed these JSON values as shown in Figure 5 and updated a web-based dashboard interface as in Figure 6.

The final website displayed real-time temperature, humidity, and accelerometer data, along with a fall-detection alert system that triggers an on-screen warning illustrated in Figure 7. Long-term events and scheduled tasks were also logged on the server. To verify wireless operation, the system was powered using a portable power bank, confirming that all data were transmitted over Wi-Fi rather than through a physical serial connection.



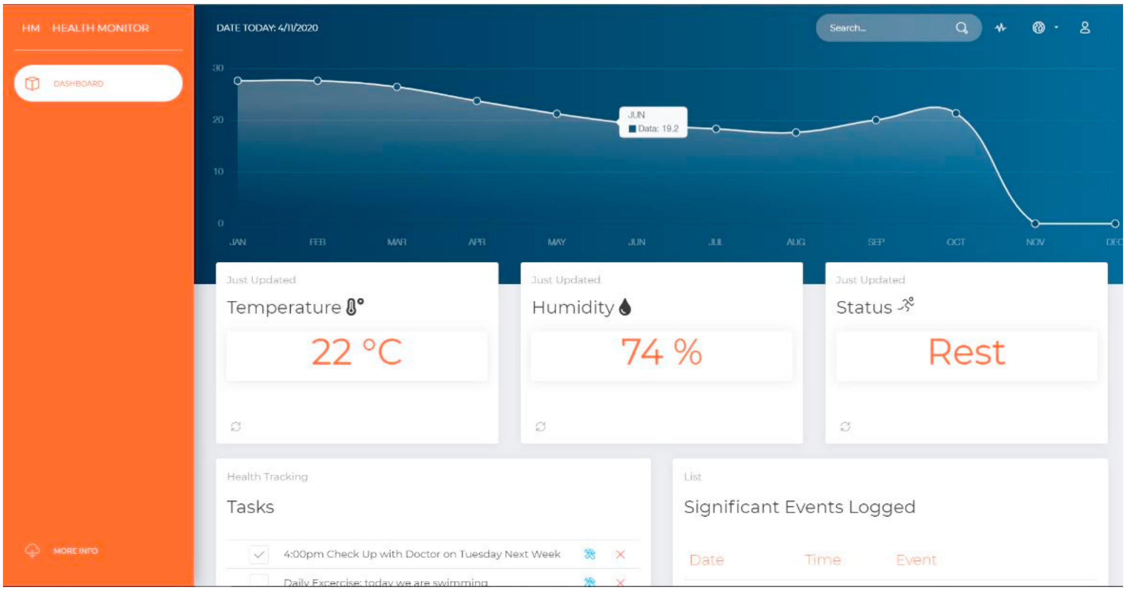
```
▸{temperature: 22, humidity: 75, monitor: "Activity"}    Health.js:16
▸{temperature: 22, humidity: 75, monitor: "Rest"}        Health.js:16
▸{temperature: 22, humidity: 76, monitor: "Rest"}        Health.js:16
▸{temperature: 22, humidity: 75, monitor: "Rest"}        Health.js:16
▸{temperature: 22, humidity: 80, monitor: "Rest"}        Health.js:16
▸{temperature: 22, humidity: 81, monitor: "Rest"}        Health.js:16
▸{temperature: 22, humidity: 81, monitor: "Activity"}    Health.js:16
▸{temperature: 22, humidity: 80, monitor: "Activity"}    Health.js:16
▸{temperature: 22, humidity: 79, monitor: "Activity"}    Health.js:16
▸{temperature: 22, humidity: 78, monitor: "Activity"}    Health.js:16
▸{temperature: 22, humidity: 77, monitor: "Activity"}    Health.js:16
```

```
{"temperature":22,"humidity":75,"monitor":"Activity"}
{"temperature":22,"humidity":75,"monitor":"Rest"}
{"temperature":22,"humidity":76,"monitor":"Rest"}
{"temperature":22,"humidity":75,"monitor":"Rest"}
{"temperature":22,"humidity":80,"monitor":"Rest"}
{"temperature":22,"humidity":81,"monitor":"Rest"}
{"temperature":22,"humidity":81,"monitor":"Activity"}
{"temperature":22,"humidity":80,"monitor":"Activity"}
{"temperature":22,"humidity":79,"monitor":"Activity"}
{"temperature":22,"humidity":78,"monitor":"Activity"}
{"temperature":22,"humidity":77,"monitor":"Activity"}
```

Fig. 5. Prototype wiring diagram.

5. Discussion

The practical and theoretical work done to achieve the goal of this project which was to create an IoT health monitoring system was a success. Throughout the development of this project's objectives are met in proving the device and network design, functionality and style works together to create a health monitor device for elderly care.



Just Updated

Temperature

22 °C

Just Updated

Humidity

74 %

Just Updated

Status

Rest

Health Tracking

Tasks

☑ 4:00pm Check Up with Doctor on Tuesday Next Week

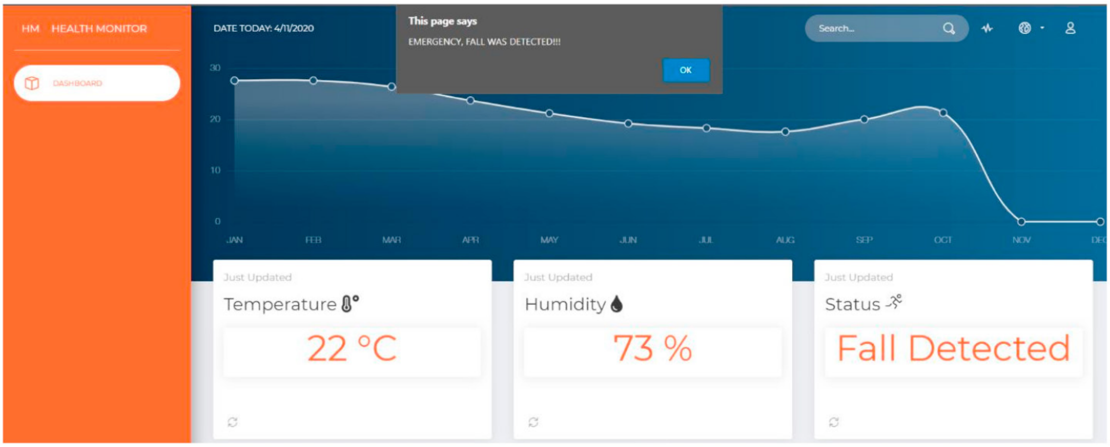
☐ Daily Excercise: today we are swimming

List

Significant Events Logged

Date	Time	Event

Fig. 6. Prototype wiring diagram.



This page says

EMERGENCY, FALL WAS DETECTED!!!

OK

Just Updated

Temperature

22 °C

Just Updated

Humidity

73 %

Just Updated

Status

Fall Detected

Fig. 7. Prototype wiring diagram.

The electronic composition and engineering work of wiring the sensors and network engineering to get the electronics to communicate over the internet was also a success. The work with the ESP32 Wi-Fi and Bluetooth module being able to connect to a website html through a WebSocket connection was a success and it was well implemented and coded. The device coding for the ADXL345, DHT22 sensors worked well and operated not only to send data every second but only send when the values change which are implemented when writing the code to increase efficiency for the operation of the device. The website functionality and overall appearance looks aesthetically pleasing to the eye rather that just a screen with values on it.

6. Conclusion

Fall-detection and health-monitoring technologies can significantly improve the care of elderly individuals who are at higher risk of falling, enabling faster and more effective emergency response. This project has developed a networked prototype that demonstrates the core concepts required for future research in IoT wearable health technologies. The electronic design, system architecture, and operational workflow to meet the project's objectives were outlined. A small network was implemented, allowing sensor data to be accessed online through a web interface using a WebSocket API. A a dashboard that displays real-time temperature, humidity, and accelerometer readings, and provides an immediate alert when a fall is detected is designed and programed. As this work served primarily as a proof of concept, the next stage would involve developing an actual wearable form factor and integrating a cloud-based server for remote data access from any location with internet connectivity. The system design also has potential industrial applications, such as monitoring package handling during shipping by detecting drops or impacts. For future health-focused enhancements, additional biosensors—such as a heart-rate sensor—could be incorporated to expand the device's remote patient-monitoring capabilities.

References

[1] John Dian, F., Vahidnia, R., & Rahmati, A. (2020). Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey. IEEE Access, 8, 69200–69211. https://doi.org/10.1109/ACCESS.2020.2986329

[2] Huang, P.-C., Lin, C.-C., Wang, Y.-H., & Hsieh, H.-J. (2019). Development of Health Care System Based on Wearable Devices. 2019 Prognostics and System Health Management Conference (PHM-Paris), 249–252. https://doi.org/10.1109/PHM-Paris.2019.00049

[3] Allison, C. (2025, October 30). 55 Apple Watch tips, hacks and cool features. Retrieved from Wareable: https://www.wareable.com/apple/apple-watch-tips-and-tricks-iwatch-7383

[4] Meskó, B., Drobni, Z., Bényei, É., Gergely, B., & Győrffy, Z. (2017). Digital health is a cultural transformation of traditional healthcare. mHealth, 3, 38. https://doi.org/10.21037/mhealth.2017.08.07

[5] León, M. A., Pannunzio, V., & Kleinsmann, M. (2022). The Impact of Perioperative Remote Patient Monitoring on Clinical Staff Workflows: Scoping Review. JMIR Hum Factors, 9(2), e37204. https://doi.org/10.2196/37204

[6] Binci, D., Palozzi, G., & Scafarto, F. (2021). Toward digital transformation in healthcare: A framework for remote monitoring adoption. The TQM Journal, 34(6), 1772–1799. https://doi.org/10.1108/TQM-04-2021-0109

[7] Vudathaneni, V. K. P., Lanke, R. B., Mudaliyar, M. C., Movva, K. V., Kalluri, L. M., Boyapati, R., krishna prasad Vudathaneni, V., & others. (2024). The impact of telemedicine and remote patient monitoring on healthcare delivery: A comprehensive evaluation. Cureus, 16(3).

[8] Moreland B, Kakara R, Henry A. Trends in Nonfatal Falls and Fall-Related Injuries Among Adults Aged ≥ 65 Years — United States, 2012–2018. MMWR Morb Mortal Wkly Rep 2020;69:875–881. DOI: http://dx.doi.org/10.15585/mmwr.mm6927a5

[9] Superseniors. (2016). Our Ageing population. Retrieved from superseniors: http://www.superseniors.msd.govt.nz/about-superseniors/media/key-statistics.html

[10] Stats NZ (2022). Retrieved from One million people aged 65+ by 2028: https://www.stats.govt.nz/news/one-million-people-aged-65-by-2028/

[11] Jia, N. (2009, July). Detecting Human Falls with a 3-Axis Digital Accelerometer. Retrieved (December 9, 2025) from Analog Dialogue: https://www.analog.com/en/analog-dialogue/articles/detecting-falls-3-axis- digital-accelerometer.html

[12] Campbell, S. (2025). BASICS OF THE I2C COMMUNICATION PROTOCOL. Retrieved from circuitbasics: https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

[13] MetService. (2020). Coded Data. Retrieved (December 9, 2025) from https://www.metservice.com/maps-radar/coded-data