



**HAL**  
open science

# Hardware/Software Co-Design of a Low-Power IoT Fall Detection Device

Dimitrios Karagiannis, Ilias Maglogiannis, Konstantina S. Nikita, Panayiotis Tsanakas

► **To cite this version:**

Dimitrios Karagiannis, Ilias Maglogiannis, Konstantina S. Nikita, Panayiotis Tsanakas. Hardware/Software Co-Design of a Low-Power IoT Fall Detection Device. 4th IFIP International Internet of Things Conference (IFIPIoT), Nov 2021, Virtual, Netherlands. pp.146-159, 10.1007/978-3-030-96466-5\_10 . hal-04471540

**HAL Id: hal-04471540**

**<https://inria.hal.science/hal-04471540v1>**

Submitted on 21 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Hardware/Software Co-Design of a Low-Power IoT Fall Detection Device

Dimitrios Karagiannis<sup>1</sup>✉, Ilias Maglogiannis<sup>2</sup>, Konstantina S. Nikita<sup>1</sup>, Panayiotis Tsanakas<sup>1</sup>

<sup>1</sup> School of Electrical and Computer Engineering

National Technical University of Athens, Zografou, Greece

<sup>2</sup> Department of Digital Systems, University of Piraeus, Piraeus, Greece

dkaragiannis@biosim.ntua.gr

**Abstract.** Falls can cause severe injuries to elder people. Many studies have been conducted in order to develop devices that improve the recognition of fall events, while very few have demonstrated approaches for the minimization of their power consumption. We propose a hardware/software co-design of an Internet of Things (IoT) fall detection device that takes advantage of accelerometer's embedded functionalities and enables its interrupt-driven operation while the rest of the circuit is in shutdown mode. In contrast with most low power fall detection devices, the proposed device supports Wi-Fi connectivity that enables ubiquitous use and real-time remote fall events monitoring through cloud services, without the need of external equipment or a local server. The device integrates a vibration motor and a cancelation button, which notify the detection of a fall event and waits for Patient's response, while the implementation of battery monitoring mechanism enables continuous device use.

**Keywords:** fall detection, low power, Internet of Things, IoT, hardware/software co-design, accelerometer

## 1 Introduction

Falls are a major issue for elderly people as they are very common and their prevention can reduce the admissions to nursing homes [1]. Moreover, falls due to seizures and nonepileptic events in an epilepsy monitoring unit (EMU) especially inside bathrooms can harm patients and solutions such as a ceiling lift system has demonstrated significant improvement [2]. Devices that can prevent or detect falls have been developed during the past years with different approaches and improvements concerning the used connectivity protocols with the server; the power consumption management; or the integration of machine learning techniques.

Some approaches for fall detection integrate the use of more sensors than just an accelerometer. In [3–5] researchers proposed systems using barometric pressure

measurements, and T. De Quadros et al. [6] designed a wrist worn fall detection device using gyroscope and magnetometer, in addition to accelerometer. MPU6050 sensor has also been used, as it can provide both accelerometer and gyroscope measurements [7, 8].

Fall detection devices have been tested in different places, like aforementioned wrist or chest and with different communication technologies such as Bluetooth, Zigbee or Wi-Fi. W. Zhuang et al. [9] have developed a smart button system placed on the chest that collects accelerometer data and send them over Bluetooth low energy to an Android mobile. In [10] a microcontroller establishes Wi-Fi connection with the server and reports fall events and battery status. Yuan et al. [11] proposed a wrist-worn wearable as a ZigBee End Device that connects to a base station in order to communicate with cellular networks. A ZigBee server can further process sensors data in order to evaluate the fall detection [7, 8].

Fall detection devices should be easy to carry and operate for many hours without the need of recharging, as it can interrupt continuous monitoring, and discourage their use. Researchers have investigated various techniques in order to lower power consumption and provide a long-lasting battery operation [12], usually applying interrupt-driven algorithms [4, 8]. In [11] a wrist-worn wearable device utilizes interrupt signals generated by the accelerometer, that lead to less data processing and, therefore, lower power consumption. J. He et al. [7] developed an interrupt-driven device that is placed in the back neck area of a wearable vest and can operate continuously for more than 30 hours with a 600 mAh battery. C. Wang et al. [5] reduced the power consumption using both hardware- and firmware-based approaches, enabling an estimated battery life (450 mAh capacity) of 664.9 usage days with an average current of 28.2  $\mu$ A, using the ADXL362 accelerometer that is more power efficient compared with the ADXL345, which is frequently used in fall detection devices. In order to lower power consumption, an algorithm that changes sampling rate dynamically, according to fall indication, has been proposed [3].

Fall events can be evaluated with different techniques, like processing the sensors data with a Kalman Filter and kNN Algorithm [7], using a SVM classification algorithm [9], or using a Fall Detection Convolutional Neural Network (FD-CNN) [8]. Machine learning methods demonstrated advantages compared to threshold-based [6].

Post Fall actions are important for immediate response to patient's condition. In [13] the accelerometer is connected with a microcontroller and a module that provides GPS/GSM service in order to send SMS in case of a fall. The device will vibrate and the patient can withdraw the alarm with a button. In [14] a hardware fall detection system has been designed that can enable an alarm and activate a mechanism in order to protect the patient. There are also commercial products with cellular connection capabilities that include fall detection and many other features - not essential for elder people - but the battery life is limited [15].

We propose an Internet of Things (IoT) fall detection device that is placed on the chest using an innovative co-design of hardware and software, to achieve low power consumption. Whereas many aforementioned devices are low power, they use a ZigBee network that requires a local server to be fully functional. Our proposed device supports serverless operation using Wi-Fi, which is broadly available and doesn't require any

external equipment for most of the patients. Newer mobile phones enable Wi-Fi hotspot using the existing cellular connectivity, which provides ubiquitous internet access. Although Wi-Fi communication is useful, it requires high power consumption that makes it not suitable for a fall detection device. In order to solve this issue, we propose a hardware architecture that enables the operation only of the accelerometer, where the rest of the circuit is in shutdown mode consuming negligible energy. Our fall detection device has very low power consumption during patient's every day activities, but in case of a fall event it turns into high power mode. Moreover, we integrated battery undervoltage detection and protection mechanisms in order to ensure continuous device operation. The device incorporates a cancellation mechanism, through a button and a vibration motor, that notifies the patient in case of a fall event detection. Although patient's immobility after a fall is usually verified with accelerometer measurements, instead in the proposed device a Cancellation button is used, which can help the detection of cases where the patient is moving after fall, because of pain and injuries or epileptic seizures.

This paper is organized as follows: Section 2 describes the Fall Detection basic principles and Section 3 demonstrates the platform architecture, the hardware of the proposed device and the algorithm it utilizes during operation and after a fall event. Section 4 evaluates the device's power consumption and Section 5 is the Discussion, where the Conclusion is in Section 6.

## 2 Fall Detection Basic Principles

For our experiment we used ADXL345, an ultralow power accelerometer from Analog Devices that can provide 3-axis acceleration with up to 13 bit resolution and  $\pm 16$  g range [16]. ADXL345 can map different functions to two interrupt pins (INT1 and INT2) with user defined thresholds: single and double tap; free-fall; activity and inactivity sensing. Interrupts enable the fall detection without any processing by the main microcontroller, as the acceleration data are processed by the accelerometer separately.

Free fall interrupt has two user defined thresholds: THRESH\_FF and TIME\_FF and is set when acceleration for all axes is less than the value stored in THRESH\_FF register for more time than the value stored in TIME\_FF register [16]. The first phase of a Fall is always the "Weightlessness" where the vector sum of the accelerations from all axes tends to 0 g, in contrast with all other motions like sitting or walking [17]. Our system takes advantage of the "Weightlessness" phase, which is essential for fall, and activates the Free Fall interrupt.

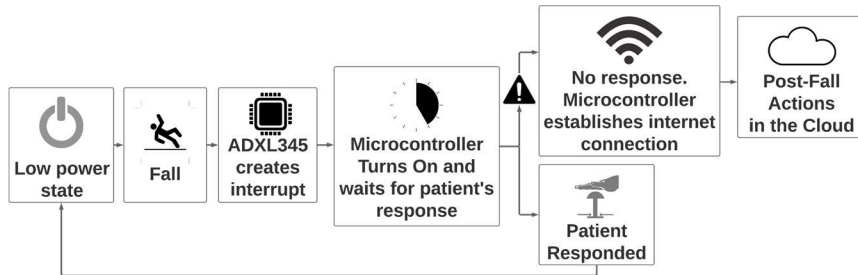


Fig. 1. Platform Overview <sup>1</sup>

### 3 Platform Architecture

#### 3.1 Platform Overview

The fall detection device operates in an ultralow power consumption state where the microcontroller and its peripherals are in shutdown mode. ADXL345 operates normally until Patient's fall. Then it activates the FREE\_FALL interrupt that turns on the microcontroller and the Vibration motor, in order to notify Patient that he/she has to interact and press a button that is integrated on the device. If there is not a Patient response, the microcontroller connects through Wi-Fi to Cloud services and notifies the Physician about the emergency. If the Patient presses the button the device returns to the ultralow power state. (Fig. 1)

#### 3.2 Device Hardware Architecture

The device consists of a 3D printed enclosure and a custom PCB with: (i) an accelerometer, (ii) a microcontroller, (iii) two Voltage Regulators, (iv) two Voltage Detectors, (v) a Vibration Motor module, (vi) a Button for interaction with the Patient, (vii) other peripherals (LED, buttons) for microcontroller programming and debugging. The hardware is divided in four sections with different functionalities (A-D) that will be analyzed separately. (Fig. 2)

<sup>1</sup> Fig. 1, 2, 4, 5, 7 created in Lucidchart. Available: [www.lucidchart.com](http://www.lucidchart.com)

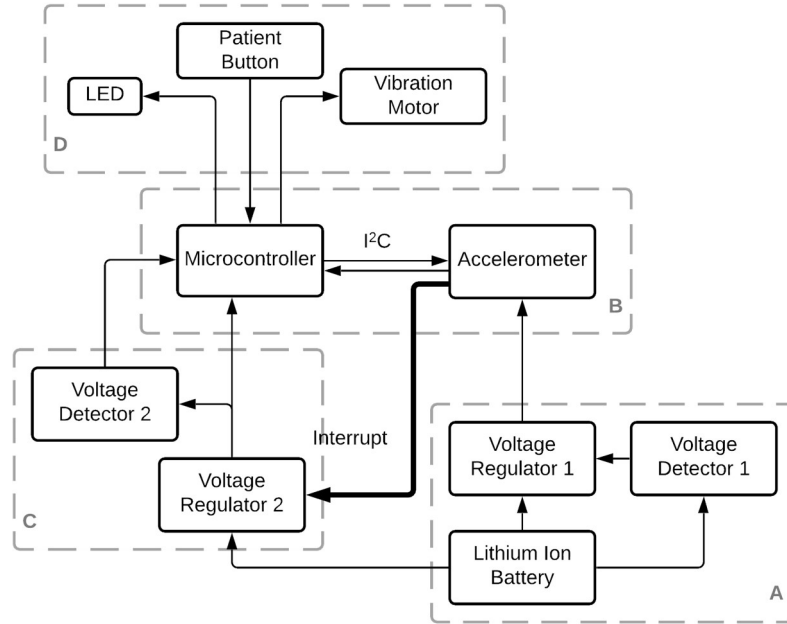


Fig. 2. Simplified Device Hardware Architecture

**Accelerometer powering circuit and battery undervoltage protection (A).** The power source is a high-density rechargeable Lithium-Ion battery with 3400 mAh capacity, 3.6V nominal and 4.2V maximum voltage. The accelerometer we used is ADXL345 that operates from 2.0V to 3.6V [16]. In order to regulate the Battery voltage, as Voltage Regulator 1 the TPS70633DRVR is selected, because it has ultra-low quiescent current (1 $\mu$ A) and enable (EN) pin that controls its shutdown or active state. Although we use Lithium-Ion battery that has integrated protection circuit with cutoff voltage at 2.5V, we added one more protection layer. As Voltage Detector 1 we selected STM1061N26WX6F that monitors battery voltage and in case it falls below 2.6V it outputs Low. This output is connected with enable (EN) pin of Voltage Regulator 1 and (when it is Low) the regulator will turn into shutdown mode. Moreover, we added capacitors that add stability and better performance to the regulator. Section A (Fig. 2) of the circuit is used only for the accelerometer's operation.

**Accelerometer and Microcontroller I<sup>2</sup>C communication (B).** ESP32-WROOM-32 is a low-cost microcontroller module that supports Wi-Fi, Bluetooth, BLE connectivity and operates at  $\sim 3.3\text{V}$ . It communicates with ADXL345 accelerometer with I<sup>2</sup>C protocol using the necessary pull-up resistors and 2 Pins (SDA, SCL). ADXL345 interrupt pin 1 (INT1) that is used as the FREE\_FALL interrupt is not connected with ESP32.

**Microcontroller powering circuit and battery voltage monitoring (C).** The aforementioned low-dropout Voltage Regulator 1 (TPS70633DRVR) powers only ADXL345 accelerometer but not the ESP32-WROOM-32 (ESP32) module. It can provide max current of 150mA, where ESP32 module needs minimum 500mA. As Voltage Regulator 2, TLV75533PDBVR is selected because it can provide 500mA to ESP32 and has low dropout voltage that enables stable 3.3V output even when the battery voltage is decreasing. Also, it supports enable (EN) pin that controls its operational or shutdown state. ADXL345 FREE\_FALL interrupt (INT1) is connected to EN pin of Voltage Regulator 2 (bold line in Fig. 2). When INT1 is Low (i.e. no fall event), Voltage Regulator 2 is in shutdown mode and everything that is powered from it, is turned off. Thus, we succeed an ultralow power consumption under usual circumstances.

After fall detection, accelerometer's FREE\_FALL interrupt (INT1) is activated and becomes High, as enable (EN) pin of Voltage Regulator 2. Thus, Voltage Regulator 2 turns on and powers the ESP32 module in order to run the fall detection algorithm. At the end of ESP32 actions, through I2C communication, ESP32 clears ADXL345 INT1 interrupt after reading INT\_SOURCE register, that leads to the shutdown of Voltage Regulator 2 and the return to the previous operation.

Voltage Detector 2 (STM1061N31W6F) monitors the output of Voltage Regulator 2 and in case the Voltage is under 3.1V, it sets the connected Pin of ESP32 Low. Every time ESP32 is turned on, this pin is checked. If it is Low, battery low voltage indication is sent to the Cloud, so Patient will be notified by his/her Physician (who monitors the Cloud) in order to charge the battery and be protected without stopping the functional fall detection device. ESP32 operation will not stop immediately if the battery is below 3.1V, but after becoming lower than 2.6V (as described in section A). Though, it is recommended to immediately recharge the battery, as voltage below 3.0V is not recommended for ESP32 operation [18].

**Interaction components (LED, vibration motor and button) (D).** We added a Vibration Motor module, a LED, and a tactile switch as a Cancellation button. In case of a fall detection, the LED turns on and the motor notifies the Patient so he/she can press the button and deactivate the emergency algorithm. Finally, a SPDT switch, two extra tactile buttons and pin headers are added for easier programming and debugging of the circuit.



### 3.3 3D printed enclosure

The designed hardware and a Li-ion battery of 3400 mAh with integrated protection circuit are placed inside an enclosure that protects and stabilizes the circuit. Also, it provides an easy way to use a cord and hang it around the Patient's neck. The enclosure is 3D printed with PLA filament. (Fig. 3)

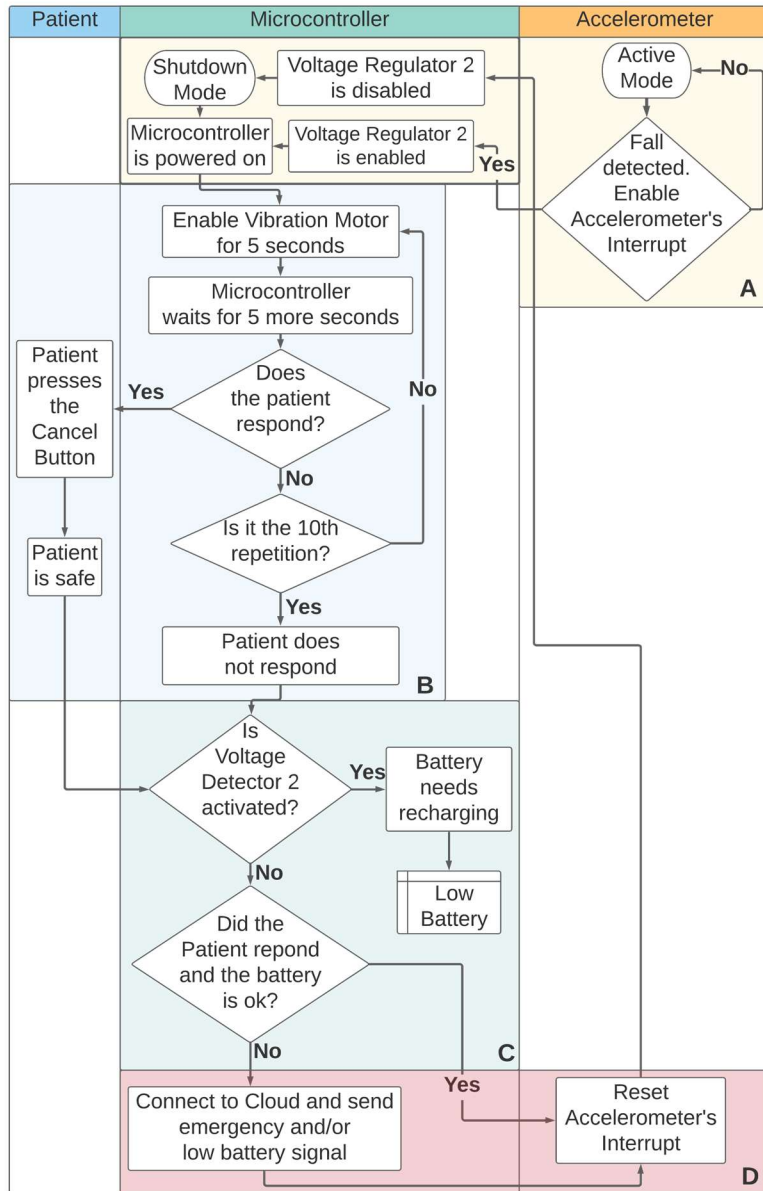


**Fig. 3.** 3D Printed Enclosure, 18650 Li-ion battery and hardware of the fall detection device. The areas with the red frame are used for easier programming of the device but they can be skipped, if we would like to minimize the PCB size.

### 3.4 Device Operation Algorithm

Fig. 4 demonstrates the Fall event confirmation from the moment of initial detection until the emergency signal that is sent to the Cloud:

- (A) The microcontroller (ESP32) and its peripherals are in shutdown mode without consuming power. The accelerometer on the other hand, is in Active mode, operates normally and the FREE\_FALL interrupt parameters have been set during the first device operation. During Patient's daily activities, only the accelerometer is running and the rest of the system is waiting for the FREE\_FALL interrupt. After fall detection, the interrupt enables Voltage Regulator 2 that is responsible for ESP32 powering.
- (B) Fall has already been detected, so ESP32 waits for Patient's response. We have set Vibration for 5 seconds, followed by 5 seconds of ESP32 inaction. The Pin connected to the Cancellation Button has already been set as an interrupt pin that can stop Vibration Motor or ESP32 inaction immediately after it is pressed. Vibration reminds Patient that the fall detection algorithm has been activated and a button press is required in order to cancel the emergency. If the Patient does not respond after 10 consecutive cycles of vibration - waiting, ESP32 will proceed to an emergency signal. Even if ADXL345 accelerometer supports inactivity interruption or we can read its acceleration measurements that should indicate Patient's immobility, we use the Cancellation button. Thus, the fall detection algorithm will not be confused by a Patient that has fallen and is moving but is injured and unable to stand up or call for help.

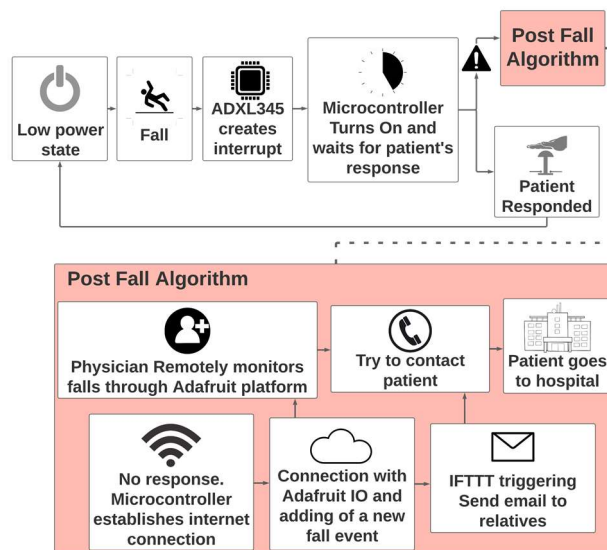


**Fig. 4.** Fall detection algorithm

- (C) If the battery is exhausted - and therefore Voltage Detector 2 is activated - ESP32 updates a variable that will be used during Cloud connection in order to notify the Physician.
- (D) Patient has not responded by pressing the Cancellation button for 100 seconds (5 seconds of Vibration and 5 seconds of waiting for 10 times), thus ESP32 will establish a Wi-Fi connection to the Cloud services where it reports the fall. If the Patient has pressed the Cancellation button, and the Battery is not low, ESP32 resets FREE\_FALL interrupt, which in turn disables Voltage Regulator 2 and turns off ESP32. ADXL345 accelerometer continues to operate, in order to detect another possible fall. In case of low battery voltage, even if the Patient responded, ESP32 will connect to the Cloud to update the battery level indication.

### 3.5 Post-Fall Algorithm

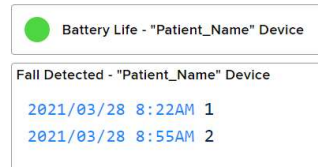
In the previous section we demonstrated the fall detection algorithm that confirms or discards the interrupt signal caused by the accelerometer. After confirmation - and/or battery's low voltage – the microcontroller (ESP32) establishes Wi-Fi connection in order to proceed to the post fall algorithm (Fig. 5). During ESP32 programming we have defined credentials (SSID and password) for Patient's home router and mobile phone Wi-Fi hotspot, which will enable mobile data sharing. Wi-Fi hotspot will provide ubiquitous internet connection even outside of Patient's home.



**Fig. 5.** After fall confirmation, ESP32 proceeds to the Post Fall Algorithm and the connection to Cloud Services.

ESP32 starts by trying to establish connection with the Patient's home router and in case it does not succeed, it tries to connect with the mobile phone's Wi-Fi hotspot.

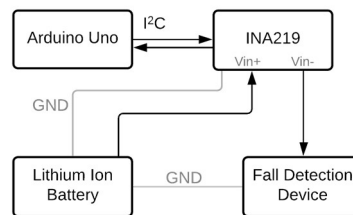
Then, it connects to Adafruit IO, a Cloud service that enables data storing and visualization [19]. In Adafruit IO we created two feeds: one for fall detection and one for battery low voltage indication. These feeds are connected and visualized to a simplified Dashboard that updates in real-time, which the Physician can remotely monitor through the Adafruit IO platform (Fig. 6). Using IFTTT Gmail service, we created a trigger that monitors Adafruit IO data feeds and whenever a fall is recorded, an email is sent to up to 20 recipients. Thus, not only the Physician can remotely and in real-time monitor the Patient, but Patient's relatives also will be alerted in an emergency case. IFTTT ("If This Then That") is a trigger-action programming model that has gained popularity [20] and can be used to further expand our platform with more emergency actions after fall event.



**Fig. 6.** If the battery voltage is lower than the cutoff, the green indicator turns red. The second block monitors Fall Detections, where value "1" indicates that the Patient's device is connected through the home network, "2" through the mobile phone, and "3" that the Patient is safe (the Cancellation button was pressed) but the device was connected to the platform to report that the battery is low.

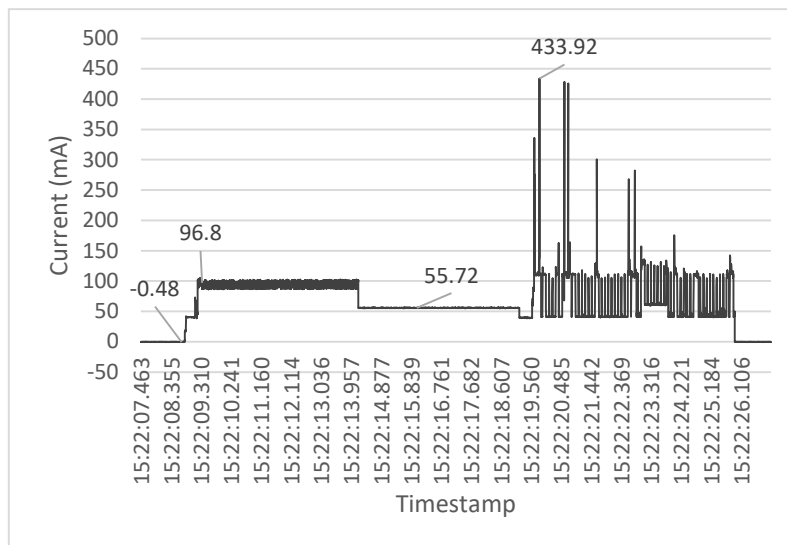
## 4 Evaluation

In order to evaluate the power consumption of the device, we simulated a fall event and measured the current with an INA219 current monitor. INA219 was placed in series with the circuit at the high side of voltage (at battery's positive terminal) as demonstrated in Fig. 7. An Arduino UNO communicates with I<sup>2</sup>C protocol with INA219 and logs power consumption measurements (Fig. 8, 9).

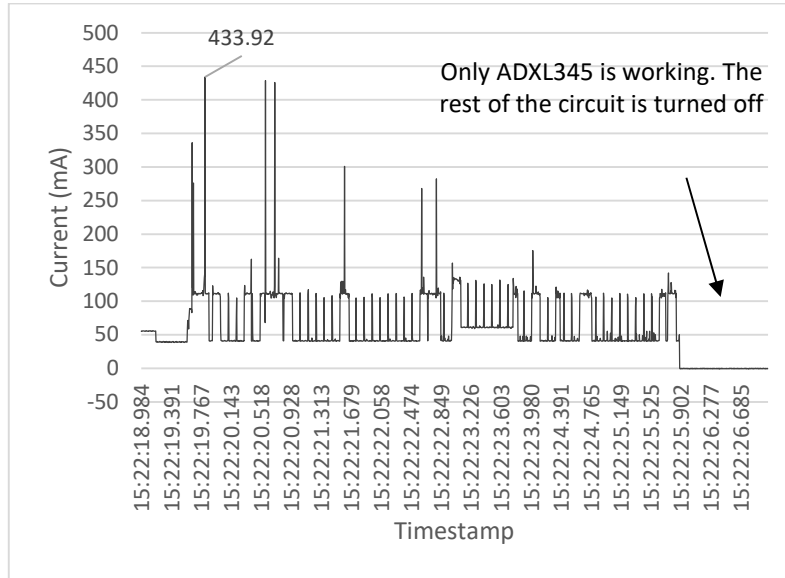


**Fig. 7.** Experimental Design used for the fall detection device power consumption evaluation, before, during and after the fall event.

Fig. 8 and 9 demonstrate the current measurements using INA219 during a fall event simulation. At the beginning (before the fall) the values are around zero or negative because the current is very low and INA219 is unable to measure it, so we used a multimeter. The measured current was  $\sim 50.5 \mu\text{A}$ . In order to accomplish this current we set ADXL345 to low power mode and the Output Data Rate to 25Hz, which is achieved by decimating a common sampling frequency [16]. FREE\_FALL interrupt is not affected by lower Output Data Rates because its operation is based on undecimated data. Using a 3400 mAh battery in an ideal scenario where the Patient has no falls at all and there are no interruptions generated from the accelerometer, the device can operate for around 2805 days or 7.69 years without recharging [21]. The current during vibration is  $\sim 100\text{mA}$  and during 5 seconds of waiting is  $\sim 55\text{mA}$ . For easier demonstration, instead of 10 cycles of vibration-waiting, according to the presented fall detection algorithm, we programmed one cycle for the simulation. During Wi-Fi connection the peak current is 433.92mA, which is multiple times the low power state current, so the connection with the Cloud Services occurs only in emergency circumstances (i.e. confirmed fall or low battery voltage). If the Patient responds during the fall detection algorithm by pressing the button and the battery level is not low, the device will return to low power state without Wi-Fi connection. Moreover, we evaluated the device response when power source voltage was less than 3.1V but more than 2.6V that led to change of green battery indication in Adafruit IO to red, and when the voltage was less than 2.6V that set the whole circuit in shutdown mode (measured current was  $\sim 1.2 \mu\text{A}$ ).



**Fig. 8.** Current measurements using INA219 during a fall event simulation. The battery voltage is 3.84V.



**Fig. 9.** Current measurements during Wi-Fi connection enlarged. After the end of the fall detection algorithm the device returns to low power state with current  $50.5\mu\text{A}$ .

## 5 Discussion

The demonstrated fall detection device enables low power consumption, supports serverless operation and doesn't require any infrastructure (other than Wi-Fi connection) or a SIM card. Even though Wi-Fi connectivity is a high-power consumption process that would quickly drain the battery, the proposed device consumes  $50.5\mu\text{A}$ , when there is no fall event. We used a 3400 mAh battery that in an ideal scenario with no fall detection at all could operate the device for 7.69 years. Even if the actual battery life is much less, the device could be easily maintained by the Patient with rare charges and provide more working hours compared to existing commercial solutions [15].

Furthermore, we incorporated a mechanism that checks battery level and protects the Patient from an unreported fall event due to an insufficiently powered device, by updating the battery status in the cloud. A second protection mechanism shuts down the whole circuit if the battery level is very low. Moreover, the integration of a Cancellation button enables the Patient to reject a false fall detection that makes the device return to the previous low power operation. In cases such as an epileptic seizure - that the Patient is moving after fall but is unable to press the Cancellation button - our device will make a more reliable detection, compared to other approaches that wait for a motionless Patient. With IFTTT we provided an easy way to add extra services, like the notification of other people that the Patient is unsafe, which is already integrated.

Although more experimental data from fall events would be useful in fine tuning the accelerometer's FREE\_FALL interrupt parameters and the fall detection algorithm that takes place after microcontroller's activation, the demonstrated low power hardware design works independently. Totally different algorithms and techniques can be applied to the proposed hardware, while the power consumption before the fall detection will be the same.

The device could be further optimized for real-world applications with size and weight reduction. Despite the fact the battery that was used provides a lot of energy, it adds a lot of extra weight and size, while the consumption is very low, thus enabling the use of a smaller one. Also, some PCB components, which demonstrated in Fig. 3, could be removed as they are used for easier development. So, a future smaller version of the device will be easier to use, hanged around the neck or embedded in the Patient's clothes, compared to a smart watch, which he/she is not used to and must wear continuously. Furthermore, there are accelerometers more power efficient than ADXL345 that could reduce the consumption additionally. Also, we can further test our device in order to detect more complex fall events, like a slow fall and improve the cancelation mechanism to include cases where the Patient pressed the Cancelation button but could not stand up afterwards (for example by repeating the vibration-waiting cycles after a few minutes). Finally, instead of just notify other people in case of a fall event using IFTTT email service, we can develop a mobile app to locate the person who is closer and make him/her responsible to visit the Patient.

## **6 Conclusion**

In this paper, we demonstrated a hardware and software co-designed fall detection device that achieves low power operation utilizing accelerometer's interrupt-based embedded functionalities, whereas the rest of the circuit is in shutdown mode, consuming negligible energy. The proposed IoT device can establish Wi-Fi connectivity to implement post-fall actions using Cloud services, without the need of external equipment or a local server, in contrary with other low-power approaches. The integration of a Cancelation button allows a Patient to declare that he/she is safe, while helps the detection of a fallen Patient who is in danger but not immobile. Mechanisms check low battery voltage or depletion, enabling continuous protection against fall events.

## **Acknowledgment**

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-02506).

## References

1. Tinetti ME, Williams CS (1997) Falls, injuries due to falls, and the risk of admission to a nursing home. *N Engl J Med* 337:1279–1284. <https://doi.org/10.1056/NEJM199710303371806>
2. Spritzer SD, Riordan KC, Berry J, et al (2015) Fall prevention and bathroom safety in the epilepsy monitoring unit. *Epilepsy Behav* 48:75–78. <https://doi.org/10.1016/j.yebeh.2015.05.026>
3. Wang C, Narayanan MR, Lord SR, et al (2014) A low-power fall detection algorithm based on triaxial acceleration and barometric pressure. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp 570–573. <https://doi.org/10.1109/EMBC.2014.6943655>
4. Wang C, Lu W, Redmond SJ, et al (2018) A Low-Power Fall Detector Balancing Sensitivity and False Alarm Rate. *IEEE J Biomed Heal Informatics* 22:1929–1937. <https://doi.org/10.1109/JBHI.2017.2778271>
5. Wang C, Lu W, Narayanan MR, et al (2016) Low-Power Fall Detector Using Triaxial Accelerometry and Barometric Pressure Sensing. *IEEE Trans Ind Informatics* 12:2302–2311. <https://doi.org/10.1109/TII.2016.2587761>
6. de Quadros T, Lazzaretti AE, Schneider FK (2018) A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device. *IEEE Sens J* 18:5082–5089. <https://doi.org/10.1109/JSEN.2018.2829815>
7. He J, Zhang Z, Yu W (2018) Interrupt-Driven Fall Detection System Realized via a Kalman Filter and kNN Algorithm. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI). pp 579–584. <https://doi.org/10.1109/SmartWorld.2018.00120>
8. He J, Zhang Z, Wang X, Yang S (2019) A Low Power Fall Sensing Technology Based on FD-CNN. *IEEE Sens J* 19:5110–5118. <https://doi.org/10.1109/JSEN.2019.2903482>
9. Zhuang W, Sun X, Zhi Y, et al (2017) A novel wearable smart button system for fall detection. In: AIP Conference Proceedings. p 020075. <https://doi.org/10.1063/1.4982440>
10. López A, Pérez D, Ferrero FJ, Postolache O (2018) A Real-Time Algorithm to Detect Falls in the Elderly. In: 2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA). pp 1–5. <https://doi.org/10.1109/MeMeA.2018.8438747>
11. Yuan J, Tan KK, Lee TH, Koh GCH (2015) Power-Efficient Interrupt-Driven Algorithms for Fall Detection and Classification of Activities of Daily Living. *IEEE Sens J* 15:1377–1387. <https://doi.org/10.1109/JSEN.2014.2357035>
12. Ren L, Peng Y (2019) Research of Fall Detection and Fall Prevention Technologies: A Systematic Review. *IEEE Access* 7:77702–77722. <https://doi.org/10.1109/ACCESS.2019.2922708>



13. Wu F, Zhao H, Zhao Y, Zhong H (2015) Development of a Wearable-Sensor-Based Fall Detection System. *Int J Telemed Appl* 2015:1–11. <https://doi.org/10.1155/2015/576364>
14. Xiuping Y, Jia-Nan L, Zuhua F (2015) Hardware Design of Fall Detection System Based on ADXL345 Sensor. In: 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA). pp 446–449. <https://doi.org/10.1109/ICICTA.2015.117>
15. Apple Watch - Battery - Apple. <https://www.apple.com/watch/battery/>. Accessed 24 Feb 2021
16. Analog Devices ADXL345 Datasheet. <https://www.analog.com/media/en/technical-documentation/datasheets/ADXL345.pdf>. Accessed 10 Jun 2021
17. Jia N (2009) Detecting Human Falls with a 3-Axis Digital Accelerometer. *Analog Dialogue* 43:3–9
18. Espressif Systems (2021) ESP32-WROOM-32 Datasheet. [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf). Accessed 10 Jun 2021
19. What is Adafruit IO? | Welcome to Adafruit IO | Adafruit Learning System. <https://learn.adafruit.com/welcome-to-adafruit-io/what-is-adafruit-io>. Accessed 24 Feb 2021
20. Ur B, Pak Yong Ho M, Brawner S, et al (2016) Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp 3227–3231. <https://doi.org/10.1145/2858036.2858556>
21. Battery Life Calculator | DigiKey Electronics. <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life>. Accessed 31 Mar 2021