IoT-Based Fall Detection System with Energy Efficient Sensor Nodes

Tuan Nguyen Gia[1], Igor Tcarenko[1], Victor K. Sarker[1], Amir M. Rahmani[1], Tomi Westerlund[1], Pasi Liljeberg[1], and Hannu Tenhunen[1,2]
[1]Department of Information Technology, University of Turku, Turku, Finland
[2]Department of Industrial and Medical Electronics, KTH Royal Institute of Technology, Stockholm, Sweden
Email: {tunggi, igotsa, vikasar, amirah, tovewe, pakrli}@utu.fi, hannu@kth.se

***Abstract*—Fall needs to be attentively considered due to its highly frequent occurrence especially with old people - up to one third of 65 and above year-old people around the world are risk of being injured due to falling. Furthermore, fall is a direct or indirect factor causing severe traumas such as brain injuries or bone fractures. However, timely medical attention might help to avoid serious consequences from a fall. A viable solution to solve this is an IoT-based system which takes advantage of wireless sensor networks, wearable devices, Fog and Cloud computing. To deliver sufficient degree of reliability, wearable devices working at the core of a fall detection system, are required to work for prolonged period of time. In this paper we investigate energy consumption of sensor nodes in an IoT-based fall detection system and present a design of a customized sensor node. In addition, we compare the customized sensor node with other sensor nodes, built on general purpose development boards. The results show that sensor nodes based on delicate customized devices are more energy efficient than the others based on general purpose devices while considering identical specification of micro-controller and memory capacity. Furthermore, our customized sensor node with energy efficiency selections can operate continuously up to 35 hours.**

***Index Terms*—Internet-of-Things, Fall Detection, Fog Computing, Energy Efficiency, Wearable Devices**

I. Introduction

Internet-of-Things (IoT) is a concept that encompasses a number of technologies, with the aim to extend Internet to real-world objects [1]. With this approach, different quantities which represent physical world, such as temperature, humidity, acceleration, just to mention a few, will be digitized. Wireless Sensor Network (WSN), plays a key role in IoT, acting as a source of data, enabling digitalization. Collected data can be further processed and analyzed by the means of fog [2] and cloud computing. IoT application area is broadening, currently including a number of domains [3], such as smart environments, personal and social fields, transportation, and logistics. The field of health-care is one of the most important areas covered. As predicted in [4], health-care will shift to home-centered fashion by 2030. Pervasive nature of IoT might significantly contribute in the change.

According to World Health Organization [5], falls are one of the dominant health-related issues among elderly. According to [6], more than one third of 65 (and above) year-old falls each year. Falls might lead to serious injuries such as head traumas or brain damages. Delay in a medical treatment in such cases threatens patient's life [7], [8]. However, only a half of the patients reports about the incident. Unreported cases might lead to dangerous health related problems. A quick response on the incident might decrease the risk of serious medical conditions after a fall.

Fall detection systems in this regard play an important role. They can be categorized into wearable and context-aware systems [9]. Latter systems feature environmental sensors such as microphones and cameras, placed around a patient to be monitored. Wearable systems estimate patient's movements by exploiting sensors attached to the body. Popular wearable solutions employ accelerometers and gyroscopes attached to a patient.

Operations in the scope of IoT, can uncover new outlook on wearable fall detection systems: computational capacities of the fog can be used to reduce the workload of wearable devices thus prolong the duration of its operational time; cloud will allow to analyze and store gathered data and notify an appropriate caregiver almost in a ubiquitous way.

In this work, we investigate primarily energy consumption sources together with a total energy consumption of wireless sensor nodes in a fall detection IoT-based system. We design a simple customized wireless sensor node and compare that with other sensor nodes based on general purpose systems in terms of energy consumption to explore ways of building energy efficient wearable sensor nodes. We demonstrate that by applying our method of customizing sensor nodes for a IoT-based fall detection system will lead to an energy efficient system design.

The remainder of the paper is organized as follows: Section 2 presents related works and motivations. Section 3 discusses system architecture and fall detection algorithm. Section 4 presents experimental setup and implementation. In section 5, experimental results are shown. Section 6 considers discussions and Section 7 concludes the work.

II. Related Work and Motivation

Reliability of a fall detection system is multifaceted. We argue that the most important ones include fall detection accuracy, operational time of the system and ability to timely deliver a notification about a fall.

Core building blocks of fall detection systems are usually embedded in wearable device accelerometers and gyroscopes, or cameras. In series of works, authors focus on improving the accuracy of fall detection system by combining several building blocks. In [10] Casilari et al. use an accelerometer and a gyroscope, embedded in a smart phone and a smart watch, in [11] authors exploit a depth camera together with an accelerometer-based wearable to increase the accuracy. Other works [12], [13] target quality (sensitivity, specificity, accuracy) of fall detection algorithms, as it sets the scope of their application. Although energy efficiency of a system directly affects duration of its work, and therefore, its reliability, it lacks due attention. A wearable device designed to fulfil the demands for both preserving patient's lifestyle and providing constant monitoring, should be capable of working for prolonged periods autonomously. This requirement makes a wearable a bottleneck of a whole system when considering duration of work.

General purpose development devices e.g. Arduino Uno, Fio are widely used as the central computational part of a sensor node of fall detection systems [14], [15]. Due to large current draw, these general purpose systems cannot be considered energy efficient. In order to improve energy efficiency, a delicate device can be utilized. In addition, the relationship between sensors' sampling rate and communication data rate is not examined in detail.

An accelerometer which provides three-dimensional acceleration values, together with the wireless transmission mechanism (to the gateway), is another source of energy drain. By a harmonious combination of accelerometer sampling rate and transmission rate, both the fall detection system's requirements and energy efficiency can be achieved. Another variable which influences energy characteristics is communication interfaces those are used in the sensor nodes and their data rates (i.e. SPI, I^2C).

To the best of our knowledge, the actual issues which limits energy efficiency when considering overall primary energy consuming source of a sensor node in the fall detection system have not been elaborately investigated. Therefore, in this paper, we design a sensor node for evaluating factors impacting on energy consumption of the sensor node in the fall detection system. These investigated factors include parameters within micro-controller (i.e. type and frequency, communication interfaces), 3D accelerometer sampling rate, Bluetooth technology (i.e. classic, low energy and data rate). In addition, we compare several sensor nodes constructed from general purpose devices with our design in terms of energy efficiency.

III. Fall detection IoT system architecture

The system architecture, shown in Fig. 1, consists of three main parts: sensor nodes, a gateway with a fog layer unit and a back-end system described as follows:

A sensor node consists of at least three primary components: 3D accelerometer sensor, micro-controller and wireless communication module. Data is gathered from the 3D accelerometer via a communication interface such as UART, SPI or I^2C. Depending on particular fall detection systems, the collected data can be pre-processed or kept intact before being transmitted to a gateway wireless. Frequently, collected raw data without pre-processing (i.e. wavelet transformation, or neural filtering) is transmitted as such. Complex processing mechanisms, like fall detection algorithms based on hidden Markov model, are carried out at the gateway because of significant computational requirements. In our fall detection system, Bluetooth is utilized for wireless communication between sensor nodes and a gateway.

Alongside with primary features of receiving data from sensor nodes and transmitting the data to Cloud servers, a gateway with fog computing in a IoT-based fall detection system can offer advanced services such as data process-

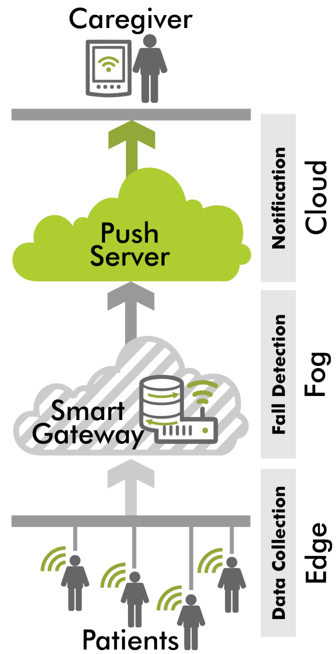

Fig. 1: The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers.

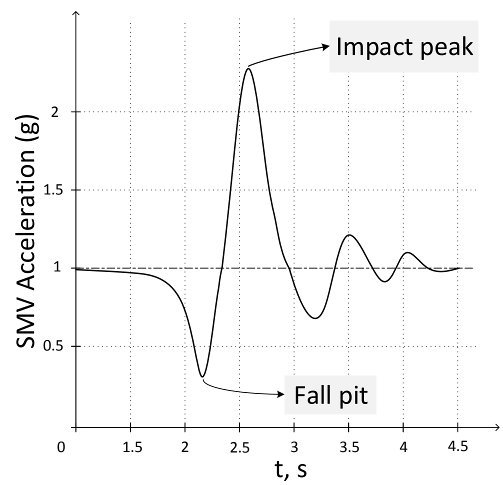

Fig. 2: Acceleration changes in time during a fall.

ing (i.e. complex filtering mechanisms, or data fusion), data compression, security, push notification service, local storage, fall detection algorithms and decision making. Depending on particular fall-detection applications, specific services might be proffered. In our fall detection system, a gateway equipped with a Bluetooth module and an Ethernet module is used for receiving raw data from sensor nodes and transmitting the data to Cloud, respectively. In addition, the gateway provides a push notification service, local storage and a fall detection mechanism. The collected data is processed with a fall detection algorithm. When a fall is detected, the gateway triggers the push notification service for notifying caregivers in real-time. Local storage is used for storing both user data for some periods of time and service data i.e. algorithms.

The back-end system includes cloud servers and an user terminal. When the Cloud receives a signal from the gateway's push notification service, it notifies appropriate party via real-time messages. An end-user (i.e. caregiver) can then view these messages with an Internet browser or a mobile application. In our system, the system checks reply messages from end-users after sending notification messages in order to verify that fall notification messages are properly received. The end-users can reply with an affirmative messages via a browser or an appropriate application to acknowledge the fall case.

IV. Experimental setup and Implementation

In order to evaluate energy consumption of a sensor node, a complete IoT-based fall detection system described in section 3 is implemented.

The gateway is implemented on Rasberry Pi v3 [16] as it is equipped with a compelling 4-core 1.2 GHz CPU, 1 GB RAM and extensible storage, which can guarantee that complex algorithms (i.e. push notification and fall-detection) run smoothly with low latencies. Additionally, Bluetooth classic, Bluetooth low energy, and Ethernet are supported. Therefore, functionality of the gateway can be conveniently implemented without needing supplementary hardware components. Ubuntu is used for managing the gateway due to its benefits such as customization, security support, users support and diversified administration tools. With Ubuntu, all tasks can be run fairly and possible hardware race condition can be avoided. Because of its performance, ease of use, scalability and secure nature, the MySQL together with local storage are used for storing 3D acceleration data, user records, and essential data used by the services.

In the gateway, a two-level threshold fall detection algorithm is applied which uses three-dimensional acceleration data to calculate fall-feature parameters such as angle between y-axis and the vertical direction, sum vector magnitude (SVM), differential SVM (DSVM) based on the following equation [17].

$$SVM_i = \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2} \tag{1}$$

$$\Phi = \arctan\left(\frac{\sqrt{y_i^2 + z_i^2}}{x_i}\right) * \frac{180}{\Pi} \tag{2}$$

$$DSVM_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{3}$$

SVM: Sum vector magnitude
i: sample number
x,y,z : accelerometer value of x, y, z axis
Φ : the angle between y-axis and vertical direction
DSVM: Differential sum vector magnitude

The algorithm first removes noise with a digital second-order Butterworth filter for posture detection and dynamic analysis. The output from the filter is used for calculating fall-feature parameters which are then checked against the first simple threshold. An example of the SVM threshold comparison method is shown in Fig. 2. The threshold is decided when the specificity is the highest with a sensitivity of 100%. When these fall-feature parameters surpasses the threshold, it indicates that a possible fall has occurred. The output of possible falling cases is applied to the second threshold. Finally, based on the output of verifying these parameters with the second threshold, a decision about the fall is drawn. A summary of processing 3D accelerometer data with the fall detection algorithm is shown in Fig. 3. When an actual fall is detected, the gateway immediately sends a push notification to the caregiver. The push-notification is primarily implemented in the gateway with an assistance of Pushbullet [18]. Pushbullet provides full API for creating push notification applications which allows sending text, files, instant messages in real-time between different types of devices (server, gateway, mobile phone, computer).

The back-end of the system includes cloud service and a terminal application. End-users such as family members, doctors or caregivers can use any Internet browser or a push notification enabled smart phone application to subscribe to the push notification server. In addition to receiving push notification messages, end-users can confirm that a fall case is noticed via replying messages by using the "confirmation" button in an application or a browser's interface.

For implementing a sensor node, several hardware components including ADXL345 three-dimensional accelerometer sensor, micro-controller and Bluetooth module are used. ADXL345 is a high resolution and low power consumption digital three-dimensional accelerometer sensor. It provides 16bit two complement output, which incorporates 3 dimensional acceleration along x, y, z axis. It consumes only 90 µA for providing output values at a rate of 400 Hz. It is equipped with SPI and I^2C interfaces for communicating with a micro-controller. In order to achieve a fair and diversified comparison, an ATMega328P and ATMega32U running at 16 MHz are primarily used in our evaluation. Accordingly, Arduino Uno board, Arduino Micro, a customized device are evaluated. The customized device is constructed from ATmega328P 16 Mhz micro-controller, 16kHz crystal oscillator and set of capacitors and resistors. Although the device can operate at both 3.3V and 5V, 3.3V is used for our experiments. The prototype and minimum setup of the customized sensor node are shown in Fig. 4 and Fig. 5, respectively.Detailed specification of these

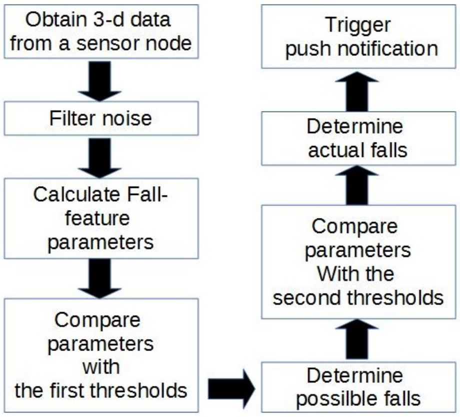

Fig. 3: Fall detection algorithm flow

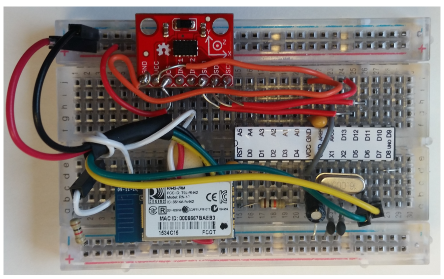

Fig. 4: Prototype of the customized device

boards are shown in table I.

In experiments, both HC-05 Bluetooth classic and Bluetooth Low Energy (BLE) are used for evaluation. HC-05 is a low cost Bluetooth module for data transmission. This module can be configured as a master or a slave. BLE Micro [19] is a ultra-low power Bluetooth module. When operating at 3V, the BLE Micro current consumption is $2\,\mu$A in IDLE mode and about 10.5 mA when transmitting data at a rate of 1 Mbps, while 2 Mbps is the maximum data rate for this module.

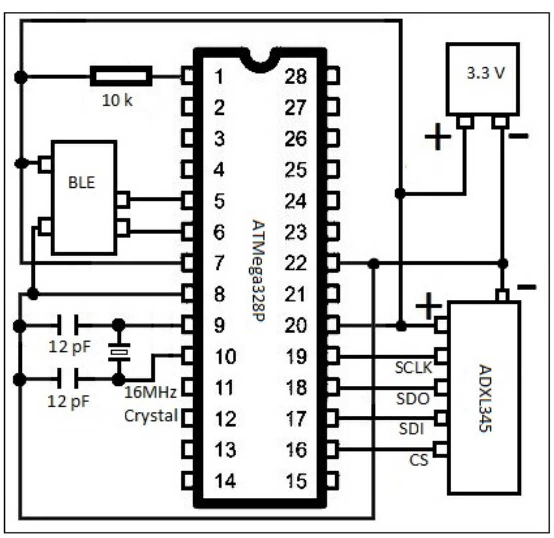

Fig. 5: Minimum setup of the customized device

TABLE I: Devices specifications

Device	Micro-controller	Flash (KB)	SRAM (KB)	Operating Voltage (V)
Arduino Uno	ATmega328P-PU 16MHz	32	2	5
Customized device	ATmega328P-PU 16MHz	32	2	3.3-5
Arduino Micro	ATmega32U4-AU 16MHz	32	2.5	5

TABLE II: Energy consumption when collecting 3D accelerometer data with different sampling rates

Device \ Sampling rate	50 samples/s (mJ)	100 samples/s (mJ)	200 samples/s (mJ)	400 samples/s (mJ)
Arduino Uno	187.6	187.8	188.01	188.05
Customized device	47.817	47.982	48.247	48.345
Arduino Micro	158.55	158.65	158.8	159.1

V. Experimental results

In order to calculate energy consumption of sensor nodes, the following equation [20] is applied: The energy equation (1) is formed based on the fact that total node energy consumption is equal to the sum of energy consumption in both waiting and operating time.

$$E = V \times I(w) \times (t(w) - t(o)) + V \times I(o) \times t(o) \quad (4)$$

E : Total energy consumption (mJ)
V : Voltage supply
$I(w)$: Average current draw during waiting time (mA)
$I(o)$: Average current draw during operating (mA)
$t(w)$: Waiting time (s)
$t(o)$: Operating time (s) In our experiments, current of devices is measured by power monitor produced by MonSoon Solution [21]. The Power Monitor including hardware and advanced software provide a robust power measurement solution for mobile devices such as general purpose development boards, mobile phones or any devices using batteries having capacity less than 3A.

According to datasheet of ADXL345 [22], the sensor is capable of providing a sampling rate of 3200Hz. However, for supporting this data rate, the sensor must operate in a normal energy inefficient mode. To reduce energy consumption, the low power mode has to be used. The maximum data rate of the low power mode is 400Hz. Therefore, 50, 100, 200, and 400 Hz data rates are used in the experiment. The first experiment is to compare energy consumption of the sensor node when collecting three-dimensional accelerometer data during a second with 50, 100, 200, 400 samples/second data rate via SPI configured with 4 Mbps communication speed. Results of this experiment are shown in Table II.

The results of the experiment show that energy consumption of the sensor node based on the customized device is one third of energy consumption of other devices based on Arduino

TABLE III: Energy consumption of the customized device when collecting 100 samples/second 3D accelerometer data with different SPI data rates and I^2C

Board \ Data Rate	125kbps SPI(mJ)	500kbps SPI(mJ)	1Mbps SPI(mJ)	4Mbps SPI(mJ)	400KHz I^2C(mJ)
Customized device	46.992	47.058	47.388	47.982	47.457

TABLE IV: Energy consumption of the sensor node for collecting 100, 200 and 400 samples/s 3D accelerometer data via 1 Mbps SPI and transmitting 100, 200, and 400 samples/s data with 9600 bps and 19200 bps Bluetooth classic, respectively

Board \ Sampling rate	100samples/s (mJ)	200samples/s (mJ)	400samples/s (mJ)
Arduino Uno	308.79	315.6	329.57
Customized device	138.24	140.64	146.18
Arduino Micro	281.15	282.85	288.45

board for all experimental cases even though they are built from the identical micro-controller ATMega328P-PU. As seen in table 1, energy consumption of the sensor node with sampling rate 50, 100, 200 and 400 samples/second is not largely different. Therefore, significant energy savings can be achieved even though trading the energy for a higher sampling rate with the interest of fulfilling application's data rate requirements.

As mentioned, ADXL345 supports SPI and I^2C for communication. Therefore, these interfaces are evaluated by sending 100 samples/s from ADXL345 to the micro-controller in order to choose most appropriate one in terms of energy efficiency. In addition, as SPI data rate impacts energy consumption of the sensor node, effects of different rates are evaluated. Results from these experiments, as shown in Table III, illustrate that applying the SPI interface for obtaining data from 3D accelerometer sensor is more energy efficient than using the I^2C interface when similar data rates are used. Table III also shows that 1 Mbps SPI is still more energy efficient than 400 Khz I^2C. In some cases of time-critical fall detection systems, 4 Mbps SPI can be used without heavy impact on energy consumption because energy consumption of the sensor node increases only slightly when increasing SPI data rate from 1 Mbps to 4 Mbps.

To provide an incisive view of energy consumption of the sensor node, the total energy consumption for obtaining and sending the data to a gateway via Bluetooth is evaluated. Several data rates of Bluetooth classic including 9600 bps, 19200 bps and 230400 bps are applied in the experiment. In order to explore the relationship between these Bluetooth data rates and sampling rate of a 3D accelerometer sensor and Bluetooth sampling, several sampling rates of the sensor such as 100, 200 and 400 samples/second are enumerated. The results are shown in Table IV and Table V. It shows that sending data via Bluetooth with lower data rate consumes less energy for all experimental devices. In addition, the customized one is the most energy efficient device among all experimental devices.

Energy consumption of sensor nodes can be dramatically

TABLE V: Energy consumption of the sensor node when collecting 100, 200, 400 samples/s 3D accelerometer data via 1Mbps SPI and sending the data to a gateway via Bluetooth classic configured with 230400bps

Board \ Sampling rate	100samples/s (mJ)	200samples/s (mJ)	400samples/s (mJ)
Customized device	145.36	147.98	152.39

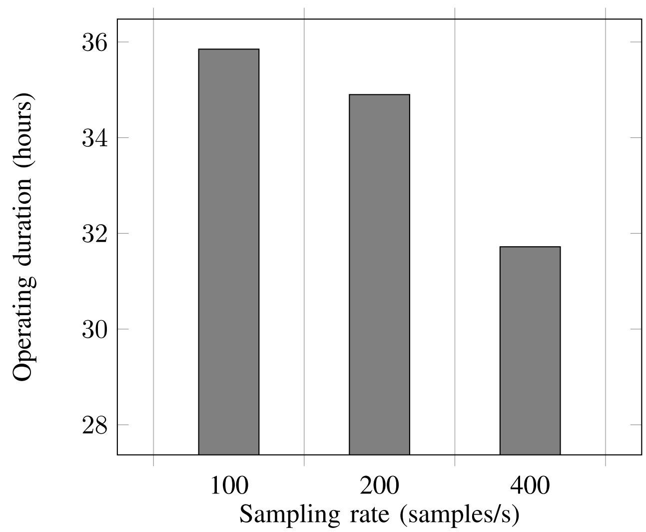

Fig. 6: Operating durations of the customized sensor node supplied with a 1100mAh battery when collecting 100, 200, and 400 samples/s 3D accelerometer data and sending the data via 9600bps BLE

lowered by using Bluetooth Low Energy (BLE) instead of Bluetooth classic. Therefore, in our sensor nodes, BLE at 9600 bps data rate is used for transmitting 3D accelerometer data with sampling rate 100, 200 samples/s while 19200 bps is used when sampling at a rate of 400 samples/s. Results are shown in Table VI. 30% energy consumption of sensor node can be reduced by using BLE. Our sensor nodes are supplied with 3.3 volt 1100 mAh lithium battery which has a small size of 15*37*59 mm and a light weight of 21 grams. With the battery, the sensor node with BLE can operate up to 35 hours when obtaining 3D accelerometer data at the data rate of 100 samples/s and sending it to the gateway with BLE at 9600 bps. The estimated duration of operating time of our sensor node is shown in Fig. 6.

TABLE VI: Energy consumption of the sensor node when collecting 100, 200, 400 samples/s 3D accelerometer data via 1Mbps SPI and sending the data to a gateway via Bluetooth low energy configured with 9600bps for 100, 200 samples/s and 19200bps for 400samples/s

Board \ Sampling rate	100samples/s (mJ)	200samples/s (mJ)	400samples/s (mJ)
Arduino Uno	238.48	247.15	260.77
Customized device	92.04	94.44	104.036
Arduino Micro	213.65	215.85	218.96

VI. Discussions

In our experiments, micro-controller's digital interfaces (i.e. SPI, I^2C), sensor's sampling rate, and wireless communication were investigated. Based on practical results, the developed sensor node is more energy efficient than other nodes based on general purpose development device regarding to the identical micro-controllers and memory capacitor. We suggest that sensor nodes for fall detection and other IoT systems should be as simple as possible in terms of hardware components for energy efficiency.

While experimenting, for sending the same amount of data, we found that SPI is more energy efficient than I^2C for all cases of similar data rates. It is recommended that SPI should be used for communicating between sensors and micro-controllers. A choice of an appropriate SPI data rate depends on sensors' sampling rate and time requirements of particular applications. When fall detection and other IoT-based systems are not extremely time-critical in terms of millisecond, low SPI data rate should be used for saving energy consumption.

Although the low power mode of ADXL345 does not support high sampling rates such as 1600 and 2000 samples/s, it provides 400 samples/s sampling rate which is sufficient for many fall detection algorithms. In the low power consumption mode, energy requirements of sensor nodes when obtaining 3D accelerometer data with 100, 200, and 400 samples/s is slightly different. Therefore, one of these data rates can be chosen for fulfilling the requirements of application while maintaining the desired energy efficiency.

Clock frequency of a micro-controller directly impacts on energy consumption of a sensor node. When a sensor node is not required to perform heavy computation, a low clock frequency should be used. For extensions, a lower clock frequency micro-controller i.e. 1MHz or 8MHz having the same family as ATMega328P could be used for comparing with a customized ATMega328P which is 16MHz but is scaled down to 1MHz and 8MHz by software. Furthermore, when 1MHz clock frequency is applied, a 16k crystal oscillator can be removed from our design because 1MHz internal clock source of ATMega328P can be utilized.

VII. Conclusions

In this paper, we evaluated energy consumption of wireless sensor nodes in a fall detection IoT-based system for extracting an energy efficiency method of designing those sensor nodes. We implemented a simple customized sensor node for achieving a high level of energy efficiency. We compared primary energy consumption sources of several sensor nodes based on our and other designs. Based on experiments, we concluded that our sensor node is energy efficient. When using an 1100 mAh battery, the sensor node can operate up to 35 hours. In addition, our push notification service can improve quality of healthcare services via a mechanism of notification together with acknowledgement.

References

[1] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the internet of things*, pp. 1–24, Springer, 2011.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] D. M. Gaba, "The future vision of simulation in health care," *Quality and safety in Health care*, vol. 13, no. suppl 1, pp. i2–i10, 2004.

[5] W. H. O. Ageing and L. C. Unit, *WHO global report on falls prevention in older age*. World Health Organization, 2008.

[6] CDC-Centers for Disease control and prevention, "Important facts about falls." Updated: Jan. 2016, Accessed Jul 2016, http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html.

[7] D. A. Sterling, J. A. O'Connor, and J. Bonadies, "Geriatric falls: injury severity is high and disproportionate to mechanism," *Journal of Trauma and Acute Care Surgery*, vol. 50, no. 1, pp. 116–119, 2001.

[8] J. A. Stevens, P. S. Corso, E. A. Finkelstein, and T. R. Miller, "The costs of fatal and non-fatal falls among older adults,"

[9] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[10] E. Casilari and M. A. Oviedo-Jiménez, "Automatic fall detection system based on the combined use of a smartphone and a smartwatch," *PloS one*, vol. 10, no. 11, p. e0140929, 2015.

[11] M. Kepski and B. Kwolek, "Embedded system for fall detection using body-worn accelerometer and depth sensor," in *: Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference on*, vol. 2, pp. 755–759, IEEE, (2015).

[12] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PloS one*, vol. 7, no. 5, p. e37062, 2012.

[13] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & posture*, vol. 28, no. 2, pp. 285–291, 2008.

[14] Y. Li, G. Chen, Y. Shen, Y. Zhu, and Z. Cheng, "Accelerometer-based fall detection sensor system for the elderly," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, vol. 3, pp. 1216–1220, IEEE, 2012.

[15] C. Shuo, "Fall detection system using arduino fio," in *Proceedings of the IRC Conference on Science, Engineering and Technology, Singapore*, vol. 13, 2015.

[16] RaspberryOrg, "Raspberry-pi-3-model-b." Updated: Jan. 2016, Accessed Jul 2016, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[17] D. Lim, C. Park, N. H. Kim, S.-H. Kim, and Y. S. Yu, "Fall-detection algorithm using 3-axis acceleration: combination with simple threshold and hidden markov model," *Journal of Applied Mathematics*, vol. 2014, 2014.

[18] Pushbullet, "Pushbullet." Updated: Jan. 2016, Accessed Jul 2016, https://www.pushbullet.com/.

[19] Seeed Studio, "Ble micro." Updated: Jan. 2016, Accessed Jul 2016, http://www.seeedstudio.com/wiki/BLE$_Micro$.

[20] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, K. Mankodiya, P. Liljeberg, and H. Tenhunen, "Fog computing in body sensor networks: An energy efficient approach," in *IEEE International Body Sensor Networks Conference (BSN15)*, 2015.

[21] M. Solutions, "Power monitor." Updated: Jan. 2016, Accessed Jul 2016, https://www.msoon.com/LabEquipment/PowerMonitor/.

[22] Analog Devices, "Adxl345-digital accelerometer." Updated: Jan. 2016, Accessed Jul 2016, http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf.