Microprocessors and Microsystems 56 (2018) 34–46

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Energy efficient wearable sensor node for IoT-based fall detection systems

Tuan Nguyen Gia*[a], Victor Kathan Sarker[a], Igor Tcarenko[a], Amir M. Rahmani[b,c], Tomi Westerlund[a], Pasi Liljeberg[a], Hannu Tenhunen[a]

[a] Department of Future Technologies, University of Turku, Turku, Finland
[b] Department of Computer Science, University of California, Irvine, USA
[c] Institute of Computer Technology, TU Wien, Vienna, Austria

ARTICLE INFO

Keywords:
Internet-of-Things
IoT
Fall detection
Energy efficiency
Wearable devices
Accelerometer
Gyroscope
Magnetometer
nRF

ABSTRACT

Falls can cause serious traumas such as brain injuries and bone fractures, especially among elderly people. Fear of falling might reduce physical activities resulting in declining social interactions and eventually causing depression. To lessen the effects of a fall, timely delivery of medical treatment can play a vital role. In a similar scenario, an IoT-based wearable system can pave the most promising way to mitigate serious consequences of a fall while providing the convenience of usage. However, to deliver sufficient degree of monitoring and reliability, wearable devices working at the core of fall detection systems are required to work for a prolonged period of time. In this work, we focus on energy efficiency of a wearable sensor node in an Internet-of-Things (IoT) based fall detection system. We propose the design of a tiny, lightweight, flexible and energy efficient wearable device. We investigate different parameters (e.g. sampling rate, communication bus interface, transmission protocol, and transmission rate) impacting on energy consumption of the wearable device. In addition, we provide a comprehensive analysis of energy consumption of the wearable in different configurations and operating conditions. Furthermore, we provide hints (hardware and software) for system designers implementing the optimal wearable device for IoT-based fall detection systems in terms of energy efficiency and high quality of service. The results clearly indicate that the proposed sensor node is novel and energy efficient. In a critical condition, the wearable device can be used continuously for 76 h with a 1000 mAh li-ion battery.

1. Introduction

Fall is one of the most trivial reasons causing traumas and serious injuries (e.g. bone fractures or traumatic brain damages caused by head traumas) [1,2]. Elderly people are likely to fall and they often have more serious consequences after falling than people of other ages. According to statistics, 30% of those over 65 and 50% of those over 80 years old fall every year with hazardous results [1]. Because of high morbidity (almost 20% of fall lead to serious traumas), about 40% of all nursing home admissions are related to fall [3].

Treatment of injuries from a fall often lasts over a long period of time and is very costly (e.g. 30,000 US dollars for a serious case in hospital) [4,5]. The proportion is as follows: 63% of fall-related costs accounts for hospitalizations, 21% is for emergency department visits and 16% is for outpatient visits. However, despite the high significance of the problem, timely aid is only delivered in half of the cases. Unreported cases lead to the deterioration of injury which might complicate treatments later.

Fear of falling amplifies the negative post-fall consequences and might decrease patient's confidence [6]. As a result, it limits the patient's activities, reduces social interactions and eventually causes depression [7,8]. Thus, there is an urgent need of fall detection systems. A quick response to the incident might decrease the risk of serious consequences after a fall. Correspondingly, it helps to reduce treatment costs and to increase chance of recovery. In [9], authors have separated fall detection systems into three groups based on wearable devices, ambiance sensors, and cameras. Systems based on wearable devices seem to be more popular because they can detect a fall more accurately regardless of the patient's location (i.e. indoor and outdoor) and do not interfere the patient's privacy and daily activities. Wearable devices often acquire parameters related to motion such as acceleration, rotation and the direction of motion [10].

It is a challenge for wearable sensor nodes to differentiate between fall events and casual daily activities, or to notify doctors in real-time. Due to their resource constraints (e.g. limited power and storage capacity), it is required to have an advanced system which helps to reduce

* Corresponding author.
E-mail addresses: tunggi@utu.fi (T. Nguyen Gia), vikasar@utu.fi (V.K. Sarker), igotsa@utu.fi (I. Tcarenko), tovewe@utu.fi (A.M. Rahmani), pakrli@utu.fi (T. Westerlund), amirr1@uci.edu (P. Liljeberg), hannu@kth.se (H. Tenhunen).

http://dx.doi.org/10.1016/j.micpro.2017.10.014
Received 5 March 2017; Received in revised form 9 October 2017; Accepted 30 October 2017
Available online 02 November 2017

computationally heavy loads on wearable sensor nodes, while maintaining or improving quality of service. Internet-of-Things (IoT) is one of the most suitable candidates for such systems as it consists of a wide range of advanced technologies such as sensing, wireless sensor network and cloud computing for interconnecting virtual objects with physical objects. IoT-based systems can help to reduce wearable devices' burdens by shifting high-computational tasks from wearable devices to their smart gateways. For example, the gateways can perform complex fall detection algorithms (i.e. algorithms based on discrete wavelet transform or data mining). In addition, smart gateways help to improve quality of service by providing advanced services i.e. local storage for storing temporary data or push notification for informing abnormality in real-time.

It is inevitable that IoT can comprehensively help to reduce power consumption of wearable devices by sharing the work load. However, IoT cannot always guarantee a high level of energy efficiency in wearable devices. Other primary issues (i.e. data acquisition and data transmission) causing high energy consumption in wearable sensor nodes must be attentively considered. When a wearable sensor node is energy inefficient, it possibly causes unreliability and reduces quality of service.

In the previous work [11], we have proposed an IoT-based fall detection system. The system comprises of energy efficient sensor nodes, a smart gateway, and a back-end system. The gateway with a Fog layer [12,13] helps to achieve energy efficiency at sensor nodes. In that paper, a sensor node attached to human chest acquires data from a three-dimensional (3-d) accelerometer and transmits the data to the smart gateway via BLE (Bluetooth Low Energy). The main computation (i.e. a customized fall detection algorithm) is performed at the smart gateway since the gateways are powerful in terms of hardware specification and it is supplied by a wall power outlet. The work shows several analysis of primary communication interface buses' power consumption. The results show that SPI (Serial Peripheral interface) consumes less power than I^2C (Inter-Integrated Circuit) and UART (universal asynchronous receiver/transmitter) while SPI's data rates are higher than others (e.g., SPI can support a high data rate of 4 Mbps and more).

The work presented in this paper is a major extension of our recent work published in [11]. In the paper, we aim to study and minimize energy consumption of the wearable sensor node in an IoT-based fall detection system. Furthermore, we analyze undisclosed issues in the previous work. For example, the analysis of advantages and disadvantages of software-based SPI and its impact on energy consumption of a sensor node are presented. Moreover, we analyze energy consumption of the sensor node in various transmission distances and different transmission conditions (e.g. line-of-sight transmission, and transmission via objects). We also investigate and discuss impacts of different sensors (e.g. accelerometer, gyroscope and magnetometer) on both total energy consumption of the sensor node and an accuracy of the fall detection mechanism. We analyze the accuracy of the fall detection system in exceptional cases such as users having abnormal postures. In addition, we discuss and provide comprehensive methods for overcoming limitations (e.g. P2P communication) in the previous work. In this paper, we present the design and implementation of an energy efficient wearable sensor node based on a customized nRF module. The design helps to solve the limitation of P2P communication by offering many-to-many communication between sensor nodes and gateways. Unlike BLE used in the previous work [11] which is connected to the micro-controller via UART, the nRF module in the proposed design uses SPI as its communication bus. Therefore, it incurs a new issue of using several SPI buses simultaneously by a single microcontroller (i.e., SPI communication buses for collecting data from sensors and for transmitting the data via nRF). Therefore, these issues are discussed to find out the most appropriate solution in terms of energy efficiency, feasibility, and complexity. The proposed wearable sensor node is low-cost, lightweight, tiny, energy efficient and flexible. It can be configured to suit to different fall detection algorithms based on motion (e.g. acceleration or angle). The wearable sensor node can provide a viable solution for everyday use without interfering user's daily life. Furthermore, we customize the fall detection algorithm presented in our previous paper for suiting to the proposed sensor node and improving QoS (e.g. the accuracy of the fall detection system).

The rest of the paper is organized as follows: Section 2 includes related work and motivation for this work. Section 3 provides an overview of the IoT-based fall detection system's architecture. Section 4 emphasizes on design principles and reasons behind component and technology selection. Section 5 illustrates the implementation details of the proposed sensor node. Section 6 provides insights about experimental setup and results. Section 7 discusses various issues and findings, and proposes possible solutions. Finally, Section 8 concludes the work.

2. Related work and motivation

Several efforts have been devoted in proposing wearable sensor nodes for fall detection systems. For instance, Casilari et al. use an accelerometer in a smart watch to detect a fall. Accelerometer data is transmitted via BLE from the smart watch to a smart phone which processes data and detects a fall. Then, the smart phone, which acts as a gateway, sends a notification to Cloud via 3G/4G [14]. In another work [15], authors use a depth camera (Kinect) with an accelerometer-based wearable to improve the accuracy of fall detection. Collected data is processed at PandaBoard for detecting a fall in real-time.

Pivato et al. [16] present a wearable wireless sensor node for fall detection. The wearable node whose size is about three times larger than a 2 Euro coin, requires low average current about 15 mA and 25 mA at 50% and 100% duty cycle, respectively. The node is equipped with a 3-d accelerometer ADXL345 and a wireless chip (i.e. CC2420) for gathering and sending acceleration data to a gateway, respectively.

Chen et al. [17] present wearable sensors for a reliable fall detection system. The sensors collect data from low-cost and low-power MEMS accelerometers and send the data via RF. By deploying the sensors at home, the position of the fallen person can be detected.

Biros et al. [18] propose a wearable sensor for a smart household environment. The wearable sensor collects 3-d acceleration and angles from an accelerometer and a gyroscope, respectively. The sensor sends the collected data via ZigBee to Arduino Uno connected to a computer for further processing and detecting a fall.

Erdogan et al. [19] discuss a data mining approach by using k-nearest neighbors for a fall detection system. A wearable device in the system is based on a general purpose board equipped with motion sensors.

In another work [20], the authors present a sensor node based on GSM communication and 3-d accelerometer for a fall detection system. A fall location can be easily detected by the system.

In other works [21,22], authors utilize general purpose boards (e.g. Arduino Uno, Arduino Fio) as the core of fall detection sensor nodes. Although the sensor nodes are low-cost and provide some useful services, they still have several drawbacks such as high power consumption and large physical size. It is known that general purpose boards are often equipped with extra components such as a voltage regulator and a FTDI USB to UART chip ultimately resulting in energy inefficiency.

In several works [14,16–20], fall detection sensor nodes based on motion data often utilizes one or several types of sensors such as accelerometer, gyroscope or magnetometer. The selection of a sensor type or a combination of several sensor types in a single sensor node is mainly focused on functions and features of the sensor(s) while energy consumption of the sensor(s) is not attentively considered. For example, the accelerometer and the gyroscope are often used together in the fall detection applications so as to improve the accuracy of fall detection.

It is known that energy consumption of a sensor node dramatically impacts quality of service. When energy consumption is high, it may

cause or lead to negative consequences such as a short operating duration, discontinuation of services or unreliability. However, to the best of our knowledge, the actual issues limiting energy efficiency of a sensor node in an IoT-based fall detection system have not been elaborately investigated. For example, energy consumption of communication buses between a micro-controller and its slave devices (i.e. sensors or a wireless communication module) is not considered in many sensor node designs. Therefore, we investigate energy consumption of communication bus interfaces such as SPI, I2C, and UART. The results showing the impact of primary communication buses on energy consumption of a sensor node can be used as a premise to design the high energy efficient sensor node suiting for different fall detection applications.

The relationship between an IoT-based fall detection sensor node's sampling rate and energy consumption has not been examined in other works. Therefore, in this paper, we analyze the relationship with different configurations and discuss optimal solutions for achieving both high levels of energy efficiency and fall detection accuracy.

A low-power sensor node in fall detection applications often uses BLE as a primary wireless communication protocol [11,23–25]. Although BLE provides many advantages (e.g. low power, fast cyclic redundancy check, and connection improvements), it still has several limitations (e.g. p2p communication, a complex stack with several profiles) which may increase service costs and may not guarantee the highest level of energy efficiency. Therefore, we analyze another low power wireless communication protocol which helps to avoid the limitations of BLE while maintaining high quality of service.

In the paper, we also investigate factors impacting on energy consumption of a wearable sensor device. These factors are such as a micro-controller, motion sensors (accelerometer, gyroscope, and magnetometer), sampling rate, wireless transmission data rate, transmission distance, and software. By applying an optimal combination of hardware design and software techniques, it is possible to provide a novel tiny, and light-weight wearable sensor node with a high level of energy efficiency.

3. Overview of an IoT-based fall detection system's architecture

An overview of an IoT-based fall detection system is presented with the purpose of showing the role and the hierarchical position of wearable devices in the system. The system architecture shown in Fig. 1 consists of three main parts including wearable sensor nodes, a gateway and a back-end system.

A sensor node of an IoT-based fall detection system is responsible for acquiring motion data (i.e. acceleration or rotation angle) and transmitting the data via a wireless communication protocol to a smart gateway. Depending on particular fall detection systems, the collected data can be pre-processed or kept intact before being transmitted. In most of the cases, collected data (raw data) is transmitted without pre-processing by complex algorithms or methods (i.e. wavelet transformation or neural filtering) [26] because pre-processing with complex mechanisms like fall detection based on k-nearest neighbor algorithm requires significant computational power. Correspondingly, energy efficiency must be sacrificed and latency dramatically increases for running such complex algorithms at a sensor node. In order to avoid these issues, complex algorithms are implemented and run at smart gateways [27,28].

In addition to primary tasks of receiving data from sensors and transmitting the data to Cloud servers, a smart gateway with a fog layer provides advanced services such as push notifications, local storage, web-host, and fall detection. Complex algorithms can be run effectively with low latency at the fog layer of a smart gateway due to their advantages of the constant power supply, embedded operating system, and powerful hardware (e.g. a gateway is often about 50–100 times more powerful than a sensor node). Correspondingly, fall can be detected and notified to doctors or caregivers in real-time.

A back-end system consists of Cloud servers and terminals (i.e. end-user's Internet browsers or mobile phone applications). Via the back-end system, doctors, and caregivers can monitor a patient in real-time or history of patient records remotely. In addition, the back-end system may help doctors in disease treatment by providing analyzed data and history of records.

4. Sensor node design

A sensor node for an IoT-based fall detection system primarily comprises of a micro-controller, a motion sensor or sensors, and an nRF block whose connections are shown in Fig. 2. The micro-controller performs main tasks of gathering data from sensors, formatting and transmitting the collected data to the nRF block, and controlling sensors and I/O interfaces (i.e. SPI, I^2C or UART). It consumes a large portion of total power consumption of the sensor node. Therefore, it is important to apply an optimal micro-controller for performing mentioned tasks efficiently in terms of latency and energy consumption.

In our application, a 8-bit micro-controller is more suitable than a 32-bit micro-controller. Based on experiments run by Atmel [29], an Atmel 8-bit AVR device is more efficient than an Atmel ARM Cortex®M0 + based 32-bit MCU in terms of hardware near-functions. For example, an Atmel 8-bit AVR device requires 12 cycles to receive one byte from SPI using interrupt while an Atmel ARM Cortex®M0 + based 32-bit MCU requires 33 cycles for performing the same task. When running a recursive 15-stage Fibonacci algorithm, a 8-bit AVR micro-controller needs 70 bytes of stack while the 32-bit ARM-based device needs 192 bytes [29]. In simple applications such as receiving data from SPI using interrupt, assuming a SPI data bandwidth of 80 kbps, the 8-bit AVR micro-controller consumes 36.1 uA while the 32-bit ARM-based micro-controller consumes 48.1 uA. During sleep mode, a 8-bit AVR micro-controller consumes 100 nA while a 32-bit ARM-based micro-controller consumes 200 nA [29].

In [11], we have shown that a 8-bit AVR ATMega micro-controller is capable of successfully performing several tasks (e.g. data gathering and data transceiving) without infringing latency requirements of real-time monitoring systems. The 8-bit AVR micro-controller supports several clock frequencies such as 4, 8, 16 and 20 MHz which completely

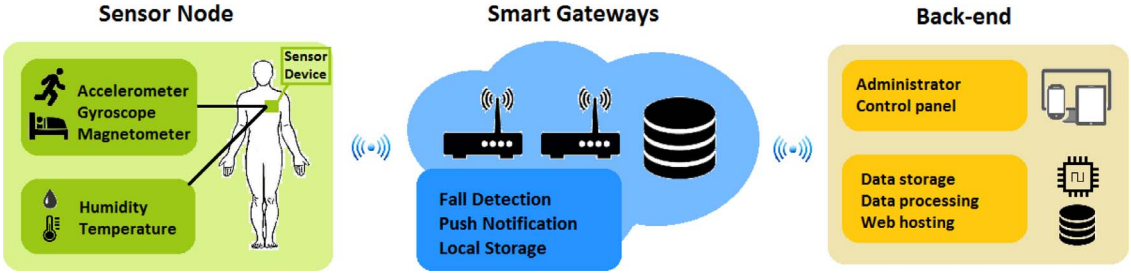

Fig. 1. The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers.

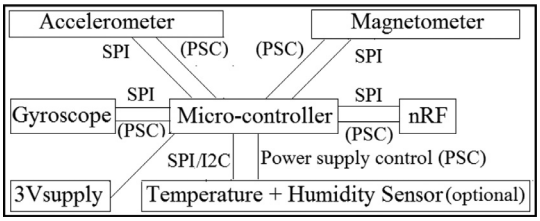

Fig. 2. Connection of sensor node's primary components.

fits to our fall detection application using a few hundred samples per second. In addition, it consumes low power in an active mode, and supports several sleep modes and advanced features for saving power consumption. The micro-controller supports all popular communication bus interfaces (e.g. UART, SPI, and I^2C). Furthermore, it is small and low-cost (around 1–2 dollars). Therefore, it is completely suitable for our fall detection application.

Depending on particular fall detection algorithms running on smart gateways, one or several types of motion sensors (such as accelerometer and gyroscope) can be integrated in a sensor node [30]. A combination of several types of motion sensors (e.g. accelerometer, gyroscope, and magnetometer) altogether may help to improve the accuracy of a fall detection system but it causes higher energy consumption. Fortunately, these sensors can be controlled by software (e.g. entering sleep mode) for saving energy. For example, replacing a 3-d accelerometer (e.g. ADXL345 accelerometer) in a sensor node by a combination of a 3-d accelerometer, a 3-d gyroscope and a temperature sensor (e.g. ADXL345, Kionix KXG07, and STML20), energy consumption during the idle mode in a second only increases about a few μW (e.g. less than 10 μW). In this paper, in order to provide a flexible and low-energy wearable sensor node suiting to different IoT-based fall detection systems, three motion sensors including 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer are integrated in our sensor node. When sensors are not in use, they are forced to sleep. In addition, in order to provide the optimal sensor node in terms of both energy efficiency and the fall detection accuracy, a comprehensive analysis and a discussion of sensor node in different configurations are presented in Section 6 and Section 7. It is known that sampling rates and communication protocols (UART, I^2C and SPI) dramatically impact on energy consumption of sensors [11]. Often, these sensors support several sampling rates of which low sampling rates (50–100 Hz) can be run in a low-power mode and high sampling rates are run in a normal mode. However, when the sampling is too low, it negatively impacts on the accuracy of fall detection. A relationship between sensors sampling rate and energy consumption is investigated in Section 6 for finding an appropriate sampling rate which provides a high level of fall detection accuracy while consuming low energy.

An nRF module consisting of an nRF integrated circuit (IC) and an on-PCB printed antenna is chosen for the design because it consumes less energy while supporting high data rates. Also, comparing to Wi-Fi, XBee and Bluetooth, nRF is more suitable for the sensor node because it consumes the least power (i.e. about 5–10%, 5–10%, and 80% less power than BLE, XBee, and Wi-Fi [31]) and it supports software customization enhancing a sensor node's flexibility. Depending on particular application requirements, a transmission data rate can be customized. In some cases, it can transmit data with a data rate of 2 Mbps.

According to our previous work [11], SPI consumes less power than I^2C and UART communication interfaces at the same data rate. Therefore, SPI is utilized for connecting the micro-controller with motion sensors as well as the nRF module. However, applying multiple SPI communication bus causes some difficulties in data management and data verification.

5. System implementation

5.1. Sensor node implementation

A sensor node must be able to operate reliably for a long period of time. To achieve this, each component of the sensor node must be energy-efficient in both hardware and software. Based on our previous work's investigation [11] and the specification of an AVR ATMega328P micro-controller, a 8-bit AVR ATMega328P micro-controller is suitable for the sensor node. In the implementation, all unneeded interfaces (e.g. UART, I^2C) and internal modules (blocks) of the micro-controller are intentionally disabled for reducing energy. For example, unneeded internal modules (i.e. Serial, ADC, or brownout detection) are turned off. Similarly, necessary interfaces and modules are forced to be disabled in most of the time. They are merely enabled or waken up only for performing their tasks. Energy consumption of the sensor node with and without disabling unneeded modules is shown in Table 3.

The micro-controller supports up to 20 MHz. However, the higher clock frequency is applied, the higher power the sensor node has to be provided because the micro-controller requires higher voltage supply and draws more current when running at high frequencies. For instance, at 16 MHz an ATMega328P micro-controller needs 5 V and consumes about 57.6 mJ for running a test function while at 8 MHz an ATMega328P micro-controller only needs 3 V and consumes approximately 46.8 mJ for running the same test function. In both cases, an array of 100 values is retrieved and a sum of two adjacent values is written back to the array. When supplying the micro-controller 2.2–2.5 V for running at 4 MHz, the micro-controller consumes less power than at 8 MHz. However, if applying 2.2 V power supply, it would be incompatible for other primary components of the sensor node. For example, sensors (e.g. MPU-9250) and nRF require 3 V power supply for a stable operation. In order to solve the incompatibility issue, the micro-controller must be supplied with 3 V or the sensor node must be equipped with a voltage regulator converting a higher voltage down to 3 V. Correspondingly, in both cases, it may waste 15–30% of total power consumption while it may not operate stably. Therefore, running the micro-controller at 8 MHz is suitable for our sensor node because extra components like voltage regulator(s) can be removed while a high clock frequency can be utilized. Another reason of choosing 8 MHz and 3 V power supply is that when a 3 V battery drains, the voltage supply from the battery may drop until around 2.7 V which is still suitable for the sensor node. In addition, the choice of 8 MHz and 3 V is suitable for extending the sensor node for the future use such as collecting e-health data (e.g. ECG, EMG and EEG). Analog front-end ICs for these signals often require 3 or 3.3 V power supply.

MPU-9250, which is 9-axis MotionTracking sensor combining a 3-d accelerometer, a 3-d MEMS gyroscope, a 3-d MEMS magnetometer and a Digital Motion Processor hardware accelerator engine, is used in the implementation for sensing motion data. The MPU-9250 sensor is fully programmable and able to support low-power and sleep modes. For example, the gyroscope sensor consumes 8 μA in sleep mode. One of advantages is that each internal module such as accelerometer, gyroscope, and magnetometer can be controlled separately. Correspondingly, the sensor node can be customized for particular fall detection applications without sacrificing energy efficiency of the sensor node intensively. Depending on the applications or scenarios, some internal modules can be activated while others can be in sleep modes. In the implementation, several scenarios described in Section 6 are applied for investigating energy consumption of our sensor nodes and the accuracy of fall detection. The sensor requires a supply voltage from 2.4 V to 3.6 V. The MPU-9250 sensor supports both SPI and I^2C.

An nRFL2401 module, which is a low-power transceiver operating in ISM frequency band from 2.4 GHz to 2.4835 GHz, is used. The module integrated with an embedded base-band protocol engine supports several operating modes. For example, the module can operate at 250 kbps, 1 Mbps, and 2 Mbps. In the implementation, 250 kbps is

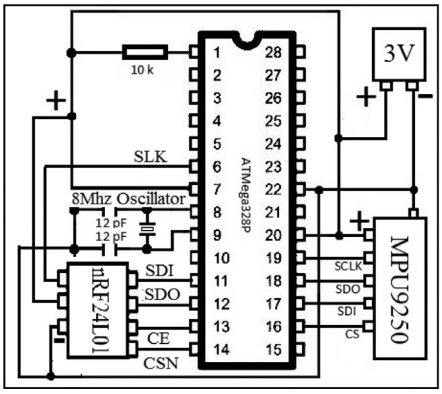

Fig. 3. Minimal setup of a sensor node using both hardware and software based SPI.

preferred because it fulfills the data rate requirement of the system and consumes the lowest energy than other data rates. The module is connected to the micro-controller via SPI.

In addition, for evaluating energy consumption of the sensor node when using SPI and software SPI, two different sensor nodes based on SPI and software SPI are implemented. The first node uses a combination of SPI and software SPI while the second node merely uses SPI. Minimal setup of these nodes is shown in Figs. 3 and 4.

Finally, the proposed wearable sensor node is built, as shown in Fig. 5. The wearable sensor is tiny, light-weight and low-cost. The total cost of the wearable sensor node is less than 11 Euros in which a motion sensor MPU9250 and an nRF24L01 module cost about 5 Euros and 2 Euros, respectively.

5.2. Gateway and back end implementation

A gateway is implemented by a combination of an nRF transceiver and Raspberry Pi [32]. An nRFL2401 module described above is used as an nRF transceiver of the gateway. The module is connected to the Raspberry Pi via SPI.

Several algorithms, presented in [11,33], are applied in a smart gateway for testing functionality of sensor nodes. These algorithms are chosen because they can be replicated easily in the gateway for the verification purposes and they provide a high level of accuracy in detecting fall. These algorithms operate based on acceleration and angular motion which vary in time during a fall, as shown in Fig. 7. In these algorithms, different types of filters are used for removing noise from the collected data. Then, the fall-related parameters such as Sum Vector Magnitude (SVM) and differential SVM (DSVM) are calculated by the formula shown in Equation (1,2 and 3). It is noted that the Eq. (2) is not applied for gyroscope.

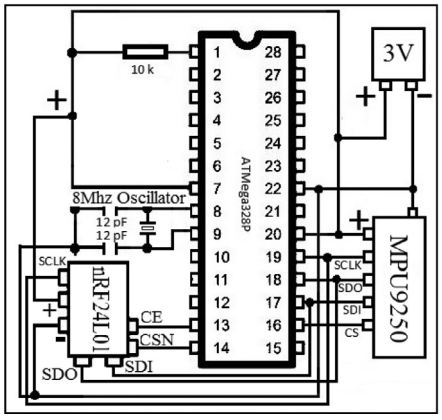

Fig. 4. Minimal setup of a sensor node using hardware SPI only.

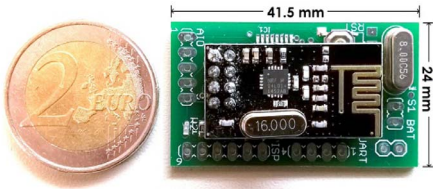

Fig. 5. Prototype of proposed sensor node beside a 2 Euro coin for size comparison.

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

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

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

SVM: Sum vector magnitude

i: sample number

x,y,z : accelerometer value or gyroscope value of x, y, z axis

Φ : the angle between y-axis and vertical direction

DSVM: Differential sum vector magnitude

These fall-related parameters will be further processed or compared with several pre-defined thresholds. If the processed data or fall-related parameters are larger than the predefined thresholds, a fall is detected. In details, each sensor has a specific threshold. For example, SVM of 3-d acceleration is around 1 g in most of the cases (e.g. standing, sitting or walking). When a patient falls, the SVM value increases instantly more than 1.9 g at the fall moment. Therefore, the threshold value can be defined as 1.6 or 1.7 g. Similarly, the threshold values of 3-d gyroscope and 3-d magnetometer can be defined. In this paper, we do not focus on fall detection algorithms in a smart gateway. Therefore, we customize the threshold-based fall detection algorithm presented in our previous paper [11]. The customized algorithm includes several stages such as filtering, calculating fall feature parameters, combining fall feature parameters from several sensors, and comparing with two-level thresholds. The detailed flow of the customized fall detection algorithm is shown in Fig. 6. Based on our experiments and results shown in Section 6, relying on data collected from a single sensor type does not provide a high level of accuracy in some cases. Therefore, in the paper, we add two extra stages to the fall detection algorithm. The first stage combines and analyzes several fall feature parameters from several sensors such as 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer. In case that only two sensor types are used (e.g. accelerometer and gyroscope), the parameters from the absent sensor will be ignored. After the first stage, there will be two cases: (i) if all fall feature parameter values from collected sensors are larger than their own

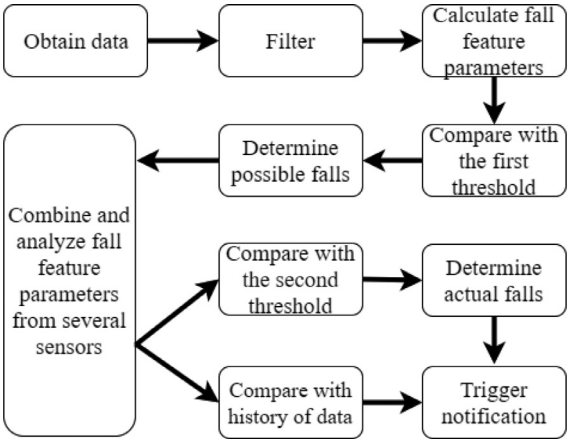

Fig. 6. Fall detection algorithm flow.

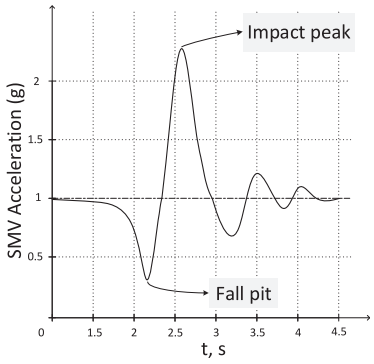

Fig. 7. Acceleration changes in time during a fall.

thresholds, they are compared with their second thresholds. If one of the comparison results shows that the fall feature parameter value is larger than the second threshold, it triggers notification for informing a fall; (ii) if one of the fall feature parameter values is less than the first threshold while other parameters are larger, they are compared with their own values in the past 1 and 2 s for finding dysfunctional or unstable sensor(s).

In our implementation, 1.6 and 1.9 g are used as the first and the second threshold for SVM of 3-d acceleration while 130 and 160 deg/s are used as the first and the second threshold for SVM of 3-d gyroscope. Depending on particular requirements of a fall detection system, it is possible to run one of complex threshold-based algorithms presented in [34–37] or complex machine-learning based algorithms presented in [38–40] at our smart gateway. In such cases, our sensor nodes are still compatible and able to operate efficiently.

The push notification is implemented at Cloud and an Android application via Google's Push API. When the gateway detects a fall, it sends a message (a patient id and time when the patient falls) to the Push service at Cloud servers which then remotely notifies responsible doctors and caregivers in real-time. In addition to mentioned services, smart gateways are implemented with a Fog layer for providing advanced services such as local storage, local host with user interface, data processing, data compression, security, channel managing, categorization. However, in this paper, these Fog services provided at smart gateways are not our main focuses. Therefore, only an overview of Fog and Fog services are presented in the paper while details of the Fog layer and services including description, structure, design and implementation are presented in our other papers and book [26–28,41–44].

In our system, all collected data from sensor nodes are temporarily stored in local database of smart gateways. The local database helps to avoid losing data when the connection between smart gateways and Cloud servers is interrupted. When the connection is re-established, gateways send all recorded data in the database to Cloud. The database is implemented with MongoDB. Categorization service is used to distinguish Intranet users and Internet users with the purpose of reducing latency of services. For example, when the system detects a fall, it checks the status of a doctor or a caregiver responsible for the person falling. If he or she is currently connected to the local network, the system sends the push notification message directly from smart gateways to him or her. This helps to avoid a long latency of transmission via Cloud. The service is implemented by a combination of scanning service and database. The "iw" package helps to check the status of connected users in the local network. Then, the results are stored in the database. The categorization service and local host with user interface allow doctors or caregivers access real-time data directly at the gateways. In order to implement the local host, HTML5, CSS, JavaScript, JSON, Python, and XML are used. Channel management helps to avoid channel conflict by assigning free channels for newly connected sensor nodes. The channel management service triggers the push notification service for informing to system administrators in case of channel conflict.

6. Experimental setup and results

Energy consumption of a sensor node for a fall detection IoT system is calculated with Eq. (4) [45]. Total energy consumption of the node is equal to a sum of energy consumed during operating and waiting.

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

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 order to provide an overview of sensor nodes used in measurements and comparisons, their hardware specifications are shown in Table 2. In the experiments, each measurement is carried out during 5–10 min and a professional power monitoring tool from Monsoon Solution is used [46]. This tool is able to accurately monitor minimum, maximum and average voltage, current draw, power consumption of a sensor node. In addition, Monsoon provides an advanced utility for plotting monitored values in time series, which helps us to detect abnormality during measurements. Although average power consumption in one second and energy consumption per second is identical, to maintain consistency throughout the paper, energy consumption per second is reported instead of power consumption retrieved from the monitor.

In order to determine the suitable method for waking up the micro-controller from deep sleep or normal sleep modes, several general-purpose timers and a watchdog timer are used. In the experiment, the nRF module is not active and the sensor node acquires data from different sensors (i.e. accelerometer, gyroscope, and magnetometer) via 1 Mbps SPI with a data rate of 50 samples/s. Energy consumption of the sensor node in one second is captured and shown in Fig. 8. Results from the Fig. 8 show that energy consumption of the sensor node significantly decreases when using a watchdog timer instead of general-purpose timers. The main reason is that a watchdog timer can wake up the micro-controller from the deepest sleep mode(s) whilst other timers cannot. Results show that an 8-bit timer is more energy efficient than a 16-bit timer. Since the maximum data rate supported by the watchdog timer of ATMega328P is 62 samples/s, a data rate of 50 samples/s is most suitable for the wearable sensor node.

For evaluating energy consumption of the wearable sensor node when using different protocols, 50 samples/s data is acquired from several sensors such as accelerometer, gyroscope and magnetometer via SPI and I^2C. In the experiment, the nRF module is not active and energy

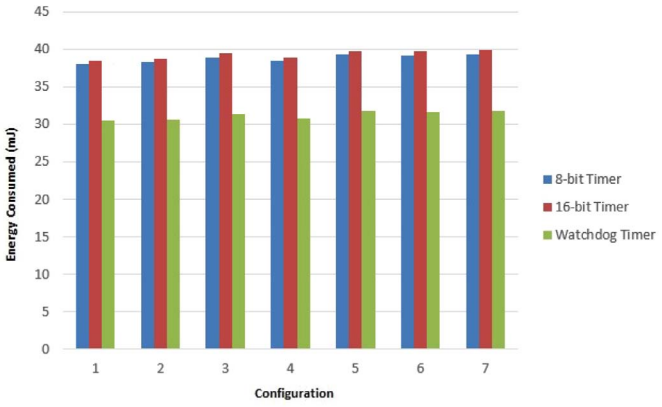

Fig. 8. Energy consumed per second when collecting data from several sensors at 50 samples/s via different techniques.

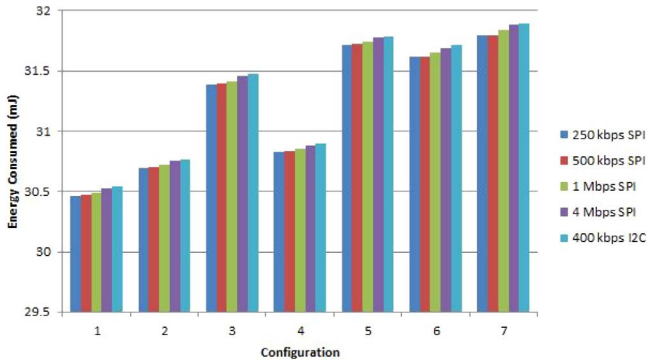

Fig. 9. Energy consumed per second of the sensor node when collecting data from multiple sensors at 50 samples/s using SPI and I^2C.

consumption of the sensor node shown in Fig. 9 is measured for one second. It can be seen that SPI consumes less energy than I^2C in most of the cases.

We compare energy consumption of several sensor nodes based on general purpose platforms and our sensor node. Energy consumption is measured when collecting data from a 3-d accelerometer for one second with a data rate of 50, 100, 200 and 500 samples/s via SPI. In the experiment, software-based techniques for energy efficiency are not applied and all modules for wireless communication (e.g. nRF and BLE) are neither active nor used. Results shown in Fig. 10 indicate that the proposed sensor node consumes the least energy for collecting 3-d acceleration via SPI in all applied data rates. One of the reasons for a high level of energy efficiency in the proposed sensor node is that the sensor node is designed with a minimum number of required components (unnecessary components e.g. FTDI or voltage regulators are removed from the design).

In order to analyze energy consumption of the sensor node, several configurations shown in Table 1 are used. An accelerometer module MPU9250 [47] can support a data rate up to 4000 Hz. However, the low-power mode of the accelerometer cannot be applied at this high data rate. Obviously, the normal operating mode requires more energy than the low-power mode. Therefore, the low-power mode of the accelerometer, which supports a maximum data rate of 500 Hz is applied. For investigating energy consumption of the accelerometer module, several data rates lower than 500 Hz are used in the experiments. Similarly, the low data rate and the low-power mode are applied for gyroscope and magnetometer. In the experiment, several data rates (i.e. 50, 100, 200, and 500 samples/s) are applied to collect data via 1 Mbps SPI in different configurations and the nRF module is not active. Results shown in Fig. 11 indicate that for data rates in a low-power mode, a 3-d accelerometer consumes the least amount of energy among three sensors while a 3-d magnetometer consumes the most energy. In addition, the results reveal that utilizing both accelerometer and gyroscope

Fig. 10. Energy consumption per second of different devices when collecting 3-d accelerometer data at different sampling rates via SPI.

Table 1
Scenarios setup.

	Accelerometer	Gyroscope	Magnetometer
Configuration 1 (Conf 1)	X		
Configuration 2 (Conf 2)		X	
Configuration 3 (Conf 3)			X
Configuration 4 (Conf 4)	X	X	
Configuration 5 (Conf 5)		X	X
Configuration 6 (Conf 6)	X		X
Configuration 7 (Conf 7)	X	X	X

Table 2
Devices specifications.

Device	Micro-controller (MHz)	Flash (KB)	SRAM (KB)	Voltage (V)
Arduino Uno	ATmega328P-PU (16)	32	2	5
Arduino Mega	ATMega1280 (16)	128	8	5
Our sensor node	ATmega328P-PU (8)	32	2	3
Arduino Micro	ATmega32U4 (16)	32	2.5	5
Sensor node in [18]	ATMega32L (8)	256	8	5
Sensor node in [19]	ATMega128L (8)	128	4	3
Sensor node in [16]	MSP430F2617 (8)	92	8	3.7
Sensor node in [21]	MSP430 (8)	48	10	3
Sensor node in [20]	MSP430F1611 (8)	48	10	3.7
Z1	MSP430 (8)	92	8	3

Table 3
Energy consumption of the sensor node when collecting 50 samples/s acceleration data via SPI in Mode 1 and Mode 2 during a second.

	Mode 1(mJ)	Mode 2(mJ)
Energy consumption of a sensor node	30.47	27.98

Mode 1: when unneeded modules are turned on or enabled
Mode 2: when unneeded modules are turned off or disabled

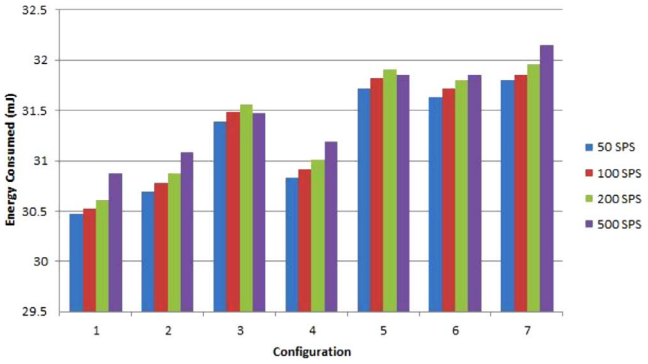

Fig. 11. Energy consumed per second when collecting multiple sensor data at different sampling rates via SPI.

modules at the same time causes a slight increase in energy consumption when compared with applying a single sensor (i.e. 3-d accelerometer or 3-d gyroscope).

In order to investigate the impact of software-based techniques on energy consumption of the sensor node, we measure energy consumption of the sensor node for one second with two cases: (i) when unneeded modules (e.g. UART, ADC, I^2C, and brownout detection) of a micro-controller are turned on or enabled; (ii) when unneeded modules

are turned off or disabled. In the experiment, the sensor node collects 3-d acceleration data at a rate of 50 samples/s in a second via 1 Mbps SPI and the nRF module is not active. Results from Table 3 show that energy required by the sensor node can be reduced by about 8–10% when unneeded internal modules of a micro-controller are disabled.

The nRF module can be configured for two modes of communication- one-way and two-way. In one way communication, a sensor node only sends data to a receiver (i.e. a gateway) regardless of the success of a transmitted package. On the other hand, in two-way communication, a sensor node sends data to a gateway and waits for an acknowledgement message from the gateway. If it receives an acknowledgement message from the gateway, it continuously sends new data to the gateway. In contrast, when it does not receive any acknowledgement message from the gateway, it automatically increases transmission power and re-sends the package which was not successfully received by the gateway in the previous attempt. There is a trade-off between QoS and energy consumption when applying these communication types. Two-way communication guarantees that a data package is received at a gateway after being sent by a sensor node. However, it causes higher energy consumption because a sensor node's down-link must be active to wait for the response package. In contrast, one-way communication consumes less energy because its down-link is disabled, however, the best possible QoS cannot be guaranteed. Correspondingly, these communication types must be attentively investigated for exposing the optimal configuration providing a high level of QoS and energy efficiency.

At first, energy consumption of a sensor node in a two-way communication is measured. Several sensor nodes are used during the experiments in which each sensor node is attached to a patient's clothing at the middle of the chest area. Sensor nodes collected 3-d accelerometer data with a data rate of 50 samples/s via 1 Mbps SPI and transmit the data via nRF to a gateway which is fixed in a single room. Several distances between sensor nodes and a smart gateway such as 5, 10 and 20 m are applied for evaluating variations of sensor nodes' energy consumption. In each measurement, both cases of the line of sight transmission and transmission via blocked objects (i.e. door and wall) are applied. Results of two-way communication are shown in Table 4. Energy consumption of a sensor node in this case includes both energy consumption of transmitting and receiving. The results indicate that energy consumption of the sensor node increases when the distance between the sensor node and the gateway increases.

Obviously, two-way communication often provides a high level of QoS because a loss package is always re-transmitted. In case of higher distances or transmission way blocked, transmission power in two-way communication is increased for ensuring a successful data transfer. In the experiments, the transmission power is retrieved effortlessly via the monitor utility. For achieving such a high level of QoS, the transmission power used in two-way communication can be re-applied into a case of one-way communication. Similarly, the same test-bed with similar distances (5, 10, and 20 m) is applied for one-way communication. Results shown in Table 5 indicate that energy consumption of a sensor node in case of one way communication is much less than in case of two-way communication in both situations (e.g. line-of-sight transmission and transmission through blocked objects) even though transmission power of a sensor node in case of one-way communication is forced to be increased for assuring a high level of QoS.

Table 4
Energy consumption of the sensor node when collecting 50 samples/s acceleration via SPI and transmitting the data via nRF in two-way communication to a gateway during a second.

Distance	5 m(mJ)	10 m(mJ)	20 m(mJ)
Line-of-sight transmission	41.96	42.28	43.99
Transmission through blocked objects	43.21	43.94	45.80

Table 5
Energy consumption of the sensor node when collecting 50 samples/s acceleration via SPI and transmitting the data via nRF to a gateway (one-way communication) during a second.

Distance	5 m(mJ)	10 m(mJ)	20 m(mJ)
Line-of-sight transmission	28.71	29.01	30.72
Transmission through blocked objects	29.81	30.48	32.61

Although many slave devices (i.e. sensors) can be connected to a master (i.e. micro-controller) via a single SPI interface, it is challenging to perform such a connection in some cases. For example, hangout wires may occur in the layout design when connecting several devices to a single SPI port or SPI libraries of slave devices may conflict. In order to avoid these issues, software SPI which written in C utilizes Pulse Width Modulation (PWM) pins for replicating an SPI transmission, can be used. In the experiments, energy consumption of two different sensor nodes is measured in which the first node uses only SPI and the second node uses a combination of SPI and software SPI. Both nodes use their watchdog timer for waking up the micro-controller from the deep sleep mode(s). Several distances (5, 10, 20 m) are applied and the data is transmitted via nRF with a line of sight transmission during these experiments. Results shown in Table 6 show that hardware SPI is more energy efficient than software SPI in all experimental cases. For avoiding missing package when applying one-way communication, we apply the same method of reusing transmission power in case of two-way communication into one-way communication for both sensor nodes (nodes using hardware SPI and nodes using both hardware and software SPI) in each measurement. Depending on particular distances, the transmission power is different.

With the purpose of providing a complete view of energy consumption of the sensor node, we investigate energy consumption of the sensor node in different configurations shown in Table 1. In the experiments, the sensor node acquires different data from one or several sensor types with a sampling rate of 50 samples/s and transmits the data via nRF with a line-of-sight transmission condition. Energy consumption is measured for one second. Results of the experiments shown in Table 7 show that 3-d accelerometer consumes the least energy while 3-d gyroscope and 3-d magnetometer consume higher and the most energy, respectively. In addition, the results reveal that applying two types of sensors in the sensor node consumes about 12–16% more energy and energy consumption of the sensor node equipped with two or three types of sensors is slightly different, approximately 3–5%.

When supplying with a 1000 mAh 3 V lithium battery having a size of 32 mm * 43 mm * 5 mm and a weight of 30 g, the wearable sensor node can operate for about 76–90 h depending on particular conditions. In the paper, we simply categorize into three conditions including the worst, normal and the best condition. In the worst condition, transmission path between a sensor node and a gateway is blocked with different indoor objects and doors. For example, the sensor node is placed is a room while a gateway is located in another room and these rooms are separated by walls and doors. In normal situation, there are very few objects in transmission path. In the experiment, some high wardrobes are placed in a room. In the best situation, the transmission path is clear (i.e. line-of-sight) and the experimentation room is almost

Table 6
Energy consumption of the sensor node when collecting 50 samples/s acceleration and transmitting the data to an nRF block via software and hardware SPI during a second.

	Distance		
Method	5 m (mJ)	10 m (mJ)	20 m (mJ)
Software SPI	30.98	31.93	33.7
Hardware SPI	28.68	29.01	30.72

Table 7
Energy consumption of the sensor node when collecting 50 samples/s motion data via SPI and transmitting the data via nRF with a distance of 20 m in different configurations.

\Distance	20 m (mJ)
Conf 1	30.72
Conf 2	30.95
Conf 3	32.1
Conf 4	34.53
Conf 5	35.56
Conf 6	35.35
Conf 7	36.68

empty (e.g. only short tables and chairs). In these experiments, the sensor node is placed on the top the battery for forming a compact device and it takes about 2–5 h to charge the battery depending on the current supplied. The sensor node collects data from 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer with a data rate of 50 samples/s via 1 Mbps SPI and transmits the data via nRF. When the sensor node is not active (e.g. all modules such as nRF and sensors are disabled), it consumes about 3.6 mW. Results of these experiments are shown in Fig. 14. It can be seen that the sensor node can operate up to 76 h in the worst condition while its operating time can reach up to 90 h in the best condition. It is recommended that, the transmission power of the sensor node should be configured for suiting the worst condition because it can provide a high level of QoS for all cases. In the worst condition, some levels of energy efficiency (about 5–8%) must be sacrificed.

For providing a comprehensive view of the wearable sensor node, the sensor node is compared with other nodes proposed by other research in terms of energy consumption, size, weight and flexibility. In our context, high flexibility indicates that a sensor node can be customized easily and flexibly for suiting to different fall detection IoT-based systems and vice versa. Results shown in Table 8 summarize that our wearable sensor node is tiny, light-weight and energy efficient. In addition, the sensor node, which is highly flexible, suits to different IoT-based fall detection systems using motion data whilst other nodes are not completely suitable for other fall detection systems. Correspondingly, a user can wear the sensor node 24/7 without interfering daily activities.

In addition to the previous experiments, to evaluate quality of acquired signals at the gateway, 6 more measurements have been carried out. In details, each measurement uses 5 separate sensor nodes placed on the body of five volunteers for acquiring both 3-d accelerometer data and 3-d gyroscope data. Then the data is transmitted via nRF to the gateway with a line-of-sight transmission path condition. Each measurement is carried out for 30 min. Data received at the gateway is applied for the fall detection algorithms mentioned in Section 5.2. Results from the experiments shown in Fig. 12 reveal that the sensor node operates reliably in different scenarios (i.e. different daily activities) in most of the cases. In order to provide an incisive view of the data received at the gateway, the data is graphed in MatLab. In addition, the exceptional case is shown in Fig. 13.

Table 8
Devices specifications.

Device	Energy consumption	Size	Weight	Flexibility
Arduino Uno	High	Large	Medium	Partly
Arduino Mega	High	Large	Medium	Partly
Arduino Micro	Medium	Small	Light	Partly
Sensor node in [18]	Low	Medium	Medium	Partly
Sensor node in [19]	Medium	Medium	Medium	Partly
Sensor node in [16]	Low	Medium	Light	Partly
Sensor node in [21]	Low	Medium	Medium	Partly
Sensor node in [20]	High	Large	Medium	Partly
Z1	Low	Medium	Medium	Partly
Our sensor node	Low	Small	Light	Completely

Fig. 12 shows 3-d accelerometer data and 3-d gyroscope data whose sampling rate is 50 samples/s during a user's daily activities such as standing, sitting and walking. As seen in Fig. 12, quality of data collected from 3-d accelerometer and 3-d gyroscope is high in different cases such as "stand still", "sit still", "stand with body movements", "sit with body movements" and "walking". Due to some movements of the upper part of the body when walking, SVM of 3-d accelerometer values and 3-d gyroscope values fluctuate. However, the fluctuation of SVM of 3-d accelerometer values and 3-d gyroscope values is not large enough to dramatically impact on the result of fall detection since a variation of the fluctuation is much smaller than the peak magnitude of SVM of 3-d accelerometer values and 3-d gyroscope values when a user falls shown in Fig. 13. In other cases, the fluctuation of SVM from 3-d accelerometer and 3-d gyroscope is small, around 1 g and 0°s, respectively, which are similar to expected values discussed in Section 5.

Fig. 13 shows SVM of 3-d accelerometer values and 3-d gyroscope values with a sampling rate is 50 samples/s when a volunteer falls. It can be seen that magnitude of SVM increases dramatically in all cases. In case of 3-d accelerometer, the peak of SVM goes over 1.6 g for all of considered cases. In some of cases, the peak even reaches up to 2.5 g or 3 g. In case of "walk and fall", when a person falls, SVM of 3-d accelerometer reaches up to 3.5 g. However, when a person stands still after falling, the SVM value does not go back to 1 g (the expected value) but it remains at 2 g. However, in reality, the 2 g value is not the correct value. The reason might be incorrect calibration in the 3-d accelerometer or the large movement of the body when standing up. In this case, if the fall detection algorithm sets the threshold for detecting a fall at 1.8 g, the fall detection results are completely incorrect. In contrast, SVM from 3-d gyroscope, which is around 0°s, is correct as expected.

It is known that all artifacts with large angles negatively impact on the fall detection decision of the gyroscope-based system because the noise amplitude caused by movement artifacts are sometime larger that pre-defined thresholds used for determining a fall case. Therefore, to validate the system in such a case, the system is applied to a person who has an unbalance stance (e.g. moving his shoulder, hands and an upper part of his body with large angles) while walking. The results of the experiment are shown in Figs. 15 and 16. It can be seen that, SVM values in all experiment cases except the case "walking", which are around 1 g from SVM of 3-d accelerometer values and 0 deg/s from SVM of 3-d gyroscope values, are as expected. In case of walking, SVM of 3-d accelerometer values is 1 g as expected while SVM of 3-d gyroscope values varies dramatically. At some instances, the SVM values are even larger than 100 deg/s. In those cases, it is obvious that if the system applies a low threshold value close to 100 deg/s, an incorrect alarm will be triggered. Defining threshold values for a fall detection system based on accelerometer and gyroscope is not an easy task. Low threshold values help to reduce missing cases when SVM of 3-d accelerometer values or 3-d gyroscope values are not large. However, they may cause incorrect alarms or notifications. In contrast, high threshold values may cause missing fall detection cases, but they help to reduce incorrect alarms of falling cases. In addition, the results in a walking case in Fig. 15 unveil that replying on merely 3-d gyroscope may lead to incorrect alarms of falling cases. Comparing between results from "walking" in Fig. 15 and "walk and fall forward" in Fig. 16, it can be seen that SVM of 3-d gyroscope values are slightly different (i.e. around 105 versus 130 deg/s) while SVM of 3-d accelerometer values are largely different. In this case, values from a 3-d accelerometer are better in terms of the fall detection accuracy. To sum up, in order to avoid an incorrect fall detection alarm, many types of sensors (i.e. 3-d accelerometer, 3-d gyroscope and 3-d magnetometer) and two-level thresholds should be considered to be applied in a sensor node.

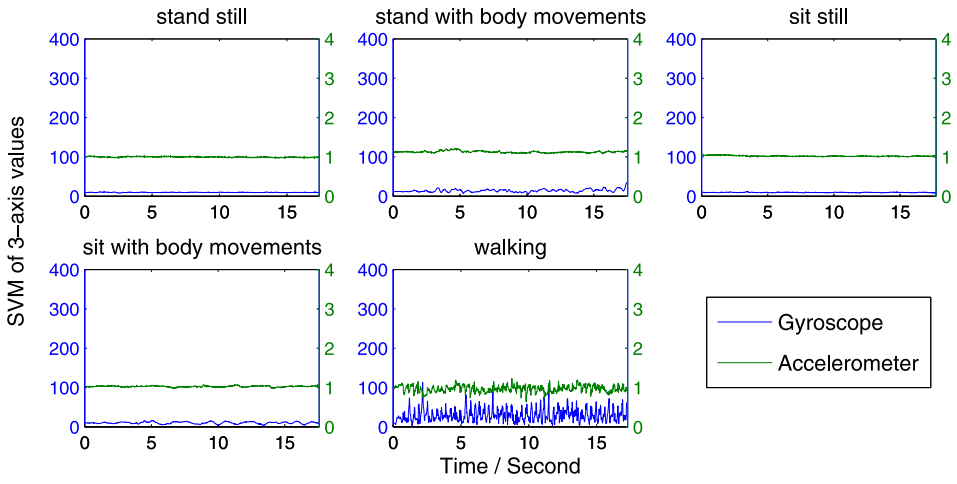

Fig. 12. Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities.

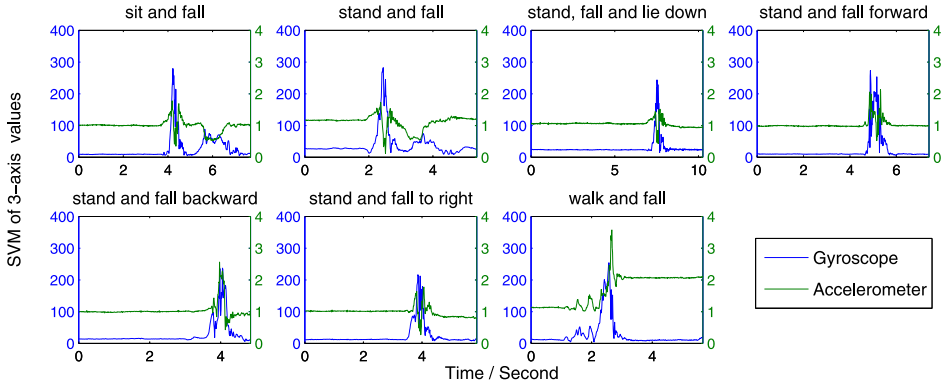

Fig. 13. Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities and fall.

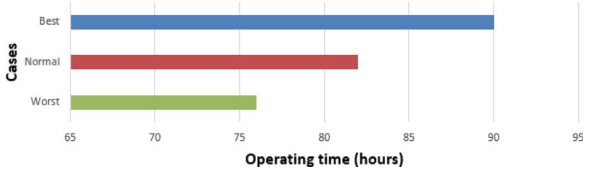

Fig. 14. Operating duration of the wearable sensor node supplied with a 1000 mAh battery when collecting data from accelerometer, gyroscope and magnetometer with a data rate of 50 samples/s via SPI and sending the data via nRF during a second under different conditions.

7. Discussions

Designing an energy efficient sensor node for fall detection and other health-care systems is not a simple task because the sensor node must fulfill strict requirements of healthcare IoT systems (e.g. latency and high quality of signal) while consuming low energy.

In order to achieve a high level of energy efficiency, wireless communication protocols are often attentively considered first. Before applying nRF for our sensor nodes, some of ESP8266 chips are integrated in sensor nodes for the experiments. Sensor nodes communicate via Wi-

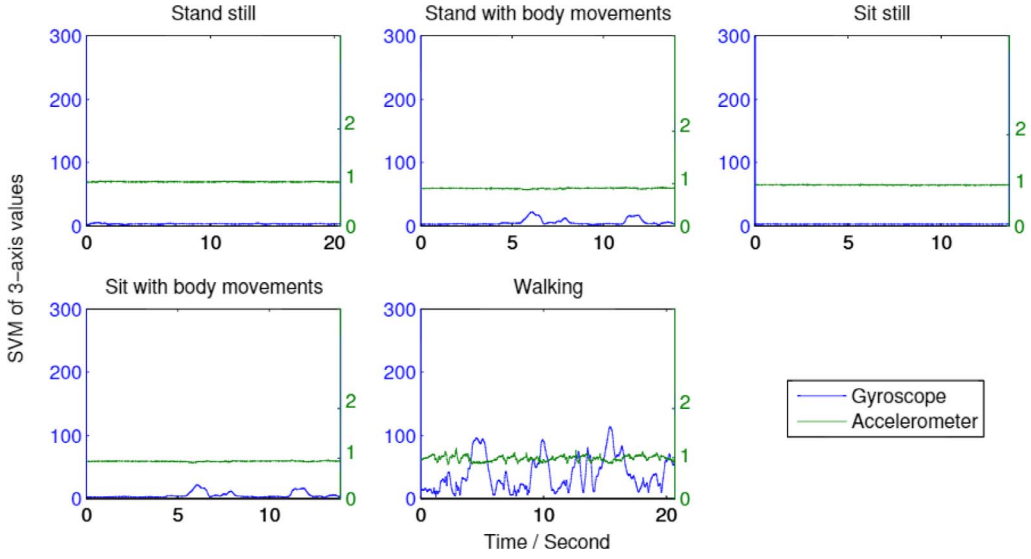

Fig. 15. Accelerometer and Gyroscope data at the gateway's nRF receiver during daily activities and fall.

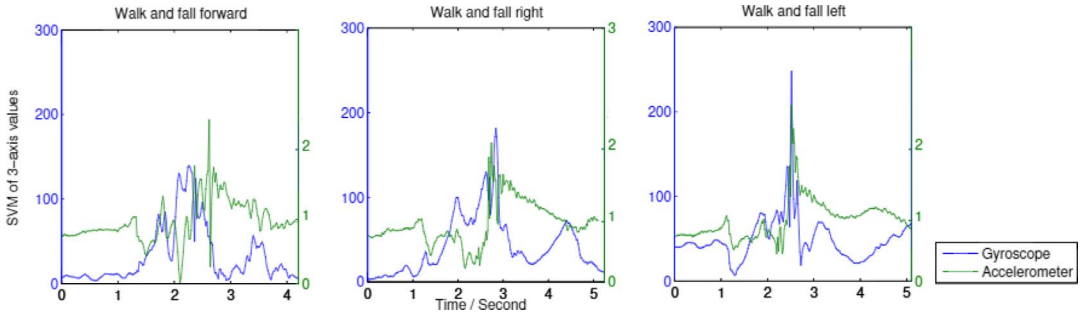

Fig. 16. Accelerometer and Gyroscope data at the gateway's nRF receiver during daily activities and fall.

Fi with a gateway fixed at the roof of the room and the transmission is line-of-sight with a distance of 7 m. When applying those ESP8266 chips, energy consumption during a second of a sensor node equipped with ESP8266 is about 270.6 mJ and the sensor node requires 460–480 mW for sending a packet. In the experiments, the sensor nodes collect data from a 3-d accelerometer, a 3-d gyroscope with a data rate of 50 samples/s. They accumulate the collected data and send one packet per second assuming that the maximum latency is 1 s. In case of other sampling rates (i.e. 50, 100 and 200 samples/s) of 3-d accelerometer and 3-d gyroscope, the maximum volume of data which the sensor node can accumulate for a single packet is 1800 Bytes, 3600 Bytes, or 7200 Bytes, respectively. The sensor node cannot accumulate more data for a single packet because the maximum latency (i.e. 1 s) requirement will be infringed. In case of the sensor node equipped with nRF, it is required 75, 150 and 300 packets to transmit the same amount of data in one second because the sensor node with nRF can send the maximum of 24 Bytes per packet. Energy consumption of the sensor node for transmitting 75, 150 and 300 packets in a second with nRF is around 31–50 mJ. Hence, in an IoT-based fall detection application, nRF is more suitable than Wi-Fi. In other applications (e.g. Electroencephalography (EEG) real-time monitoring), Wi-Fi might be more suitable due to a large amount of data (e.g. 10,000 Bytes/s per channel) is collected in a short period of time.

In terms of energy efficiency between low-power wireless communication protocols (i.e. BLE, ANT, 6LoWPAN and nRF) nRF and BLE are more energy efficient [48,49]. Precisely, nRF is more energy efficient than BLE [50,51]. For example, power consumption of a BLE chip and an nRF chip is around 46.2 mW and 37.29 mW at 0 dBm output power, respectively [50,51]. One of the main reasons of low energy in nRF is that nRF does not use any software stack. In our experiments, by replacing BLE in the sensor node presented in our previous work [11] by nRF, approximately 10% energy can be saved. In terms of connectivity, nRF is more suitable for IoT-based fall detection systems than BLE because BLE supports peer-to-peer communication while nRF supports many-to-many communication. Although nRF is energy efficient, it has several limitations (e.g. difficulty for a gateway to handle data sent simultaneously by many sensor nodes). When applying nRF, a transmission payload must be attentively considered for achieving energy efficiency and accuracy. In default, nRF uses a maximum payload of 32 Bytes as a static payload in each packet. When data is less than 32 Bytes (e.g. 4 Bytes), the nRF protocol automatically adds extra Bytes for filling up 32 Bytes payload (e.g. adding 28 Bytes). However, sending data with a size of 32 Bytes is not an optimal choice because some bytes of data may be collapsed at the receiver(s). Therefore, it is recommended to send the data with a size of 20–24 Bytes per packet.

8. Conclusions

We presented the design and implementation of an energy efficient wearable device for IoT-based fall detection systems. The device is tiny, light-weight and flexible hence suits to different IoT-based fall detection systems and can be used regularly without interfering user's daily activities. In this paper, we evaluated energy consumption of wearable sensor nodes in different configurations and scenarios to find optimal solutions for improving energy efficiency. We investigated configuration parameters (i.e. communication bus interface, and sampling rate) affecting energy consumption of the wearable device. In addition, important hardware and software factors or techniques impacting on the life-time of the sensor node are investigated. Besides, we evaluated energy consumption of the device in different transmission conditions for providing hints to system administrators for avoiding missed data while maintaining a high level of energy efficiency in the wearable device. Furthermore, we compared the wearable device with different devices proposed by others. The result shows that our wearable sensor node is the best among compared nodes. The results from conducted experiments conclude that our sensor node can operate around 76 h with a 1000 mAh battery in a tough transmission condition. Moreover, we implemented a complete IoT-based fall detection system consisting of smart gateways with Fog computing and a back-end system. When a fall occurs, the system can detect and remotely inform responsible personnel such as a doctor or caregiver in real-time. It can be concluded that the proposed wearable device is a solution to drawbacks of typical sensor nodes in IoT-based fall detection systems.

Acknowledgements

We sincerely thank MATTI program from University of Turku Graduate School, Academy of Finland, Nokia Foundation and Finnish Foundation of Technology Promotion (TES) for supporting our project.

References

[1] D.A. Sterling, J.A. O'Connor, J. Bonadies, Geriatric falls: injury severity is high and disproportionate to mechanism, J. Trauma Acute Care Surg. 50 (1) (2001) 116–119.
[2] J.A. Stevens, et al., The costs of fatal and non-fatal falls among older adults, Injury Prev. 12 (5) (2006) 290–295.
[3] M.E. Tinetti, M. Speechley, S.F. Ginter, Risk factors for falls among elderly persons living in the community, N. Engl. J. Med. 319 (26) (1988) 1701–1707.
[4] B.H. Alexander, F.P. Rivara, M.E. Wolf, The cost and frequency of hospitalization for fall-related injuries in older adults. Am. J. Public Health 82 (7) (1992) 1020–1023.
[5] Important facts about falls. Updated: Jan. 2017, Accessed Feb 2017, https://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html.
[6] S.M. Friedman, et al., Falls and fear of falling: which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention, J. Am. Geriatr. Soc. 50 (8) (2002) 1329–1335.
[7] A. C. Scheffer, et al., Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons, Age Ageing 37 (1) (2008) 19–24.
[8] R. Igual, et al., Challenges, issues and trends in fall detection systems, Biomed. Eng. Online 12 (1) (2013) 1.
[9] M. Mubashir, et al., A survey on fall detection: principles and approaches, Neurocomputing 100 (2013) 144–152.
[10] J. T. Perry, et al., Survey and evaluation of real-time fall detection approaches, High-Capacity Optical Networks and Enabling Technologies (HONET), 2009 6th International Symposium on, IEEE, 2009, pp. 158–164.
[11] T. N. Gia, et al., Iot-based fall detection system with energy efficient sensor nodes, Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE, IEEE, 2016, pp. 1–6.
[12] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16.

[13] A. Rahmani, et al., Fog Computing in the Internet of Things, Springer International Publishing, 2018.
[14] E. Casilari, M.A. Oviedo-Jiménez, Automatic fall detection system based on the combined use of a smartphone and a smartwatch, PLoS ONE 10 (11) (2015) e0140929.
[15] M. Kepski, B. Kwolek, Embedded system for fall detection using body-worn accelerometer and depth sensor, (IDAACS), 2015 IEEE 8th International Conference on, 2 IEEE, 2015, pp. 755–759.
[16] P. Pivato, et al., A wearable wireless sensor node for body fall detection, Measurements and Networking Proceedings (M&N), 2011 IEEE International Workshop on, IEEE, 2011, pp. 116–121.
[17] D. Chen, W. Feng, Y. Zhang, X. Li, T. Wang, A wearable wireless fall detection system with accelerators, Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, IEEE, 2011, pp. 2259–2263.
[18] O. Biros, et al., Implementation of wearable sensors for fall detection into smart household, Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on, IEEE, 2014, pp. 19–22.
[19] S.Z. Erdogan, T.T. Bilgin, A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data, IET Commun. 6 (18) (2012) 3281–3287.
[20] F. Wu, H. Zhao, Y. Zhao, H. Zhong, Development of a wearable-sensor-based fall detection system, Int. J. Telemed. Appl. 2015 (2015) 2.
[21] Y. Li, et al., Accelerometer-based fall detection sensor system for the elderly, 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, 3 IEEE, 2012, pp. 1216–1220.
[22] C. Shuo, Fall detection system using Arduino Fio, Proceedings of the IRC Conference on Science, Engineering and Technology, Singapore, 13 (2015).
[23] Y. Cheng, C. Jiang, J. Shi, A fall detection system based on sensortag and Windows 10 iot core (2015).
[24] R. Freitas, M. Terroso, M. Marques, J. Gabriel, A.T. Marques, R. Simoes, Wearable sensor networks supported by mobile devices for fall detection, SENSORS, 2014 IEEE, IEEE, 2014, pp. 2246–2249.
[25] I. Tcarenko, et al., Energy-efficient iot-enabled fall detection system with messenger-based notification, International Conference on Wireless Mobile Communication and Healthcare, Springer, 2016, pp. 19–26.
[26] T. N. Gia, et al., Fog computing in healthcare internet of things: a case study on ecg feature extraction, (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, IEEE, 2015, pp. 356–363.
[27] A. M. Rahmani, et al., Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach, Future Gener. Comput. Syst. (2017).
[28] B. Negash, et al., Leveraging fog computing for healthcare iot, Fog Computing in the Internet of Things, Springer, 2018, pp. 145–169.
[29] I. Fredriksen, P. Kastnes, Choosing a mcu for your next design; 8 bit or 32 bit?, 2014, Accessed Jul 2017, http://www.atmel.com/images/45107a-choosing-a-mcu-fredriksen_article_103114.pdf.
[30] Y.S. Delahoz, M.A. Labrador, Survey on fall detection and fall prevention using wearable and external sensors, Sensors 14 (10) (2014) 19806–19842.
[31] F. Touati, R. Tabish, U-healthcare system: state-of-the-art review and challenges, J. Med. Syst. 37 (3) (2013) 9949.
[32] RaspberryOrg, Raspberry-pi-3-model-b. Updated: Jan. 2016, Accessed Jul 2016, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.
[33] A. Z. Rakhman, et al., Fall detection system using accelerometer and gyroscope based on smartphone, Information Technology, Computer and Electrical Engineering (ICITACEE), 2014 1st International Conference on, IEEE, 2014, pp. 99–104.
[34] P-K. Chao, et al., A comparison of automatic fall detection by the cross-product and magnitude of tri-axial acceleration, Physiol. Meas. 30 (10) (2009) 1027.
[35] A. Bourke, et al., Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities, J. Biomech. 43 (15) (2010) 3051–3057.
[36] Q. T. Huynh, et al., Optimization of an accelerometer and gyroscope-based fall detection algorithm, J. Sens. 2015 (2015).
[37] L. Palmerini, et al., A wavelet-based approach to fall detection, Sensors 15 (5) (2015) 11575–11586.
[38] G. Rescio, et al., Support vector machine for tri-axial accelerometer-based fall detector, Advances in Sensors and Interfaces (IWASI), 2013 5th IEEE International Workshop on, IEEE, 2013, pp. 25–30.
[39] J. He, C. Hu, X. Wang, A smart device enabled system for autonomous fall detection and alert, Int. J. Distrib. Sens. Netw. 12 (2) (2016) 2308183.
[40] R. M. Gibson, et al., Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic, Appl. Soft Comput. 39 (2016) 94–103.
[41] T. N. Gia, et al., Low-cost fog-assisted health-care iot system with energy-efficient sensor nodes, The 13th International Wireless Communications and Mobile Computing Conference (IWCMC 2017), IEEE, 2017.
[42] S. R. Moosavi, et al., End-to-end security scheme for mobility enabled healthcare internet of things, Future Gener. Comput. Syst. 64 (2016) 108–124.
[43] A. M. Rahmani, et al., Smart e-health gateway: bringing intelligence to internet-of-things based ubiquitous healthcare systems, (CCNC), 2015 12th Annual IEEE, IEEE, 2015, pp. 826–834.
[44] T. N. Gia, et al., Iot-based continuous glucose monitoring system: a feasibility study, Procedia Comput. Sci. 109 (2017) 327–334.
[45] T. N. Gia, et al., Fog computing in body sensor networks: an energy efficient approach, IEEE International Body Sensor Networks Conference (BSN15), (2015).
[46] M. Solutions, Power monitor. Updated: Jan. 2016, Accessed Jul 2016, https://www.msoon.com/LabEquipment/PowerMonitor/.
[47] Mpu-9250. Updated: Jan. 2017, Accessed Feb 2017, https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/.
[48] T. N. Gia, et al., Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems, NORCHIP, 2014, IEEE, 2014, pp. 1–6.
[49] A. Dementyev, et al., Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario, Wireless Symposium (IWS), 2013 IEEE International, IEEE, 2013, pp. 1–4.
[50] S. Kamath, J. Lindh, Measuring bluetooth low energy power consumption, Texas instruments application note AN092, Dallas(2010).
[51] nrf24l01+. Updated: Jan. 2017, Accessed Feb 2017, https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P.

Tuan Nguyen Gia received his Master's degree (Tech) from the University of Turku (Finland) in 2014. He received a 4-year funded position in University of Turku, a grant scholarship from Nokia Foundation for an efficient and excellent researcher in 2015, and a grant scholarship from Finnish Foundation of Technology Promotion in 2017. He joined Internet-of-Things for Healthcare (IoT4Health) research group, University of Turku in 2014. Currently, he works as a researcher and a doctoral candidate in the same group.

Victor Kathan Sarker received his Bachelor's degree on Electronic and Telecommunication Engineering from North South University, Dhaka, Bangladesh in 2012. Currently he is pursuing Master's degree on Embedded Computing in University of Turku. Besides, he works in the Internet-of-Things for Healthcare (IoT4Health) research group in Department of Future Technologies, University of Turku.

Igor Tcarenko received his Master (Tech) from the University of Turku (Turku, Finland) in 2014. Currently, he works at the Internet-of-Things for Healthcare (IoT4Health) research group in IT department, University of Turku.

Amir M. Rahmani received his Master's degree from Department of Electrical and Computer Engineering, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of Information Technology, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation & Technology (EIT) ICT Labs, in 2014. He is currently Marie Curie Global Fellow at University of California Irvine (USA) and TU Wien (Austria). He is also an adjunct professor (Docent) in embedded parallel and distributed computing at the University of Turku, Finland. He is the author of more than 140 peer-reviewed publications. He is a senior member of the IEEE.

Tomi Westerlund received his Ph.D. (Tech) from the University of Turku (Turku, Finland) in 2008. He joined the Department of Information Technology, University of Turku as a Senior Researcher, and in 2015 became a University Research Fellow. Since 2013, he has acted yearly as a visiting scholar at Fudan University, Shanghai, China. His current research interest is Internet of Things (IoT); how we can utilise IoT technology to provide better services and improve the quality of life. With that in mind, the main application areas for his research are smart agriculture, smart cities and health technology.

Pasi Liljeberg received the M.Sc. and Ph.D. degrees in electronics and information technology from the University of Turku, Turku, Finland, in 1999 and 2005, respectively. He is a Senior University Lecturer in Embedded Electronics Laboratory and an adjunct professor in embedded computing architectures at the University of Turku, Embedded Computer Systems laboratory. During the period 2007–2009, he held an Academy of Finland researcher position. He is the author of more than 200 peer-reviewed publications, has supervised nine Ph.D. theses. Liljeberg is the leader of the Internet-of-Things for Healthcare (IoT4Health) research group.

Hannu Tenhunen received the diplomas from the Helsinki University of Technology, Finland, 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, 1986. In 1985, he joined the Signal Processing Laboratory, Tampere University of Technology, Finland, as an associate professor and later served as a professor and department director. Since 1992, ha has been a professor at the Royal Institute of Technology (KTH), Sweden, where he also served as a dean. He has more than 600 reviewed publications and 16 patents internationality. He is a member of the IEEE.